

Article

A Family of Derivative Free Algorithms for Multiple-Roots of Van Der Waals Problem

Sunil Kumar ¹, Ramandeep Behl ², Eulalia Martínez ^{3,*}, Fouad Mallawi ² and Sattam Alharbi ²

¹ Department of Mathematics, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Channai 601103, India; sfageria1988@gmail.com or k_sunil@ch.amrita.edu

² Department of Mathematics Sciences, King Abdulaziz University, Jeddah 21589, Saudi Arabia; ramanbehl87@yahoo.in (R.B.); fmallawi@hotmail.com (F.M.); rlal@kau.edu.sa (S.A.)

³ Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, 46022 Valencia, Spain

* Correspondence: eumarti@mat.upv.es

Abstract: There are a good number of higher-order iterative methods for computing multiple zeros of nonlinear equations in the available literature. Most of them required first or higher-order derivatives of the involved function. No doubt, high-order derivative-free methods for multiple zeros are more difficult to obtain in comparison with simple zeros and with first order derivatives. This study presents an optimal family of fourth order derivative-free techniques for multiple zeros that requires just three evaluations of function ϕ , per iteration. The approximations of the derivative/s are based on symmetric divided differences. We also demonstrate the application of new algorithms on Van der Waals, Planck law radiation, Manning for isentropic supersonic flow and complex root problems. Numerical results reveal that the proposed derivative-free techniques are more efficient in comparison terms of CPU, residual error, computational order of convergence, number of iterations and the difference between two consecutive iterations with other existing methods.

Keywords: multiple roots; symmetric divided differences; Van der Waals equation; Planck law equation; convergence

MSC: 65H05; 41A25; 49M15



Citation: Kumar, S.; Behl, R.;

Martínez, E.; Mallawi, F.; Alharbi, S.

A Family of Derivative Free Algorithms for Multiple-Roots of Van Der Waals Problem. *Symmetry* **2022**, *14*, 562. <https://doi.org/10.3390/sym14030562>

Academic Editor: Wei-Shih Du

Received: 17 December 2021

Accepted: 8 March 2022

Published: 11 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Finding the multiple zeros of nonlinear equations is an important and challenging task in the field of numerical analysis and applied sciences [1,2]. In this study, we consider iterative methods to find a multiple root α (having a known multiplicity $n > 1$) of a nonlinear equation of the following form:

$$\Phi(t) = 0,$$

where $\Phi : D \subset \mathbb{C} \rightarrow \mathbb{C}$ is an analytic function in D surrounding the required zero α .

Several higher-order techniques have been developed and analyzed in the literature (see [3–13]). Most of them are based on the modified Newton's method [14], which is given by:

$$t_{k+1} = t_k - n \frac{\Phi(t_k)}{\Phi'(t_k)}, \quad k = 0, 1, 2, \dots \quad (1)$$

It has a second-order convergence and is one of the best known one-point iterative methods for multiple zeros. However, it requires the evaluation of the first-order derivative at each step. Finding the derivative is not an easy task. In addition, the derivative-free methods are important in the cases where the derivative Φ' of function Φ is either very small, or does not exist or is not easy to evaluate.

Traub–Steffensen [15] proposed the following derivative free method:

$$\Phi'(t_k) \simeq \frac{\Phi(t_k + b \Phi(t_k)) - \Phi(t_k)}{b \Phi(t_k)}, \quad b \in \mathbb{R} - \{0\},$$

or

$$\Phi'(t_k) \simeq \Phi[u_k, t_k],$$

for Φ' in the Newton method (1). Here, $\Phi[u_k, t_k] = \frac{\Phi(u_k) - \Phi(t_k)}{u_k - t_k}$ is a divided difference and $u_k = t_k + b \Phi(t_k)$. Then, the recursive scheme (1) takes the form of the Traub–Steffensen method, which is defined as below:

$$t_{k+1} = t_k - n \frac{\Phi(t_k)}{\Phi[u_k, t_k]}. \tag{2}$$

Recently, some higher-order derivative-free methods have been presented in the literature (see [16–19]). Kumar et al. [16] suggested a second order one-point derivative free scheme. In addition, Behl et al. [17] and Kumar et al [18,19] advanced fourth order convergent derivative free methods for multiple zeros. According to the Kung–Traub hypothesis [20], the methods of [17–19] require three functional evaluations per iteration. Therefore, they have optimal convergence order.

The purpose of this study is to design some new efficient derivative-free techniques that are capable of achieving a high order convergence with a minimum number of evaluations of the involved function. Following these ideas, we derive two-step derivative-free techniques that have fourth order convergence. The new methods consume only three evaluations of the involved function, per iteration. So, it is an optimal scheme in the sense of the Kung–Traub hypothesis [20]. The algorithm is based on the Traub–Steffensen method (2) and is further modified in the second stage by using the Traub–Steffensen-like iteration. Numerical results also demonstrate the superiority of our methods over the existing ones.

2. Development of Scheme

For $n > 1$, we propose the following iterative approach:

$$\begin{aligned} v_k &= t_k - n \frac{\Phi(t_k)}{\Phi[u_k, t_k]}, \\ t_{k+1} &= v_k - nQ(x_k, y_k) \frac{\Phi(t_k)}{\Phi[u_k, t_k] + \Phi[v_k, u_k]}, \end{aligned} \tag{3}$$

where $x_k = \sqrt[n]{\frac{\Phi(v_k)}{\Phi(t_k)}}$ and $y_k = \sqrt[n]{\frac{\Phi(v_k)}{\Phi(u_k)}}$.

The results are calculated for different values of n . Firstly, we consider the case for $n = 2$ and establish the fourth-order convergence in the following Theorem 1.

Theorem 1. Consider that $t = \alpha$ is a multiple zero of Φ having multiplicity $n = 2$. We also assume that $\Phi : D \subset \mathbb{C} \rightarrow \mathbb{C}$ is an analytic function in D that contains the vicinity of the required zero α . Then, the algorithm (3) has fourth-order convergence, if

$$Q_{00} = 0, \quad Q_{10} = \frac{5}{4}, \quad Q_{01} = \frac{1}{4}, \quad Q_{20} = 6 - Q_{02} - 2Q_{11}, \quad Q_{11}, Q_{02}, \in \mathbb{R}$$

where $Q_{ij} = \frac{\partial^{i+j}}{\partial x^i \partial y^j} Q(x_k, y_k)|_{(x_k=0, y_k=0)}$, for $0 \leq i, j \leq 2$.

Proof. The error at the k -th stage is given by $\epsilon_k = t_k - \alpha$. Adopt the Taylor’s series expansion for the function $\Phi(t_k)$ about α with the assumptions $\Phi(\alpha) = 0, \Phi'(\alpha) = 0$ and $\Phi^{(2)}(\alpha) \neq 0$, we have:

$$\Phi(t_k) = \frac{\Phi^{(2)}(\alpha)}{2!} \epsilon_k^2 (1 + N_1 \epsilon_k + N_2 \epsilon_k^2 + N_3 \epsilon_k^3 + N_4 \epsilon_k^4 + \dots), \tag{4}$$

where $N_m = \frac{2!}{(2+m)!} \frac{\Phi^{(2+m)}(\alpha)}{\Phi^{(2)}(\alpha)}$ for $m \in \mathbb{N}$.

Similarly, we have the following Taylor’s series expansion of $\Phi(u_k)$ about α :

$$\Phi(u_k) = \frac{\Phi^{(2)}(\alpha)}{2!} \epsilon_{u_k}^2 (1 + N_1 \epsilon_{u_k} + N_2 \epsilon_{u_k}^2 + N_3 \epsilon_{u_k}^3 + N_4 \epsilon_{u_k}^4 + \dots), \tag{5}$$

where $\epsilon_{u_k} = u_k - \alpha = \epsilon_k + \frac{b\Phi^{(2)}(\alpha)}{2!} \epsilon_k^2 (1 + N_1 \epsilon_k + N_2 \epsilon_k^2 + N_3 \epsilon_k^3 + \dots)$.

By inserting expressions (4) and (5) in the first step of (3), we yield:

$$\begin{aligned} \epsilon_{v_k} &= v_k - \alpha \\ &= \frac{1}{2} \left(\frac{b\Phi^{(2)}(\alpha)}{2} + N_1 \right) \epsilon_k^2 - \frac{1}{16} ((b\Phi^{(2)}(\alpha))^2 - 8b\Phi^{(2)}(\alpha)N_1 + 12N_1^2 - 16N_2) \epsilon_k^3 + O(\epsilon_k^4). \end{aligned} \tag{6}$$

Expand the Taylor series expansion of $\Phi(v_k)$ about α , it follows:

$$\Phi(v_k) = \frac{\Phi^{(2)}(\alpha)}{2!} \epsilon_{w_k}^2 (1 + N_1 \epsilon_{w_k} + N_2 \epsilon_{w_k}^2 + N_3 \epsilon_{w_k}^3 + \dots). \tag{7}$$

Using (4), (5) and (7), we have:

$$\begin{aligned} x_k &= \frac{1}{2} \left(\frac{b\Phi^{(2)}(\alpha)}{2} + N_1 \right) \epsilon_k - \frac{1}{16} ((b\Phi^{(2)}(\alpha))^2 - 6b\Phi^{(2)}(\alpha)N_1 + 16(N_1^2 - N_2)) \epsilon_k^2 + \frac{1}{64} ((b\Phi^{(2)}(\alpha))^3 \\ &\quad - 22b\Phi^{(2)}(\alpha)N_1^2 + 4(29N_1^3 + 14b\Phi^{(2)}(\alpha)N_2) - 2N_1(3(b\Phi^{(2)}(\alpha))^2 + 104N_2) + 96N_3) \epsilon_k^3 + O(\epsilon_k^4), \end{aligned} \tag{8}$$

and

$$\begin{aligned} y_k &= \frac{1}{2} \left(\frac{b\Phi^{(2)}(\alpha)}{2} + N_1 \right) \epsilon_k - \frac{1}{16} (3(b\Phi^{(2)}(\alpha))^2 - 2b\Phi^{(2)}(\alpha)N_1 + 16(N_1^2 - N_2)) \epsilon_k^2 + \frac{1}{64} (7(b\Phi^{(2)}(\alpha))^3 \\ &\quad + 24b\Phi^{(2)}(\alpha)N_2 - 14b\Phi^{(2)}(\alpha)N_1^2 + 116N_1^3 - 2N_1(11(b\Phi^{(2)}(\alpha))^2 + 104N_2) + 96N_3) \epsilon_k^3 + O(\epsilon_k^4). \end{aligned} \tag{9}$$

From the expressions (8) and (9) that $x_k = O(\epsilon_k)$ and $y_k = O(\epsilon_k)$, respectively. Then, we expand the Taylor series expansion of weight function $Q(x_k, y_k)$ in the neighborhood of $(0, 0)$ in the following way:

$$Q(x_k, y_k) \approx Q_{00} + x_k Q_{10} + y_k Q_{01} + \frac{1}{2} x_k^2 Q_{20} + x_k y_k Q_{11} + \frac{1}{2} y_k^2 Q_{02}. \tag{10}$$

Inserting (4)–(10) in the last step of (3), we obtain

$$\begin{aligned} \epsilon_{k+1} &= -\frac{2}{3} Q_{00} \epsilon_k + \frac{1}{36} (b\Phi^{(2)}(\alpha)(9 + 10Q_{00} - 6Q_{01} - 6Q_{10}) + 6(3 + 2Q_{00} - 2Q_{01} - 2Q_{10})N_1) \epsilon_k^2 \\ &\quad + \frac{1}{432} (-(b\Phi^{(2)}(\alpha))^2(27 + 56Q_{00} - 84Q_{01} + 9Q_{02} - 48Q_{10} + 18Q_{11} + 9Q_{20}) + 12b\Phi^{(2)}(\alpha)(18 + 14Q_{00} \\ &\quad + 5Q_{01} - 3Q_{02} - Q_{10} - 6Q_{11} - 3Q_{20})N_1 - 12(20Q_{00} + 3(9 - 10Q_{01} + Q_{02} - 10Q_{10} + 2Q_{11} + Q_{20}))N_1^2 \\ &\quad + 144(3 + 2Q_{00} - 2Q_{01} - 2Q_{10})N_2) \epsilon_k^3 + \psi_m \epsilon_k^4 + O(\epsilon_k^5), \end{aligned} \tag{11}$$

where

$$\begin{aligned} \psi_m &= \psi_m(b, Q_{00}, Q_{10}, Q_{01}, Q_{20}, Q_{11}, Q_{02}, N_1, N_2, N_3), \\ &= \frac{1}{5184} (81(b\Phi^{(2)}(\alpha))^3 + 328(b\Phi^{(2)}(\alpha))^3 Q_{00} - 816(b\Phi^{(2)}(\alpha))^3 Q_{01} + 207(b\Phi^{(2)}(\alpha))^3 Q_{02} - 312(b\Phi^{(2)}(\alpha))^3 Q_{10} \\ &\quad + 306(b\Phi^{(2)}(\alpha))^3 Q_{11} + 99(b\Phi^{(2)}(\alpha))^3 Q_{20} - 12b\Phi^{(2)}(\alpha)(40Q_{00} + 3(45 + 5Q_{01} - 29Q_{02} - 19Q_{10} - 46Q_{11} - 17Q_{20}))N_1^2 \\ &\quad + 72(81 + 72Q_{00} - 131Q_{01} + 27Q_{02} - 131Q_{10} + 54Q_{11} + 27Q_{20})N_1^3 + 144b\Phi^{(2)}(\alpha)(36 + 24Q_{00} + 7Q_{01} - 6Q_{02} \\ &\quad - 5Q_{10} - 12Q_{11} - 6Q_{20})N_2 - 6N_1((b\Phi^{(2)}(\alpha))^2(135 + 216Q_{00} - 202Q_{01} - 75Q_{02} - 154Q_{10} - 78Q_{11} - 3Q_{20}) \\ &\quad + 48(45 + 34Q_{00} - 51Q_{01} + 6Q_{02} - 51Q_{10} + 12Q_{11} + 6Q_{20})N_2) + 7776N_3 + 5184Q_{00}N_3 - 5184Q_{01}N_3 - 5184Q_{10}N_3). \end{aligned}$$

Here, the expression of ψ_m is not reproduced explicitly due to its considerable length.

We set the coefficients of ϵ_k , ϵ_k^2 and ϵ_k^3 to zero at the same time and solving the resulting equations. Then, we obtain

$$Q_{00} = 0, \quad Q_{10} = \frac{5}{4}, \quad Q_{01} = \frac{1}{4}, \quad Q_{20} = 6 - Q_{02} - 2Q_{11}, \tag{12}$$

where $Q_{02}, Q_{11} \in \mathbb{R}$.

By using expression (12) in (11), we have:

$$\epsilon_{k+1} = \frac{1}{192} (b\Phi^{(2)}(\alpha) + 2N_1) ((b\Phi^{(2)}(\alpha))^2(3 + 4Q_{02} + 4Q_{11}) + 2b\Phi^{(2)}(\alpha)(11 + 4Q_{02} + 4Q_{11})N_1 + 62N_1^2 - 24N_2)\epsilon_k^4 + O(\epsilon_k^5).$$

Hence proved Theorem 1. \square

Theorem 2. We adopt the statement of Theorem 1 in the same sense. Then, the algorithm (3) has at least fourth order convergence for $n = 3$, if

$$Q_{00} = 0, \quad Q_{10} = \frac{4}{3} - Q_{01}, \quad Q_{20} = \frac{16}{3} - Q_{02} - 2Q_{11},$$

where $Q_{01}, Q_{02} \in \mathbb{R}$.

Proof. Keeping in the mind that $\Phi(\alpha) = 0, \Phi'(\alpha) = 0, \Phi^{(2)}(\alpha) = 0$ and $\Phi^{(3)}(\alpha) \neq 0$. Expand $\Phi(t_k)$ about α with the help of Taylor series expansion, we have

$$\Phi(t_k) = \frac{\Phi^{(3)}(\alpha)}{3!} \epsilon_k^3 (1 + \bar{N}_1 \epsilon_k + \bar{N}_2 \epsilon_k^2 + \bar{N}_3 \epsilon_k^3 + \bar{N}_4 \epsilon_k^4 + \dots), \tag{13}$$

where $\bar{N}_m = \frac{3!}{(3+m)!} \frac{\Phi^{(3+m)}(\alpha)}{\Phi^{(3)}(\alpha)}$ for $m \in \mathbb{N}$.

Similarly, Taylor series expansion of $\Phi(u_k)$ about α provide the following expression:

$$\Phi(u_k) = \frac{\Phi^{(3)}(\alpha)}{3!} \epsilon_{u_k}^3 (1 + \bar{N}_1 \epsilon_{u_k} + \bar{N}_2 \epsilon_{u_k}^2 + \bar{N}_3 \epsilon_{u_k}^3 + \bar{N}_4 \epsilon_{u_k}^4 + \dots), \tag{14}$$

where $\epsilon_{u_k} = u_k - \alpha$.

Using (13) and (14) in first step of (3), we get:

$$\begin{aligned} \sigma_{v_k} &= v_k - \alpha \\ &= \frac{\bar{N}_1}{3} \epsilon_k^2 + \frac{1}{18} (3b\Phi^{(3)}(\alpha) - 8\bar{N}_1^2 + 12\bar{N}_2) \epsilon_k^3 + \left(\frac{16}{27} \bar{N}_1^3 + \frac{\bar{N}_1}{9} (2b\Phi^{(3)}(\alpha) - 13\bar{N}_2) + \bar{N}_3 \right) \epsilon_k^4 + O(\epsilon_k^5). \end{aligned} \tag{15}$$

Expand the Taylor series expansion of $\Phi(v_k)$ about α , which is given by:

$$\Phi(v_k) = \frac{\Phi^{(3)}(\alpha)}{3!} \epsilon_{v_k}^3 (1 + \bar{N}_1 \epsilon_{v_k} + \bar{N}_2 \epsilon_{v_k}^2 + \bar{N}_3 \epsilon_{v_k}^3 + \bar{N}_4 \epsilon_{v_k}^4 + \dots). \tag{16}$$

From the expressions (13), (14) and (16), we have:

$$x_k = \frac{\bar{N}_1}{3} \epsilon_k + \left(\frac{b\Phi^{(3)}(\alpha)}{6} - \frac{5}{9} \bar{N}_1^2 + \frac{2}{3} \bar{N}_2 \right) \epsilon_k^2 + \left(\frac{23}{27} \bar{N}_1^3 + \frac{\bar{N}_1}{18} (3b\Phi^{(3)}(\alpha) - 32\bar{N}_2) + \bar{N}_3 \right) \epsilon_k^3 + O(\epsilon_k^4). \tag{17}$$

$$y_k = \frac{\bar{N}_1}{3} \epsilon_k + \left(\frac{b\Phi^{(3)}(\alpha)}{6} - \frac{5}{9} \bar{N}_1^2 + \frac{2}{3} \bar{N}_2 \right) \epsilon_k^2 + \left(\frac{23}{27} \bar{N}_1^3 + \frac{2}{9} \bar{N}_1 \left(\frac{b\Phi^{(3)}(\alpha)}{2} - 8\bar{N}_2 \right) + \bar{N}_3 \right) \epsilon_k^3 + O(\epsilon_k^4). \tag{18}$$

By using (10) and (13)–(18) in the last step of (3), we obtain:

$$\begin{aligned} \epsilon_{k+1} &= -\frac{3}{4} Q_{00} \epsilon_k + \frac{1}{12} (4 + 3Q_{00} - 3Q_{01} - 3Q_{10}) \bar{N}_1 \epsilon_k^2 + \frac{1}{144} \left(-2(32 + 24Q_{00} - 36Q_{01} \right. \\ &\quad \left. + 3Q_{02} - 36Q_{10} + 6Q_{11} + 3Q_{20}) \bar{N}_1^2 + 3(b\Phi^{(3)}(\alpha)(8 + 9Q_{00} - 6Q_{01} - 6Q_{10}) + 8(3Q_{00} \right. \\ &\quad \left. - 3Q_{01} - 3Q_{10} + 4)\bar{N}_2) \right) \epsilon_k^3 + \varphi_m \epsilon_k^4 + O(\epsilon_k^5), \end{aligned} \tag{19}$$

where $\varphi_m = \varphi_m(b, Q_{00}, Q_{10}, Q_{01}, Q_{20}, Q_{11}, Q_{02}, \bar{N}_1, \bar{N}_2, \bar{N}_3)$.

We set the coefficients of ϵ_k^2 and ϵ_k^3 to zero and solve the resulting equations. Then, we obtain:

$$Q_{00} = 0, \quad Q_{10} = \frac{4}{3} - Q_{01}, \quad Q_{20} = \frac{16}{3} - Q_{02} - 2Q_{11}. \tag{20}$$

Adopt the expression (20) in (19), we have the following error equation:

$$\epsilon_{k+1} = \frac{\bar{N}_1}{72} \left(b\Phi^{(3)}(\alpha)(3Q_{01} - 2) + 16\bar{N}_1^2 - 8\bar{N}_2 \right) \epsilon_k^4 + O(\epsilon_k^5).$$

Hence, we proved Theorem 2. □

3. Generalization of the Method

For the multiplicity $n \geq 4$, we define the following Theorem 3 for the method (3).

Theorem 3. Using the statement of Theorem 1, the algorithm (3) for the case $n \geq 4$ has at least fourth order convergence, if:

$$Q_{00} = 0, \quad Q_{10} = \frac{n+1}{n} - Q_{01}, \quad Q_{20} = \frac{4n+4}{4} - Q_{02} - 2Q_{11}, \quad Q_{01}, Q_{02}, Q_{11} \in \mathbb{R}.$$

Moreover, the error equation of (3) is given by

$$\epsilon_{k+1} = \left(\frac{9+n}{2n^3} P_1^3 - \frac{1}{n^2} P_1 P_2 \right) \epsilon_k^4 + O(\epsilon_k^5).$$

Proof. Keeping in mind that $\Phi^{(j)}(\alpha) = 0, j = 0, 1, 2, \dots, n - 1$, and $\Phi^n(\alpha) \neq 0$, we have the following Taylor series expansion of $\Phi(t_k)$ about α :

$$\Phi(t_k) = \frac{\Phi^n(\alpha)}{n!} \epsilon_k^n (1 + P_1 \epsilon_k + P_2 \epsilon_k^2 + P_3 \epsilon_k^3 + P_4 \epsilon_k^4 + \dots), \tag{21}$$

where $P_m = \frac{n!}{(m+n)!} \frac{\Phi^{(m+n)}(\alpha)}{\Phi^n(\alpha)}$ for $m \in \mathbb{N}$.

Similarly, expand $\Phi(u_k)$ about α leads us

$$\Phi(u_k) = \frac{\Phi^n(\alpha)}{n!} \epsilon_{u_k}^n (1 + P_1 \epsilon_{u_k} + P_2 \epsilon_{u_k}^2 + P_3 \epsilon_{u_k}^3 + P_4 \epsilon_{u_k}^4 + \dots), \tag{22}$$

where $\epsilon_{u_k} = u_k - \alpha = \epsilon_k + \frac{b\Phi^n(\alpha)}{n!} \epsilon_k^n (1 + P_1 \epsilon_k + P_2 \epsilon_k^2 + P_3 \epsilon_k^3 + \dots)$.

Use the expressions (21) and (22) in the first step of Equation (3), we obtain

$$\epsilon_{v_k} = \begin{cases} \frac{P_1}{4} \epsilon_k^2 + \frac{1}{16} (8P_2 - 5P_1^2) \epsilon_k^3 + \left(\frac{25}{64} P_1^3 - P_1 P_2 + \frac{1}{16} (b\Phi^{(4)}(\alpha) + 12P_3) \right) \epsilon_k^4 + O(\epsilon_k^5), & \text{if } n = 4, \\ \frac{P_1}{n} \epsilon_k^2 + \frac{1}{n^2} (2nP_2 - (1+n)P_1^2) \epsilon_k^3 + \frac{1}{n^3} ((1+n)^2 P_1^3 - n(4+3n)P_1 P_2 + 3n^2 P_3) \epsilon_k^4 + O(\epsilon_k^5), & \text{if } n \geq 5. \end{cases} \tag{23}$$

Expansion of $\Phi(v_k)$ around α yields:

$$\Phi(v_k) = \frac{\Phi^n(\alpha)}{n!} \epsilon_{v_k}^n (1 + P_1 \epsilon_{v_k} + P_2 \epsilon_{v_k}^2 + P_3 \epsilon_{v_k}^3 + P_4 \epsilon_{v_k}^4 + \dots). \tag{24}$$

Using (21), (22) and (24) in the expressions of x_k and y_k , we have:

$$x_k = \begin{cases} \frac{P_1}{4} \epsilon_k + \frac{1}{8} (4P_2 - 3P_1^2) \epsilon_k^2 + \frac{1}{128} (67P_1^3 - 152P_1 P_2 + 8(b\Phi^{(4)}(\alpha) + 12P_3)) \epsilon_k^3 + O(\epsilon_k^4), & \text{if } n = 4, \\ \frac{P_1}{n} \epsilon_k + \frac{1}{n^2} (2nP_2 - (2+n)P_1^2) \epsilon_k^2 + \frac{1}{2n^3} ((2n^2 + 7n + 7)P_1^3 - 2n(7+3n)P_1 P_2 + 6n^2 P_3) \epsilon_k^3 + O(\epsilon_k^4), & \text{if } n \geq 5, \end{cases} \tag{25}$$

and

$$y_k = \begin{cases} \frac{P_1}{4} \epsilon_k + \frac{1}{8} (4P_2 - 3P_1^2) \epsilon_k^2 + \frac{1}{128} (67P_1^3 - 152P_1 P_2 + 8(b\Phi^{(4)}(\alpha) + 12P_3)) \epsilon_k^3 + O(\epsilon_k^4), & \text{if } n = 4, \\ \frac{P_1}{n} \epsilon_k + \frac{1}{n^2} (2nP_2 - (2+n)P_1^2) \epsilon_k^2 + \frac{1}{2n^3} ((2n^2 + 7n + 7)P_1^3 - 2n(7+3n)P_1 P_2 + 6n^2 P_3) \epsilon_k^3 + O(\epsilon_k^4), & \text{if } n \geq 5. \end{cases} \tag{26}$$

Insert expressions (10) and (21)–(26) in the second step of (3), we have:

$$\begin{aligned} \epsilon_{k+1} = & -\frac{n}{n+1}Q_{00}\epsilon_k + \frac{1}{n+1}(n+1+nQ_{00}-nQ_{01}-nQ_{10})P_1\epsilon_k^2 + \frac{1}{2n^2(n+1)}(-2(n+1)^2+2n(n+1)Q_{00} \\ & -2n(n+3)Q_{01}+nQ_{02}-2n(n+3)Q_{10}+2nQ_{11}+nQ_{20})P_1^2+8(n+1+nQ_{00}-nQ_{01}-nQ_{10})N_2\epsilon_k^3 \\ & +\phi_m\epsilon_k^4+O(\epsilon_k^5), \end{aligned} \tag{27}$$

where $\phi_m = \phi_m(b, Q_{00}, Q_{10}, Q_{01}, Q_{20}, Q_{11}, Q_{02}, P_1, P_2, P_3)$ for $n = 4$ and $\phi_m = \phi_m(Q_{00}, Q_{10}, Q_{01}, Q_{20}, Q_{11}, Q_{02}, P_1, P_2, P_3)$ for $n \geq 5$.

If we set coefficients ϵ_k, ϵ_k^2 and ϵ_k^3 equal to zero and solve the resulting equations, we get:

$$Q_{00} = 0, \quad Q_{10} = \frac{n+1}{n} - Q_{01}, \quad Q_{20} = \frac{4n+4}{4} - Q_{02} - 2Q_{11}. \tag{28}$$

Adopt the expression (28) in (27), we have the following error equation:

$$\epsilon_{k+1} = \left(\frac{9+n}{2n^3}P_1^3 - \frac{1}{n^2}P_1P_2\right)\epsilon_k^4 + O(\epsilon_k^5). \tag{29}$$

Hence, the theorem is proved. □

Remark 1. The algorithms (3) reaches at fourth order convergence provided the conditions of Theorem 3 are satisfied. Only three evaluations of function, namely $\Phi(t_k), \Phi(u_k)$ and $\Phi(v_k)$, are used per iteration in order to achieve this convergence rate. Therefore, the Kung–Traub hypothesis [20] confirms the optimal convergence of our algorithm (3).

Remark 2. It is worth noting that b , which is employed in u_k , only exists in the error equations of the cases $n = 2$ and $n = 3$, but not in $n \geq 4$. However, for $n \geq 4$, we have noticed that it occurs in terms of ϵ_k^5 and higher order. In general, such terms are expensive to compute. Furthermore, these terms are not required to demonstrate the desired fourth-order convergence.

Some Special Cases

We have explored several cases based on the weight function $Q(x, y)$ which satisfy the conditions of Theorems 1–3. But, some the important simple forms are given below:

- (1) $Q_{(x_k, y_k)} = \frac{(4+3n)x_k + 8(1+n)x_k^2 + ny_k}{4n}$.
- (2) $Q_{(x_k, y_k)} = -\frac{(4+3n)^3x_k + ny_k(16+8n(3+y_k) + n^2(9+8y_k))}{4n(4+3n)(8x_k + n(8x_k - 3) - 4) - 32n^2(1+n)y_k}$.
- (3) $Q_{(x_k, y_k)} = \frac{(4+3n)^3x_k + ny_k(16+8n(3+y_k) + n^2(9+8y_k))}{-32n(4+7n+3n^2)x_k + (4+3n)^3x_k^2 + 4n(16+8n(3+y_k) + n^2(9+8y_k))}$.

The corresponding technique to each of the above forms can be expressed as follows:

Method 1 (M1):

$$t_{k+1} = v_k - n \left[\frac{(4+3n)x_k + 8(1+n)x_k^2 + ny_k}{4n} \right] \frac{\Phi(t_k)}{\Phi[u_k, t_k] + \Phi[v_k, u_k]}. \tag{30}$$

Method 2 (M2):

$$t_{k+1} = v_k + n \left[\frac{(4+3n)^3x_k + ny_k(16+8n(3+y_k) + n^2(9+8y_k))}{4n(4+3n)(8x_k + n(8x_k - 3) - 4) - 32n^2(1+n)y_k} \right] \frac{\Phi(t_k)}{\Phi[u_k, t_k] + \Phi[v_k, u_k]}. \tag{31}$$

Method 3 (M3):

$$t_{k+1} = v_k - n \left[\frac{(4 + 3n)^3 x_k + ny_k(16 + 8n(3 + y_k) + n^2(9 + 8y_k))}{-32n(4 + 7n + 3n^2)x_k + (4 + 3n)^3 x_k^2 + 4n(16 + 8n(3 + y_k) + n^2(9 + 8y_k))} \right] \times \frac{\Phi(t_k)}{\Phi[u_k, t_k] + \Phi[v_k, u_k]}. \tag{32}$$

4. Numerical Results

We choose the combinations of (30)–(32) with scheme (3) for $(b = -0.5)$, called by (M1a), (M2a) and (M3a), respectively. In addition, we again consider the combinations of (30)–(32) in the scheme (3) with $(b = -0.4)$, denoted by (M1b), (M2b) and (M3b), respectively. The examples not only illustrate the feasibility and effectiveness of our methods, but also confirm the theoretical aspects. In order to verify the computational order of convergence (COC), we use the following formula (see [21])

$$COC = \frac{\ln |(t_{k+2} - \alpha)/(t_{k+1} - \alpha)|}{\ln |(t_{k+1} - \alpha)/(t_k - \alpha)|}, \quad k = 1, 2, \dots \tag{33}$$

The performance of the new algorithms is compared with the following six known methods:

(i) *Li–Liao–Cheng method (LLC)* [5]:

$$v_k = t_k - \frac{2n}{n + 2} \frac{\Phi(t_k)}{\Phi'(t_k)},$$

$$t_{k+1} = t_k - \frac{n(n - 2) \left(\frac{n}{n+2}\right)^{-n} \Phi'(v_k) - n^2 \Phi'(t_k)}{\Phi'(t_k) - \left(\frac{n}{n+2}\right)^{-n} \Phi'(v_k)} \frac{\Phi(t_k)}{2\Phi'(t_k)}.$$

(ii) *Li–Cheng–Neta method (LCN)* [6]:

$$v_k = t_k - \frac{2n}{n + 2} \frac{\Phi(t_k)}{\Phi'(t_k)},$$

$$t_{k+1} = t_k - \alpha_1 \frac{\Phi(t_k)}{\Phi'(v_k)} - \frac{\Phi(t_k)}{\alpha_2 \Phi'(t_k) + \alpha_3 \Phi'(v_k)},$$

where

$$\alpha_1 = -\frac{1}{2} \frac{\left(\frac{n}{n+2}\right)^n n(n^4 + 4n^3 - 16n - 16)}{n^3 - 4n + 8},$$

$$\alpha_2 = -\frac{(n^3 - 4n + 8)^2}{n(n^4 + 4n^3 - 4n^2 - 16n + 16)(n^2 + 2n - 4)},$$

$$\alpha_3 = \frac{n^2(n^3 - 4n + 8)}{\left(\frac{n}{n+2}\right)^n (n^4 + 4n^3 - 4n^2 - 16n + 16)(n^2 + 2n - 4)}.$$

(iii) *Sharma–Sharma method (SSM)* [7]:

$$v_k = t_k - \frac{2n}{n + 2} \frac{\Phi(t_k)}{\Phi'(t_k)},$$

$$t_{k+1} = t_k - \frac{n}{8} \left[(n^3 - 4n + 8) - (n + 2)^2 \left(\frac{n}{n+2}\right)^n \frac{\Phi'(t_k)}{\Phi'(v_k)} \right. \\ \left. \times \left(2(n - 1) - (n + 2) \left(\frac{n}{n+2}\right)^n \frac{\Phi'(t_k)}{\Phi'(v_k)} \right) \right] \frac{\Phi(t_k)}{\Phi'(t_k)}.$$

(iv) Zhou–Chen–Song method (ZCS) [8]:

$$v_k = t_k - \frac{2n}{n+2} \frac{\Phi(t_k)}{\Phi'(t_k)},$$

$$t_{k+1} = t_k - \frac{n}{8} \left[n^3 \left(\frac{n+2}{n} \right)^{2n} \left(\frac{\Phi'(v_k)}{\Phi'(t_k)} \right)^2 - 2n^2(n+3) \left(\frac{n+2}{n} \right)^n \frac{\Phi'(v_k)}{\Phi'(t_k)} + (n^3 + 6n^2 + 8n + 8) \right] \frac{\Phi(t_k)}{\Phi'(t_k)}.$$

(v) Soleymani–Babajee–Lotfi method (SBM) [10]:

$$v_k = t_k - \frac{2n}{n+2} \frac{\Phi(t_k)}{\Phi'(t_k)},$$

$$t_{k+1} = t_k - \frac{\Phi'(v_k)\Phi(t_k)}{q_1(\Phi'(v_k))^2 + q_2\Phi'(v_k)\Phi'(t_k) + q_3(\Phi'(t_k))^2},$$

where

$$q_1 = \frac{1}{16} n^{3-n} (n+2)^n,$$

$$q_2 = \frac{8 - n(n+2)(n^2 - 2)}{8m},$$

$$q_3 = \frac{1}{16} (n-2)n^{n-1}(n+2)^{3-n}.$$

(vi) Kansal–Kanwar–Bhatia method (KKB) [13]:

$$v_k = t_k - \frac{2n}{n+2} \frac{\Phi(t_k)}{\Phi'(t_k)},$$

$$t_{k+1} = t_k - \frac{n}{4} \Phi(t_k) \left(1 + \frac{n^4 p^{-2n} \left(p^{n-1} - \frac{\Phi'(v_k)}{\Phi'(t_k)} \right)^2 (p^n - 1)}{8(2p^n + n(p^n - 1))} \right) \times \left(\frac{4 - 2n + n^2(p^{-n} - 1)}{\Phi'(t_k)} - \frac{p^{-n}(2p^n + n(p^n - 1))^2}{\Phi'(t_k) - \Phi'(v_k)} \right),$$

where $p = \frac{n}{n+2}$.

The calculations are performed with *Mathematica* [22] using multiple-precision arithmetic. We consider the five nonlinear problems for comparisons, which are depicted in the Table 1. The numerical results in Tables 2–7 include:

1. The multiplicity of the corresponding function.
2. The number of iterations (k) based on the stopping criterion $|t_{k+1} - t_k| + |\Phi(t_k)| < 10^{-100}$.
3. The first three estimated errors $|t_{k+1} - t_k|$ in the iterations.
4. The computational order of convergence (COC) using (33).
5. The CPU time required in the execution of program which is computed by the *Mathematica* command "TimeUsed[]".

The configurations of the used computer for the calculation work are given below:

Processor: Intel(R) Pentium(R) CPU B960@2.20GHz,
 Made: HP
 Installed memory (RAM): 4 GB RAM
 Window edition: Windows 7 Professional
 System type: 32-bit-Operating System.

Table 1. The considered five nonlinear problems for numerical experiments.

Problems	Root	Initial Guess
Van der Waals problem [23] $\Phi_1(t) = t^3 - 5.22t^2 + 9.0825t - 5.2675$	1.75	2.6
Academic problem: $\Phi_2(t) = -\frac{t^4}{12} + \frac{t^2}{2} + t + e^t(t - 3) + \sin(t) + 3$	0	0.1
Planck law radiation problem: [23] $\Phi_3(t) = (e^{-t} - 1 + \frac{t}{5})^3$	4.9651142317...	5.6
Manning problem for isentropic supersonic flow: [24] $\Phi_4(t) = [\tan^{-1}(\frac{\sqrt{5}}{2}) - \tan^{-1}(\sqrt{t^2 - 1}) + \sqrt{6}(\tan^{-1}(\sqrt{\frac{t^2 - 1}{6}}) - \tan^{-1}(\frac{1}{2}\sqrt{\frac{5}{6}})) - \frac{11}{63}]^4$	1.8411294068...	1.5
Complex root problem: $\Phi_5(t) = t(t^2 + 1)(2e^{t^2 + 1} + t^2 - 1) \cosh^3(\frac{\pi t}{2})$	i	1.1 i

The multiplicity of the above considered functions is calculated by the following formula:

$$n = \frac{t_k - t_0}{d_k - d_0'}$$

where $d_k = \frac{\Phi(t_k)}{g_k}$ and $g_k = \frac{\Phi(t_k + \Phi(t_k)) - \Phi(t_k)}{\Phi(t_k)}$. We applied this formula in our method M1 and obtained the multiplicity. The obtained results are depicted in the Table 2. Similarly, we can apply M2 and M3.

Table 2. Multiplicity of considered problems in Table 1.

Problems	Multiplicity
$\Phi_1(t)$	2
$\Phi_2(t)$	3
$\Phi_3(t)$	3
$\Phi_4(t)$	4
$\Phi_5(t)$	5

Table 3. Results of the methods for problem $\Phi_1(t)$.

Methods	k	$ t_2 - t_1 $	$ t_3 - t_2 $	$ t_4 - t_3 $	COC	CPU
LLC	6	1.03(-1)	9.79(-3)	4.38(-5)	4.000	0.0748
LCN	6	1.03(-1)	9.79(-3)	4.38(-5)	4.000	0.0932
SSM	6	1.05(-1)	1.04(-2)	5.73(-5)	4.000	0.0779
ZCM	6	1.09(-1)	1.19(-2)	9.96(-5)	4.000	0.0767
SBM	6	1.03(-1)	9.79(-3)	4.38(-5)	4.000	0.0781
KKB	6	1.02(-1)	9.33(-3)	3.26(-5)	4.000	0.0765
M1a	5	1.38(-3)	7.08(-8)	6.01(-25)	4.000	0.0621
M1b	6	3.72(-2)	1.87(-3)	2.21(-7)	4.000	0.0652
M2a	4	1.38(-3)	1.99(-8)	9.65(-28)	4.000	0.0597
M2b	5	3.79(-2)	1.10(-3)	8.24(-26)	4.000	0.0634
M3a	5	1.38(-3)	3.72(-8)	2.28(-26)	4.000	0.0643
M3b	6	3.76(-2)	1.46(-3)	4.52(-8)	4.000	0.0667

Table 4. Results of the methods for problem $\Phi_1(t)$.

Methods	k	$ t_2 - t_1 $	$ t_3 - t_2 $	$ t_4 - t_3 $	COC	CPU
LLC	3	9.72(−7)	1.13(−26)	0	4.000	0.7334
LCN	3	9.74(−7)	1.14(−26)	0	4.000	0.7481
SSM	3	9.78(−7)	1.17(−26)	0	4.000	0.8890
ZCM	3	9.88(−7)	1.23(−26)	0	4.000	0.8743
SBM	3	9.79(−7)	1.17(−26)	0	4.000	1.0292
KKB	3	9.58(−7)	1.05(−26)	0	4.000	0.7492
M1a	3	4.65(−7)	2.55(−28)	0	4.000	0.3272
M1b	3	4.90(−7)	3.42(−28)	0	4.000	0.3431
M2a	3	2.77(−7)	1.94(−29)	0	4.000	0.3743
M2b	3	3.15(−7)	3.67(−29)	0	4.000	0.3287
M3a	3	3.32(−7)	4.73(−29)	0	4.000	0.3590
M3b	3	3.66(−7)	7.80(−29)	0	4.000	0.3435

Table 5. Results of the methods for problem $\Phi_1(t)$.

Methods	k	$ t_2 - t_1 $	$ t_3 - t_2 $	$ t_4 - t_3 $	COC	CPU
LLC	4	7.50(−5)	2.55(−20)	3.43(−82)	4.000	0.8582
LCN	4	7.51(−5)	2.56(−20)	3.44(−82)	4.000	0.9523
SSM	4	7.51(−5)	2.56(−20)	3.48(−82)	4.000	1.1247
ZCS	4	7.52(−5)	2.58(−20)	3.57(−82)	4.000	1.1401
SBM	4	7.51(−5)	2.56(−20)	3.49(−82)	4.000	1.2872
KKB	4	7.49(−5)	2.53(−20)	3.31(−82)	4.000	0.9525
M1a	3	9.35(−6)	1.21(−24)	0	4.000	0.2527
M1b	3	9.49(−6)	1.26(−24)	0	4.000	0.2234
M2a	3	8.85(−6)	8.37(−25)	0	4.000	0.2812
M2b	3	8.91(−6)	8.35(−25)	0	4.000	0.2793
M3a	3	9.00(−6)	9.35(−25)	0	4.000	0.2199
M3b	3	9.08(−6)	9.43(−25)	0	4.000	0.2345

Table 6. Results of the methods for problem $\Phi_1(t)$.

Methods	k	$ t_2 - t_1 $	$ t_3 - t_2 $	$ t_4 - t_3 $	COC	CPU
LLC	4	1.07(−3)	1.14(−14)	1.46(−58)	4.000	1.6843
LCN	4	1.07(−3)	1.13(−14)	1.43(−58)	4.000	1.7481
SSM	4	1.07(−3)	1.12(−14)	1.35(−58)	4.000	1.7789
ZCS	4	1.07(−3)	1.10(−14)	1.23(−58)	4.000	1.8104
SBM	4	1.07(−3)	1.08(−14)	1.16(−58)	4.000	1.9812
KKB	4	1.07(−3)	1.19(−14)	1.82(−58)	4.000	1.8882
M1a	4	5.74(−5)	2.78(−20)	8.59(−81)	4.000	1.4823
M1b	4	3.53(−5)	2.22(−20)	3.48(−81)	4.000	1.1866
M2a	4	3.71(−5)	1.86(−20)	1.17(−81)	4.000	1.1854
M2b	4	3.51(−5)	1.49(−20)	4.83(−82)	4.000	1.2019
M3a	4	3.71(−5)	2.10(−20)	2.15(−81)	4.000	1.2124
M3b	4	3.51(−5)	1.68(−20)	8.79(−82)	4.000	1.2173

Table 7. Results of the methods for problem $\Phi_1(t)$.

Methods	k	$ t_2 - t_1 $	$ t_3 - t_2 $	$ t_4 - t_3 $	COC	CPU
LLC	4	2.15(−5)	7.98(−20)	1.50(−77)	4.000	1.3732
LCN	4	2.15(−5)	8.01(−20)	1.53(−77)	4.000	2.0124
SSM	4	2.16(−5)	8.08(−20)	1.59(−77)	4.000	1.9973
ZCS	4	2.16(−5)	8.19(−20)	1.68(−77)	4.000	1.9969
SBM	4	2.18(−5)	8.56(−20)	2.03(−77)	4.000	2.4342
KKB	4	2.14(−5)	7.62(−20)	1.23(−77)	4.000	1.9509
M1a	4	7.23(−6)	5.38(−22)	1.65(−86)	4.000	0.4219
M1b	4	7.93(−6)	7.79(−22)	7.25(−86)	4.000	0.4058
M2a	4	2.61(−6)	5.84(−24)	1.47(−94)	4.000	0.4215
M2b	4	3.90(−6)	2.92(−23)	9.23(−92)	4.000	0.4207
M3a	4	3.84(−6)	3.14(−23)	1.41(−91)	4.000	0.4058
M3b	4	4.97(−6)	8.83(−23)	8.80(−90)	4.000	0.4219

Remark 3. The numerical results in Tables 3–7 show that the proposed techniques exhibit the consistent convergence behavior. Our methods consume the same or a fewer number of iterations for the considered problems in comparison to the other mentioned methods. Tables 3–7 demonstrate that the estimated errors of the presented algorithms are less than other methods. Furthermore, our methods execute the results in the short span of time as compared to the existing ones.

5. Conclusions

This study proposed some optimal derivative-free numerical techniques for multiple zeros on nonlinear equations. The fourth order is investigated based on the standard hypotheses. The applicability of new techniques has been illustrated on five nonlinear equations that were converted from real-life situations. The performances of our methods were compared to other existing methods of identical order. The numerical results show that the new derivative-free algorithms are superior to the existing ones.

Author Contributions: All authors have contributed equally to the development of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. D-130-713-1443.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (D-130-713-1443). The authors, therefore, acknowledge with thanks DSR technical and financial support.

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

References

- Argyros, I.K. *Convergence and Applications of Newton-Type Iterations*; Springer: New York, NY, USA, 2008.
- Sahlan, M.N.; Afshari, H. Three new approaches for solving a class of strongly nonlinear two-point boundary value problems. *Bound. Value Probl.* **2021**, *2021*, 60. [\[CrossRef\]](#)
- Hansen, E.; Patrick, M. A family of root finding methods. *Numer. Math.* **1977**, *27*, 257–269. [\[CrossRef\]](#)
- Dong, C. A family of multipoint iterative functions for finding multiple roots of equations. *Int. J. Comput. Math.* **1987**, *21*, 363–367. [\[CrossRef\]](#)
- 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.
- 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. [\[CrossRef\]](#)

7. Sharma, J.R.; Sharma, R. Modified Jarratt method for computing multiple roots. *Appl. Math. Comput.* **2010**, *217*, 878–881. [[CrossRef](#)]
8. Zhou, X.; Chen, X.; Song, Y. Constructing higher-order methods for obtaining the multiple roots of nonlinear equations. *J. Comput. Math. Appl.* **2011**, *235*, 4199–4206. [[CrossRef](#)]
9. Sharifi, M.; Babajee, D.K.R.; Soleymani, F. Finding the solution of nonlinear equations by a class of optimal methods. *Comput. Math. Appl.* **2012**, *63*, 764–774. [[CrossRef](#)]
10. Soleymani, F.; Babajee, D.K.R.; Lotfi, T. On a numerical technique for finding multiple zeros and its dynamics. *J. Egypt. Math. Soc.* **2013**, *21*, 346–353. [[CrossRef](#)]
11. 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](#)]
12. Geum, Y.H.; Kim, Y.I.; Neta, B. Constructing a family of optimal eighth-order modified Newton-type multiple-zero finders along with the dynamics behind their purely imaginary extraneous fixed points. *J. Comp. Appl. Math.* **2018**, *333*, 131–156. [[CrossRef](#)]
13. Kansal, M.; Kanwar, V.; Bhatia, S. On some optimal multiple root-finding methods and their dynamics. *Appl. Appl. Math.* **2015**, *10*, 349–367.
14. Schröder, E. Über unendlich viele Algorithmen zur Auflösung der Gleichungen. *Math. Ann.* **1870**, *2*, 317–365. [[CrossRef](#)]
15. Traub, J.F. *Iterative Methods for the Solution of Equations*; Chelsea Publishing Company: New York, NY, USA, 1982.
16. Kumar, D.; Sharma, J.R.; Argyros, I.K. Optimal one-point iterative function free from derivatives for multiple roots. *Mathematics* **2020**, *8*, 709. [[CrossRef](#)]
17. Behl, R.; Bhalla, S.; Magreñán, Á.A.; Moysi, A. An Optimal Derivative Free Family of Chebyshev-Halley's Method for Multiple Zeros. *Mathematics* **2021**, *9*, 546. [[CrossRef](#)]
18. Kumar, S.; Kumar, D.; Sharma, J.R.; Jäntschi, L. A Family of Derivative Free Optimal Fourth Order Methods for Computing Multiple Roots. *Symmetry* **2020**, *12*, 1969. [[CrossRef](#)]
19. Kumar, S.; Kumar, D.; Sharma, J.R.; Argyros, I.K. An efficient class of fourth-order derivative-free method for multiple-roots. *Int. J. Nonlinear Sci. Numer. Simul.* **2021**, *2021*, 000010151520200161. [[CrossRef](#)]
20. Kung, H.T.; Traub, J.F. Optimal order of one-point and multipoint iteration. *J. Assoc. Comput. Mach.* **1974**, *21*, 643–651. [[CrossRef](#)]
21. Cordero, A.; Torregrosa, J.R. Variants of Newton's method using fifth-order quadrature formulas. *Appl. Math. Comput.* **2007**, *190*, 686–698. [[CrossRef](#)]
22. Wolfram, S. *The Mathematica Book*, 5th ed.; Wolfram Media: Champaign, IL, USA, 2003.
23. Bradie, B. *A Friendly Introduction to Numerical Analysis*; Pearson Education Inc.: New Delhi, India, 2006.
24. Hoffman, J.D. *Numerical Methods for Engineers and Scientists*; McGraw-Hill Book Company: New York, NY, USA, 1992.