mathematics

MDPI

Article

Efficient Fourth-Order Scheme for Multiple Zeros: Applications and Convergence Analysis in Real-Life and Academic Problems

Sunil Kumar [1], Ramandeep Behl [2,*] and Azizah Alrajhi [2]

1 Department of Mathematics, University Centre for Research and Development, Chandigarh University, Mohali 140413, India; sfageria1988@gmail.com
2 Mathematical Modelling and Applied Computation Research Group (MMAC), Department of Mathematics, King Abdulaziz University, Jeddah 21589, Saudi Arabia; aalrajhi@kau.edu.sa
* Correspondence: ramanbehl87@yahoo.in or rlal@kau.edu.sa

Abstract: High-order iterative techniques without derivatives for multiple roots have wide-ranging applications in the following: optimization tasks, where the objective function lacks explicit derivatives or is computationally expensive to evaluate; engineering; design finance; data science; and computational physics. The versatility and robustness of derivative-free fourth-order methods make them a valuable tool for tackling complex real-world optimization challenges. An optimal extension of the Traub–Steffensen technique for finding multiple roots is presented in this work. In contrast to past studies, the new expanded technique effectively handles functions with multiple zeros. In addition, a theorem is presented to analyze the convergence order of the proposed technique. We also examine the convergence analysis for four real-life problems, namely, Planck's law radiation, Van der Waals, the Manning equation for isentropic supersonic flow, the blood rheology model, and two well-known academic problems. The efficiency of the approach and its convergence behavior are studied, providing valuable insights for practical and academic applications.

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

MSC: 65G99; 65H10

Citation: Kumar, S.; Behl, R.; Alrajhi, A. Efficient Fourth-Order Scheme for Multiple Zeros: Applications and Convergence Analysis in Real-Life and Academic Problems. *Mathematics* **2023**, *11*, 3146. https://doi.org/10.3390/math11143146

Academic Editor: Andrey Amosov

Received: 28 May 2023
Revised: 3 July 2023
Accepted: 11 July 2023
Published: 17 July 2023

1. Introduction

Diverse areas of optimization and numerical analysis present obstacles for the development of derivative-free approaches. Traditional iterative techniques rely on higher-order derivatives, or only first-order derivatives in the case of multi-point classes, to guide the search for optimal solutions. However, due to the lack of formal mathematical formulations, deriving derivatives in real-world situations may be computationally expensive, impractical, or even impossible. This restriction makes it difficult for standard methodologies to be applied to complicated systems and real-world issues. Such difficulties are addressed by derivative-free approaches, which only rely on function evaluations. A considerable problem still exists in constructing such algorithms with great efficiency, convergence, and robustness. Due to this, novel strategies must be created that can successfully handle optimization issues without the use of explicit derivatives. The one-point modified Traub–Steffensen [1] approach is one among the most well-known derivative-free techniques for multiple roots, as indicated by

$$t_{k+1} = t_k - n\frac{\Phi(t_k)}{\Phi[u_k, t_k]}, k = 0, 1, 2, \ldots, \tag{1}$$

where $u_k = t_k + b\,\Phi(t_k)$, $b \in \mathbb{R} - \{0\}$, $\Phi[u_k, t_k] = \frac{\Phi(u_k) - \Phi(t_k)}{u_k - t_k}$, and n is the multiplicity of the root.

A system's stability can also be examined using multiple roots. Multiple roots in a dynamical system correlate to several equilibrium points, and researching the stability of these points can aid in understanding how the system behaves under various circumstances. In numerous domains, including optimization, system analysis, and stability analysis, the utilization of multiple roots of nonlinear equations can yield useful information. We can better comprehend the problem and create more effective remedies by identifying multiple roots.

For numerous zeros of nonlinear functions, various authors recently devised optimal and non-optimal (in terms of the Kung–Traub hypothesis; see [2]) non-derivative approaches [3–7].

In order to attain a high order of convergence with a minimal amount of function evaluations, a high-efficiency non-derivative technique for the multiple-roots method is proposed. On the basis of these ideas, we create a two-step derivative-free approach that converges to the fourth order. The proposed method is optimal in terms of the Kung–Traub hypothesis [2] since it only performs three function evaluations throughout each cycle.

We compare our new derivative method with existing methods of the same order, i.e., with a derivative [8–13] and without a derivative [3–5]. In the numerical section, new methods and existing methods are applied to real-life problems, i.e., Planck law radiation, Van der Waals, Manning for isentropic supersonic flow, the blood rheology model and two academics problems. The van der Waals equation is an equation of state that attempts to explain how the behavior of real gas molecules, which have finite sizes and interact with one another, differs from that of an ideal gas. The spectrum distribution of energy emitted by a black body at a specific temperature is described by the Planck law equation, a fundamental equation in quantum mechanics. It is used to explain the black body radiation that is seen and how temperature affects it. The flow rate of a fluid through a channel is determined using the empirical Manning equation in open channel flow. In order to calculate the fluid flow rate, the channel's shape, slope, and roughness are all taken into consideration. In isentropic supersonic flow, where the fluid is moving at a high velocity and the pressure waves it generates are moving at or above the speed of sound, this equation is specifically used. Models of blood rheology use mathematics to illustrate how blood moves through the circulatory system. These simulations take into account properties like viscosity, elasticity, and shear stress to comprehend the intricate dynamics of blood flow. Comparing the newly proposed method to the current iterative root-finding method, the convergence area is wider.

The outline of the paper is given as follows: Section 2 of the paper delves into the development of this proposed scheme and provides three essential theorems, each specifically designed for $n = 2, n = 3$, and $n = 4$. These theorems establish the foundation for the subsequent sections. In Section 3, the main general theorem for the convergence order is unveiled, elucidating the scheme's efficiency. Moreover, Section 4 expounds on the authors' numerical experiments, where they rigorously test the scheme's efficacy on both real-life and academic problems. Finally, Section 5 offers concluding remarks, summarizing the paper's contributions and outlining potential future research directions.

2. Development of Scheme

Consider the following iterative approach for $n > 1$,

$$\begin{aligned} v_k &= t_k - n\frac{\Phi(t_k)}{\Phi[u_k, t_k]} \\ t_{k+1} &= v_k - n\frac{w_k}{d_1 + d_2 w_k}\frac{\Phi(t_k)}{d_3\Phi[v_k, t_k] + d_4\Phi[v_k, u_k]}, \end{aligned} \tag{2}$$

where d_1, d_2, d_3 and d_4 are the unknown parameters with $w_k = \sqrt[n]{\frac{\Phi(v_k)}{\Phi(t_k)}}$.

In order to establish the validity of our result, we conducted rigorous testing by varying the value of n across multiple instances. Through extensive experimentation and analysis, we consistently observed the same outcome, thus providing compelling evidence for the generalizability and reliability of our findings across different values of n. We first examine the situation $n = 2$ and demonstrate the validity of the following theorem.

Theorem 1. *With $n = 2$ of Φ, let's assume that $t = alpha$ is a multiple solution. We presume $\Phi : D \subset \mathbb{C} \to \mathbb{C}$ is an analytic function in D vicinity of the zero α. So, the algorithm (2) has convergence order four, when $d_1 = \frac{2}{d_3+d_4}$, $d_2 = -\frac{4}{d_3+d_4}$ and $d_3 = d_4$.*

Proof. $\epsilon_k = t_k - \alpha$ is used to indicate an error at the k-th stage. As a result of Taylor's expansion of $\Phi(t_k)$ about α, we yield

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

where $N_m = \frac{2!}{(2+m)!}\frac{\Phi^{(2+m)}(\alpha)}{\Phi^{(2)}(\alpha)}$ for $m \in \mathbb{N}$, $\Phi(\alpha) = 0$, $\Phi'(\alpha) = 0$ and $\Phi^{(2)}(\alpha) \neq 0$.

Similarly, we have that $\Phi(u_k)$ about α

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

where $\epsilon_{u_k} = u_k - \alpha = \epsilon_k + \frac{\Delta_2}{2!}\epsilon_k^2\big(1 + N_1\epsilon_k + N_2\epsilon_k^2 + N_3\epsilon_k^3 + \cdots\big)$ and $\Delta_2 = b\Phi^{(2)}(\alpha)$.

From the first step of (2), we obtain

$$\begin{aligned}\epsilon_{v_k} &= v_k - \alpha \\ &= \frac{1}{2}\Big(\frac{\Delta_2}{2} + N_1\Big)\epsilon_k^2 - \frac{1}{16}(\Delta_2^2 - 8\Delta_2N_1 + 12N_1^2 - 16N_2)\epsilon_k^3 + O(\epsilon_k^4).\end{aligned} \tag{5}$$

Expanding $\Phi(v_k)$ around α, it yields

$$\Phi(v_k) = \frac{\Phi^{(2)}(\alpha)}{2!}\epsilon_{w_k}^2\big(1 + N_1\epsilon_{w_k} + N_2\epsilon_{w_k}^2 + N_3\epsilon_{w_k}^3 + \cdots\big). \tag{6}$$

Using (3) and (6), we have

$$\begin{aligned}w_k = {}& \frac{1}{2}\Big(\frac{\Delta_2}{2} + N_1\Big)\epsilon_k - \frac{1}{16}\big(\Delta_2^2 - 6b\Phi^{(2)}(\alpha)N_1 + 16(N_1^2 - N_2)\big)\epsilon_k^2 \\ & + \frac{1}{64}\big(\Delta_2^3 - 22\Delta_2N_1^2 + 4(29N_1^3 + 14\Delta_2N_2) \\ & - 2N_1(3\Delta_2^2 + 104N_2) + 96N_3\big)\epsilon_k^3 + O(\epsilon_k^4).\end{aligned} \tag{7}$$

Inserting (3), (4), (6) and (7) in the last step of (2), and after some calculations, yields

$$\begin{aligned}\epsilon_{k+1} = {}& \frac{(d_3d_1 + d_1d_4 - 2)}{4d_1(d_3 + d_4)}(\Delta_2 + 2N_1)\epsilon_k^2 + \frac{1}{16d_1^2(d_3 + d_4)^2}\Big[\Delta_2^2\Big(-(d_1d_3)^2 \\ & + d_3(4d_1 + 2d_2 - 2d_1^2d_4) + d_4(8d_1 + 2d_2 - d_1^2d_4)\Big) + 4\Delta_2\Big(2(d_3d_1)^2 \\ & + d_4(d_1 + 2d_2 + 2d_1^2d_4) + d_3(2d_2 - d_1 + 4d_1^2d_4)\Big)N_1 - 4(d_3 + d_4)\Big(3d_1^2(d_3 + d_4) \\ & - 10d_1 - 2d_2\Big)N_1^2 + 16d_1(d_3 + d_4)(d_3d_1 + d_1d_4 - 2)N_2\Big]\epsilon_k^3 + \psi_m\epsilon_k^4 + O(\epsilon_k^5),\end{aligned}$$

where

$$
\begin{aligned}
\psi_m &= \psi_m(b, d_1, d_2, d_3, d_4, N_1, N_2, N_3), \\
&= -\frac{1}{64 d_1^3 (d_3+d_4)^3} \Big[8 d_1^2 d_3^2 \Delta_2^3 + 6 d_1 d_2 d_3^2 \Delta_2^3 + 2 d_2^2 d_3^2 \Delta_2^3 - d_1^3 d_3^3 \Delta_2^3 + 28 d_1^2 d_3 d_4 \Delta_2^3 \\
&\quad + 16 d_1 d_2 d_3 d_4 \Delta_2^3 + 4 d_2^2 d_3 d_4 \Delta_2^3 - 3 d_1^3 d_3^2 d_4 \Delta_2^3 + 28 d_1^2 d_4^2 \Delta_2^3 + 10 d_1 d_2 d_4^2 \Delta_2^3 \\
&\quad + 2 d_2^2 d_4^2 \Delta_2^3 - 3 d_1^3 d_3 d_4^2 \Delta_2^3 - d_1^3 d_4^3 \Delta_2^3 + 4(d_3+d_4)\Big(6 d_2^2 (d_3+d_4) \\
&\quad + 5 d_1^3 (d_3+d_4)^2 + d_1^2(-5 d_3 + 3 d_4) + 2 d_1 d_2 (5 d_3 + 7 d_4)\Big)\Delta_2 N_1^2 \\
&\quad - 8(d_3+d_4)^2(-45 d_1^2 - 18 d_1 d_2 - 2 d_2^2 + 9 d_1^3 (d_3+d_4)) N_1^3 \\
&\quad - 16 d_1 (d_3+d_4)(d_1(-3 d_3 + d_4) + 4 d_2 (d_3+d_4) + 4 d_1^2 (d_3+d_4)^2)\Delta_2 N_2 \\
&\quad + 2 N_1 \Big\{ (6 d_2^2 (d_3+d_4)^2 + 5 d_1^3 (d_3+d_4)^3 - 2 d_1 d_2 (d_3^2 - 2 d_3 d_4 - 3 d_4^2) \\
&\quad - 2 d_1^2 (5 d_3^2 + 16 d_3 d_4 + 7 d_4^2))\Delta_2^2 - 16 d_1 (d_3+d_4)^2 \Big(17 d_1 + 4 d_2 \\
&\quad - 5 d_1^2 (d_3+d_4)\Big) N_2 \Big\} + 192 d_1^2 d_3^2 N_3 - 96 d_1^3 d_3^3 N_3 + 384 d_1^2 d_3 d_4 N_3 \\
&\quad - 288 d_1^3 d_3^2 d_4 N_3 + 192 d_1^2 d_4^2 N_3 - 288 d_1^3 d_3 d_4^2 N_3 - 96 d_1^3 d_4^3 N_3 \Big].
\end{aligned}
$$

Now, fixing the coefficients of ϵ_k^2 and ϵ_k^3 to zero, we obtain

$$d_1 = \frac{2}{d_3+d_4}, \quad d_2 = -\frac{4}{d_3+d_4}, \quad d_3 = d_4. \tag{8}$$

Now, by using (8) in (2), we have

$$\epsilon_{k+1} = -\frac{1}{64}(\Delta_2 + 2N_1)\Big(3\Delta_2^2 + 10\Delta_2 N_1 - 2N_1^2 + 8N_2\Big)\epsilon_k^4 + O(\epsilon_k^5).$$

□

Theorem 2. *If Theorem 1 is adopted, then the algorithm (2) for $n = 3$ has at least an order of four of convergence if $d_1 = \frac{3}{d_3+d_4}$ and $d_2 = -\frac{6}{d_3+d_4}$.*

Proof. Considering that $\Phi(\alpha) = \Phi'(\alpha) = \Phi^{(2)}(\alpha) = 0$ and $\Phi^{(3)}(\alpha) \neq 0$, expanding $\Phi(t_k)$ around α by Taylor series gives

$$\Phi(t_k) = \frac{\Phi^{(3)}(\alpha)}{3!}\epsilon_k^3 (1 + \bar{N}_1 \epsilon_k + \bar{N}_2 \epsilon_k^2 + \bar{N}_3 \epsilon_k^3 + \bar{N}_4 \epsilon_k^4 + \cdots), \tag{9}$$

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

Similarly, expanding $\Phi(u_k)$ about α yields

$$\Phi(u_k) = \frac{\Phi^{(3)}(\alpha)}{3!}\epsilon_{u_k}^3 (1 + \bar{N}_1 \epsilon_{u_k} + \bar{N}_2 \epsilon_{u_k}^2 + \bar{N}_3 \epsilon_{u_k}^3 + \bar{N}_4 \epsilon_{u_k}^4 + \cdots), \tag{10}$$

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

Inserting (9) and (10) in (2), yields

$$\sigma_{v_k} = v_k - \alpha$$
$$= \frac{\bar{N}_1}{3}\epsilon_k^2 + \frac{1}{18}\Big(3b\Phi^{(3)}(\alpha) - 8\bar{N}_1^2 + 12\bar{N}_2\Big)\epsilon_k^3 + \Big(\frac{16}{27}\bar{N}_1^3 + \frac{\bar{N}_1}{9}(2b\Phi^{(3)}(\alpha) - 13\bar{N}_2) \tag{11}$$
$$+ \bar{N}_3\Big)\epsilon_k^4 + O(\epsilon_k^5). \tag{12}$$

In a similar fashion, expanding $\Phi(v_k)$ about α, we have

$$\Phi(v_k) = \frac{\Phi^{(3)}(\alpha)}{3!}\epsilon_{v_k}^3\big(1 + \bar{N}_1\epsilon_{v_k} + \bar{N}_2\epsilon_{v_k}^2 + \bar{N}_3\epsilon_{v_k}^3 + \bar{N}_4\epsilon_{v_k}^4 + \cdots\big). \tag{13}$$

From (9) and (13), we have

$$\begin{aligned} w_k = {} & \frac{\bar{N}_1}{3}\epsilon_k + \Big(\frac{b\Phi^{(3)}(\alpha)}{6} - \frac{5}{9}\bar{N}_1^2 + \frac{2}{3}\bar{N}_2\Big)\epsilon_k^2 \\ & + \Big(\frac{23}{27}\bar{N}_1^3 + \frac{\bar{N}_1}{18}(3b\Phi^{(3)}(\alpha) - 32\bar{N}_2) + \bar{N}_3\Big)\epsilon_k^3 + O(\epsilon_k^4). \end{aligned} \tag{14}$$

By using (9)–(14) in the last step of (2), we obtain

$$\begin{aligned} \epsilon_{k+1} = {} & \frac{1}{3}\Big(1 - \frac{3}{d_1(d_3+d_4)}\Big)\bar{N}_1\epsilon_k^2 + \frac{1}{18d_1^2(d_3+d_4)}\Big((36d_1 + 6d_2 - 8d_1^2(d_3+d_4))\bar{N}_1^2 \\ & + 3d_1(-3 + d_3d_1 + d_1d_4)(b\Phi^{(3)}(\alpha) + 4\bar{N}_2)\Big)\epsilon_k^3 + \varphi_m\epsilon_k^4 + O(\epsilon_k^5), \end{aligned} \tag{15}$$

where $\varphi_m = \varphi_m(b, d_1, d_2, d_3, d_4, \bar{N}_1, \bar{N}_2, \bar{N}_3)$.

The equations can now be solved by setting the coefficients of ϵ_k^2 and ϵ_k^3 to zero. We have

$$d_1 = \frac{3}{d_3 + d_4}, \quad d_2 = -\frac{6}{d_3 + d_4}. \tag{16}$$

The error Equation (15) is given by

$$\epsilon_{k+1} = \frac{\bar{N}_1}{54(d_3+d_4)}\Big(4(d_3+d_4)\bar{N}_1^2 - 3(b\Phi^{(3)}(\alpha)(d_3-d_4) + 2(d_3+d_4)\bar{N}_2)\Big)\epsilon_k^4 + O(\epsilon_k^5).$$

Hence, Theorem 2 is proved. □

Now, we state the theorem for $n = 4$.

Theorem 3. *If Theorem 1 is adopted, then the algorithm (2) for $n = 4$ has at least 4th order of convergence if $d_1 = \frac{4}{d_3+d_4}$ and $d_2 = -\frac{8}{d_3+d_4}$. Furthermore, the error equation for (2) is provided by*

$$\epsilon_{k+1} = \Big(\frac{5}{128}\bar{\bar{N}}_1^3 - \frac{1}{16}\bar{\bar{N}}_1\bar{\bar{N}}_2\Big)\epsilon_k^4 + O(\epsilon_k^5),$$

where $\bar{\bar{N}}_m = \frac{4!}{(4+m)!}\frac{\Phi^{(4+m)}(\alpha)}{\Phi^{(4)}(\alpha)}$ for $m \in \mathbb{N}$.

3. Generalization of the Method

Theorem 4. *If Theorem 1 is adopted, then the algorithm (2) for $n \geq 4$ has at least an order of four of convergence, if $d_1 = \frac{n}{d_3+d_4}$ and $d_2 = -\frac{2n}{d_3+d_4}$. Moreover, the error equation of (2) is given by*

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

Proof. First, we expand $\Phi(t_k)$ about α, and we have

$$\Phi(t_k) = \frac{\Phi^n(\alpha)}{n!}\epsilon_k^n\big(1 + P_1\epsilon_k + P_2\epsilon_k^2 + P_3\epsilon_k^3 + P_4\epsilon_k^4 + \cdots\big), \tag{17}$$

where $P_m = \frac{n!}{(m+n)!}\frac{\Phi^{(m+n)}(\alpha)}{\Phi^{(n)}(\alpha)}$ for $m \in \mathbb{N}$ with $\Phi^{(j)}(\alpha) = 0, j = 0, 1, 2, ..., n-1$, and that $\Phi^n(\alpha) \neq 0$,.

Similarly, expanding $\Phi(u_k)$ about α leads to

$$\Phi(u_k) = \frac{\Phi^n(\alpha)}{n!}\epsilon_{u_k}^n\big(1 + P_1\epsilon_{u_k} + P_2\epsilon_{u_k}^2 + P_3\epsilon_{u_k}^3 + P_4\epsilon_{u_k}^4 + \cdots\big), \tag{18}$$

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

From the first step of Equation (2) results

$$\begin{aligned}\epsilon_{v_k} &= v_k - \alpha \\ &= \frac{P_1}{n}\epsilon_k^2 + \frac{1}{n^2}\big(2nP_2 - (1+n)P_1^2\big)\epsilon_k^3 + \frac{1}{n^3}\big((1+n)^2P_1^3 - n(4+3n)P_1P_2 + 3n^2P_3\big)\epsilon_k^4 + O(\epsilon_k^5).\end{aligned} \tag{19}$$

Expanding $\Phi(v_k)$ around α further yields

$$\Phi(v_k) = \frac{\Phi^n(\alpha)}{n!}\epsilon_{v_k}^n\big(1 + P_1\epsilon_{v_k} + P_2\epsilon_{v_k}^2 + P_3\epsilon_{v_k}^3 + P_4\epsilon_{v_k}^4 + \cdots\big). \tag{20}$$

Using (17) and (20) in the expressions of w_k, we have that

$$\begin{aligned}w_k &= \frac{P_1}{n}\epsilon_k + \frac{1}{n^2}\big(2nP_2 - (2+n)P_1^2\big)\epsilon_k^2 \\ &+ \frac{1}{2n^3}\big((7+7n+2n^2)P_1^3 - 2n(7+3n)P_1P_2 + 6n^2P_3\big)\epsilon_k^3 + O(\epsilon_k^4).\end{aligned} \tag{21}$$

Inserting (17)–(21) in the second step of (2) gives

$$\begin{aligned}\epsilon_{k+1} &= \frac{1}{n}\Big(1 - \frac{n}{d_1d_3 + d_1d_4}\Big)P_1\epsilon_k^2 + \frac{1}{n^2d_1^2(d_3+d_4)}\big(\big((3n+n^2)d_1 + nd_2 - (n+1)d_1^2(d_3+d_4)\big)P_1^2 \\ &+ 2nd_1(-n + d_1(d_3+d_4))P_2\big)\epsilon_k^3 + \phi_m\epsilon_k^4 + O(\epsilon_k^5),\end{aligned} \tag{22}$$

where $\phi_m = \phi_m(n, d_1, d_2, d_3, d_4, P_1, P_2, P_3)$.

Again, fixing the coefficients of ϵ_k^2 and ϵ_k^3 equal to zero, we obtain

$$d_1 = \frac{n}{d_3 + d_4} \quad d_2 = -\frac{2n}{d_3 + d_4}. \tag{23}$$

The error Equation (22) is given by

$$\epsilon_{k+1} = \Big(\frac{1+n}{2n^3}P_1^3 - \frac{1}{n^2}P_1P_2\Big)\epsilon_k^4 + O(\epsilon_k^5). \tag{24}$$

Thus, the theorem is proved. □

Remark 1. *Assuming that the requirements of Theorem 4 are met, the algorithm (2) approaches fourth convergence order. To accomplish this convergence rate, only three functional evaluations $\Phi(t_k)$, $\Phi(u_k)$ and $\Phi(v_k)$, are utilized per iteration. As a result, the algorithm (2) is optimal [2].*

Remark 2. *It is crucial to keep in mind that the variable b, which is utilized in u_k, only appears when $n = 2$ and $n = 3$, rather than in the scenario where $n \geq 4$. However, we discovered that it does so for $n \geq 4$ in terms of ϵ_k^5 and higher order. Calculating such terms is often expensive. Furthermore, the requisite fourth-order convergence need not be shown using these terms.*

Remark 3. *The algorithm* (2) *is written as follows for future reference:*

$$
\begin{aligned}
v_k &= t_k - n\frac{\Phi(t_k)}{\Phi[u_k, t_k]} \\
t_{k+1} &= v_k - \frac{2w_k}{1-2w_k}\frac{\Phi(t_k)}{\Phi[v_k, t_k] + \Phi[v_k, u_k]}.
\end{aligned} \tag{25}
$$

For the numeric examples, the scheme (25) *is known as NM.*

4. Numerical Results

To resolve some nonlinear equations, we use the NM with $(b = 0.01)$. The examples support the theoretical findings while also demonstrating the method's viability and effectiveness. In the continuation, we use the formula (see [14])

$$
\mathrm{COC} = \frac{\ln|(t_{k+2}-\alpha)/(t_{k+1}-\alpha)|}{\ln|(t_{k+1}-\alpha)/(t_k-\alpha)|}, \quad k \ = 1, 2, \ldots \tag{26}
$$

to compute the computational order of convergence. The new algorithm's performance is compared to that of the nine existing methods:

(i) *Li et al.* [9] *iteration function* (LLC):

$$
\begin{aligned}
v_\tau &= \sigma_\tau - \frac{2\mu}{\mu+2}\frac{\Phi(\sigma_\tau)}{\Phi'(\sigma_\tau)}, \\
\sigma_{\tau+1} &= \sigma_\tau - \frac{\mu(\mu-2)\left(\frac{\mu}{\mu+2}\right)^{-\mu}\Phi'(v_\tau) - \mu^2\Phi'(\sigma_\tau)}{\Phi'(\sigma_\tau) - \left(\frac{\mu}{\mu+2}\right)^{-\mu}\Phi'(v_\tau)}\frac{\Phi(\sigma_\tau)}{2\Phi'(\sigma_\tau)}.
\end{aligned}
$$

(ii) *Li et al.* [10] *iteration function* (LCN):

$$
\begin{aligned}
v_\tau &= \sigma_\tau - \frac{2\mu}{\mu+2}\frac{\Phi(\sigma_\tau)}{\Phi'(\sigma_\tau)}, \\
\sigma_{\tau+1} &= \sigma_\tau - \alpha_1\frac{\Phi(\sigma_\tau)}{\Phi'(v_\tau)} - \frac{\Phi(\sigma_\tau)}{\alpha_2\Phi'(\sigma_\tau) + \alpha_3\Phi'(v_\tau)},
\end{aligned}
$$

where

$$
\begin{aligned}
\alpha_1 &= -\frac{1}{2}\frac{\left(\frac{\mu}{\mu+2}\right)^{\mu}\mu(\mu^4+4\mu^3-16\mu-16)}{\mu^3-4\mu+8}, \\
\alpha_2 &= -\frac{(\mu^3-4\mu+8)^2}{\mu(\mu^4+4\mu^3-4\mu^2-16\mu+16)(\mu^2+2\mu-4)}, \\
\alpha_3 &= \frac{\mu^2(\mu^3-4\mu+8)}{\left(\frac{\mu}{\mu+2}\right)^{\mu}(\mu^4+4\mu^3-4\mu^2-16\mu+16)(\mu^2+2\mu-4)}.
\end{aligned}
$$

(iii) *Sharma and Sharma* [11] *iteration function* (SSM):

$$
\begin{aligned}
v_\tau &= \sigma_\tau - \frac{2\mu}{\mu+2}\frac{\Phi(\sigma_\tau)}{\Phi'(\sigma_\tau)}, \\
\sigma_{\tau+1} &= \sigma_\tau - \frac{\mu}{8}\left[(\mu^3-4\mu+8) - (\mu+2)^2\left(\frac{\mu}{\mu+2}\right)^{\mu}\frac{\Phi'(\sigma_\tau)}{\Phi'(v_\tau)}\right. \\
&\quad \left.\times\left(2(\mu-1) - (\mu+2)\left(\frac{\mu}{\mu+2}\right)^{\mu}\frac{\Phi'(\sigma_\tau)}{\Phi'(v_\tau)}\right)\right]\frac{\Phi(\sigma_\tau)}{\Phi'(\sigma_\tau)}.
\end{aligned}
$$

(iv) *Iteration function from Zhou et al.* [13] (ZCS):

$$\begin{aligned} v_\tau &= \sigma_\tau - \frac{2\mu}{\mu+2}\frac{\Phi(\sigma_\tau)}{\Phi'(\sigma_\tau)}, \\ \sigma_{\tau+1} &= \sigma_\tau - \frac{\mu}{8}\Big[\mu^3\Big(\frac{\mu+2}{\mu}\Big)^{2\mu}\Big(\frac{\Phi'(v_\tau)}{\Phi'(\sigma_\tau)}\Big)^2 - 2\mu^2(\mu+3)\Big(\frac{\mu+2}{\mu}\Big)^{\mu}\frac{\Phi'(v_\tau)}{\Phi'(\sigma_\tau)} \\ &\qquad + (\mu^3+6\mu^2+8\mu+8)\Big]\frac{\Phi(\sigma_\tau)}{\Phi'(\sigma_\tau)}. \end{aligned}$$

(v) *Iteration function from Soleymani et al.* [12] (SBM):

$$\begin{aligned} v_\tau &= \sigma_\tau - \frac{2\mu}{\mu+2}\frac{\Phi(\sigma_\tau)}{\Phi'(\sigma_\tau)}, \\ \sigma_{\tau+1} &= \sigma_\tau - \frac{\Phi'(v_\tau)\Phi(\sigma_\tau)}{q_1(\Phi'(v_\tau))^2 + q_2\Phi'(v_\tau)\Phi'(\sigma_\tau) + q_3(\Phi'(\sigma_\tau))^2}, \end{aligned}$$

where

$$\begin{aligned} q_1 &= \frac{1}{16}\mu^{3-\mu}(\mu+2)^{\mu}, \\ q_2 &= \frac{8-\mu(\mu+2)(\mu^2-2)}{8m}, \\ q_3 &= \frac{1}{16}(\mu-2)\mu^{\mu-1}(\mu+2)^{3-\mu}. \end{aligned}$$

(vi) *Iteration function from Kansal et al.* [8] (KKB):

$$\begin{aligned} v_\tau &= \sigma_\tau - \frac{2\mu}{\mu+2}\frac{\Phi(\sigma_\tau)}{\Phi'(\sigma_\tau)}, \\ \sigma_{\tau+1} &= \sigma_\tau - \frac{\mu}{4}\Phi(\sigma_\tau)\left(1 + \frac{\mu^4 p^{-2\mu}\Big(p^{\mu-1} - \frac{\Phi'(v_\tau)}{\Phi'(\sigma_\tau)}\Big)^2(p^\mu-1)}{8(2p^\mu + \mu(p^\mu-1))}\right) \\ &\qquad \times\Big(\frac{4-2\mu+\mu^2(p^{-\mu}-1)}{\Phi'(\sigma_\tau)} - \frac{p^{-\mu}(2p^\mu+\mu(p^\mu-1))^2}{\Phi'(\sigma_\tau)-\Phi'(v_\tau)}\Big), \end{aligned}$$

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

(vii) *Iteration function from Sharma et al.* [3] (SKJ):

$$\begin{aligned} v_\tau &= \sigma_\tau - \mu\frac{\Phi(\sigma_\tau)}{\Phi[u_\tau,\sigma_\tau]}, \\ \sigma_{\tau+1} &= v_\tau - \big(w_\tau + \mu\, w_\tau^2 + (\mu-1)x_\tau + \mu\, w_\tau\, x_\tau\big)\frac{\Phi(\sigma_\tau)}{\Phi[u_\tau,\sigma_\tau]}. \end{aligned}$$

(viii) *Iteration function from Behl et al.* [4] (BAM):

$$\begin{aligned} v_\tau &= \sigma_\tau - \mu\frac{\Phi(\sigma_\tau)}{\Phi[u_\tau,\sigma_\tau]}, \\ \sigma_{\tau+1} &= v_\tau - \mu\frac{w_\tau + x_\tau}{2(1-2w_\tau)}\frac{\Phi(\sigma_\tau)}{\Phi[u_\tau,\sigma_\tau]}, \end{aligned}$$

where $x_\tau = \sqrt[\mu]{\frac{\Phi(v_\tau)}{\Phi(u_\tau)}}$.

(ix) *Iteration function from Kumar et al.* [5] (KKS):

$$v_\tau = \sigma_\tau - \mu \frac{\Phi(\sigma_\tau)}{\Phi[u_\tau, \sigma_\tau]},$$
$$\sigma_{\tau+1} = v_\tau - \frac{(\mu+2)w_\tau}{1-2w_\tau} \frac{\Phi(\sigma_\tau)}{\Phi[u_\tau, \sigma_\tau] + \Phi[v_\tau, u_\tau]}.$$

The computations are made using multiple-precision arithmetic in *Mathematica* [15]. The multiplicities of the considered functions are demonstrated in Table 1. In addition, Tables 2–7 contain the following points:

1. The multiplicity n of the relevant function.
2. The number of iterations (k) mentioned on the basis of stopping criteria $|t_{k+1} - t_k| + |\Phi(t_k)| \leq 10^{-100}$.
3. The first three estimated errors $|t_{k+1} - t_k|$ of the iterative methods are recorded.
4. Utilize (26) in order to compute the COC.
5. The amount of CPU time required to run a program is determined by the *Mathematica* command "TimeUsed[ ]". Figures 1–6 display the CPU timing consumed by our and the existing methods.

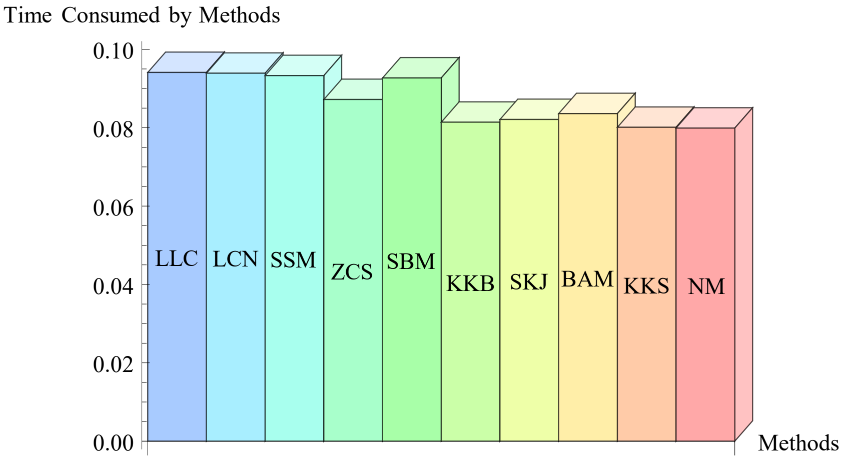

Figure 1. Bar chart of problem $\Phi_1(t)$.

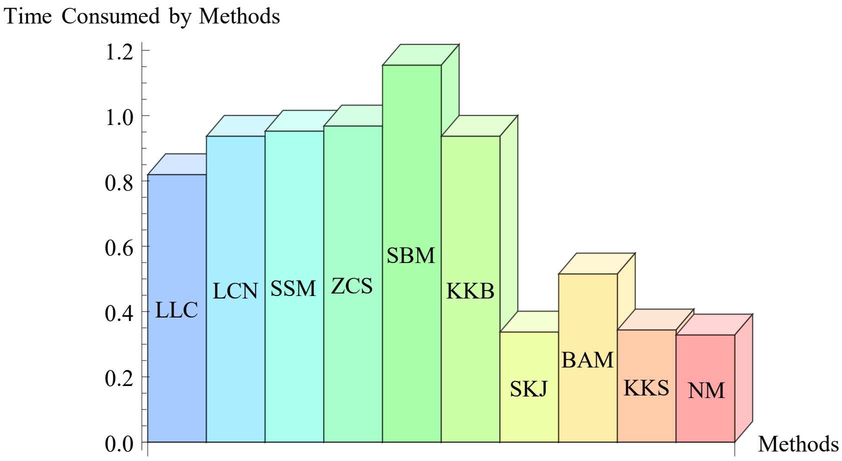

Figure 2. Bar chart of problem $\Phi_2(t)$.

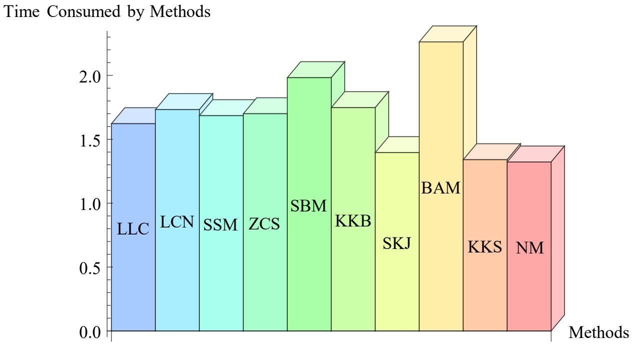

Figure 3. Bar chart of problem $\Phi_3(t)$.

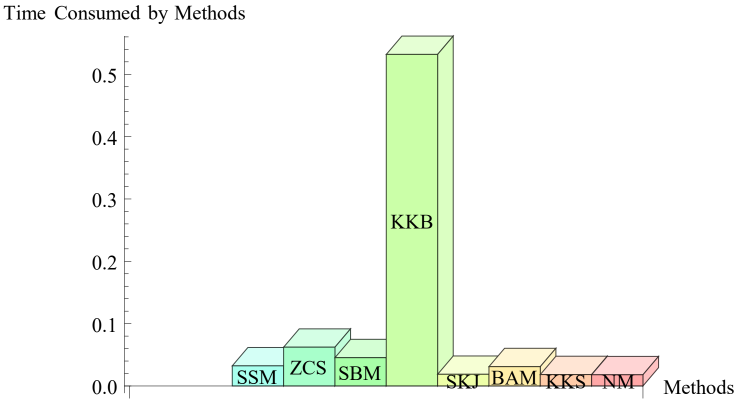

Figure 4. Bar chart of problem $\Phi_4(t)$.

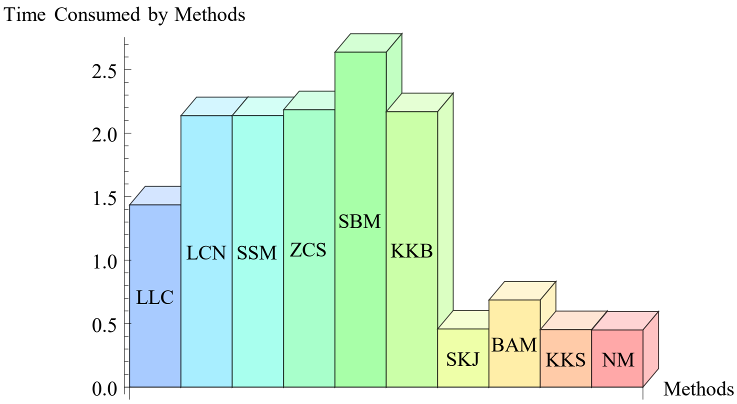

Figure 5. Bar chart of problem $\Phi_5(t)$.

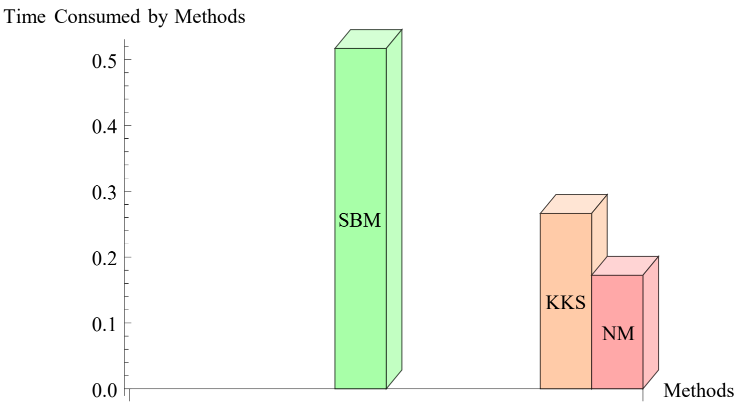

Figure 6. Bar chart of problem $\Phi_6(t)$.

Example 1. *Van der Waals equation* [16]*: The Van der Waals equation is a non-differential equation that models the behavior of real gases, taking into account intermolecular forces and the finite size of gas molecules. Compared to the ideal gas law, it offers a more accurate explanation of gas behavior. The equation introduces correction terms for attractive forces and molecular volume. The Van der Waals equation finds applications in thermodynamics, chemical engineering, and material science, enabling the study of real gas behavior and phase transitions under various conditions:*

$$\left(P+\frac{a_1n^2}{V^2}\right)(V-na_2)=nRT, \tag{27}$$

where we have the following:
R: universal gas constant;
V: volume;
P: pressure;
T: temperature.

To calculate the value of V, we can write (27) as

$$PV^3-(na_2P+nRT)V^2+a_1n^2V-a_1a_2n^2=0. \tag{28}$$

There are values of n, P, R, T, a_1 and a_2 of a particular gas. In this way, the expression (28) has three solutions. So by using a specific set of values, we have

$$\Phi_1(t)=t^3+9.0825t-5.22t^2-5.2675.$$

It has the following three roots: 1.72, 1.75, and 1.75. The desired root is therefore $alpha=1.75$. The approaches are evaluated using the initial estimate of $t_0=2.3$. Table 2 presents the computed results.

Example 2. *Planck's radiation law problem* [16]*: The mathematical equation for Planck's law of radiation is given by*

$$\varphi(\lambda)=B(\lambda,T)=\frac{2hc^2}{\lambda^5}*\frac{1}{e^{hc/\lambda*kT}-1}, \tag{29}$$

where we have the following:
$B(\lambda,T)$ represents the spectral radiance or energy density per unit wavelength at a given wavelength λ and temperature T.
k and h stand for the Boltzmann and Planck constants, respectively.
c is the speed of light in vacuum.
λ is the wavelength of radiation.
T stand for temperature in Kelvin.

The equation expresses the spectral distribution of radiation emitted by a black body at a specific temperature T across different wavelengths λ. It shows how the radiance or intensity of the emitted radiation changes with both the temperature and wavelength.

The Planck law radiation problem focuses on the spectral distribution of radiation emitted by a blackbody at different temperatures. It describes the relationship between the intensity or energy density of radiation and the wavelength, providing insights into the behavior of electromagnetic radiation and its thermal properties in various physical systems.

From (29), we obtain

$$\varphi'(\lambda)=\left(\frac{2c^2h\lambda^{-6}}{e^{ch/\lambda kT}-1}\right)\left(\frac{(ch/\lambda kT)e^{ch/\lambda kT}}{e^{ch/\lambda kT}-1}-5\right)=AB.$$

It is clear that a maximum value for $varphi$ occurs when $B=0$, implying that

$$\left(\frac{(ch/\lambda kT)e^{ch/\lambda kT}}{e^{ch/\lambda kT}-1}\right)=5.$$

By choosing $t = ch/\lambda kT$, *we have*

$$1 - \frac{x}{5} = e^{-t}. \tag{30}$$

We examined this case three times and produced the necessary nonlinear function

$$\Phi_2(t) = \left(e^{-t} - 1 + \frac{t}{5}\right)^3. \tag{31}$$

It is clear from the above equation that it has a multiple root $t = 0$ *but it is not taken into account. Bardie* [16] *has more information. Hence, we choose* $\alpha \approx 4.96511423$ *as a multiple root with* $t_0 = 5.4$. *Table* 3 *displays the numerical results.*

Example 3. *The Manning equation is used to approximate the mean flow velocity in open channel flow (see* [17]*). The mathematical equation is as follows:*

$$\delta = \sqrt{b}\left(\tan^{-1}\sqrt{\frac{M_2^2-1}{b}} - \tan^{-1}\sqrt{\frac{M_1^2-1}{b}}\right) - \tan^{-1}\sqrt{M_2^2-1} + \tan^{-1}\sqrt{M_1^2-1}.$$

Then, it shows the relationship between the Mach numbers M_1 *and* M_2 *before and after the corner, and* $b = \frac{\gamma+1}{\gamma-1}$, γ *is the specific heat ratio of the gas. The Manning equation is widely used in civil engineering and hydraulic applications for estimating flow velocities and designing open channel systems. It provides a practical approach to estimate flow characteristics in a variety of channel configurations.*

We resolve the equation for M_2 *in a unique case study when* $M_1 = \frac{3}{2}$, $\gamma = \frac{7}{5}$ *and* $\delta = 10^0$. *As it is, we have*

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

where $t = M_2$.

We took this case into consideration four times and discovered the necessary nonlinear function.

$$\Phi_3(t) = \left[\arctan\left(\frac{\sqrt{5}}{2}\right) - \arctan\sqrt{t^2-1} + \sqrt{6}\left\{\arctan\left(\sqrt{\frac{t^2-1}{6}}\right) - \arctan\left(\frac{1}{2}\sqrt{\frac{5}{6}}\right)\right\} - \frac{11}{63}\right]^4.$$

The above function has zero at $\alpha = 1.8411294068\ldots$. *Utilizing the original estimate* $t_0 = 1.5$, *this zero is determined. Table* 4 *presents the numerical outcomes.*

Example 4. *Blood rheology model: We take into account the research on the blood rheology model, which looks at the physical and flow properties of blood. The term Caisson fluid is used to describe blood, a non-Newtonian fluid. The Caisson fluid model predicts how fundamental fluids flow in tubes so that there is a velocity gradient from wall to wall and the fluid's central core travels like a plug with little distortion. When analyzing the plug flow of Caisson fluids, the function*

$$H = \frac{t^4}{21} + \frac{4t}{3} - \frac{16\sqrt{t}}{7} + 1$$

is used as a nonlinear equation.

We use the non-linear equation

$$\Phi_4(t) = \frac{1}{441}t^8 - \frac{8}{63}t^5 - \frac{2}{21}t^4 + \frac{16}{9}t^2 - \frac{376}{147}t + 1,$$

to calculate the flow rate reduction for $H = 0$. *This function has zero* $\alpha = 1$. *To compute this zero, let us use the original estimate* $t_0 = -2.5$. *Table* 5 *displays the computed findings.*

Example 5. *Now, we consider the preceding standard complex root problem:*

$$\Phi_5(t) = t(t^2+1)(2e^{t^2+1}+t^2-1)\cosh^3\left(\frac{\pi t}{2}\right).$$

This function has zero $\alpha = i$. Let us choose the initial approximation $t_0 = 1.3i$ to compute this zero. Table 6 depicts the computed findings.

Example 6. *As a final consideration, we look at a non-differential function at $t = 0$, which is given by*

$$\Phi_6(t) = \frac{(t^2+t-1)(t-3)^4}{e^t-1}.$$

This function has zero $\alpha = 3$. As a starting point, let us compute this zero using the approximation $t_0 = 0.2$. The computed outcomes are shown in Table 7 of the report.

To find the multiplicity of the above-chosen functions, adopt 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 obtain the multiplicity by using this formula with our new approach, NM. In Table 1, the calculated outcomes are displayed.

Table 1. Multiplicity of considered functions.

Problems	Multiplicity
$\Phi_1(t)$	2
$\Phi_2(t)$	3
$\Phi_3(t)$	4
$\Phi_4(t)$	3
$\Phi_5(t)$	5
$\Phi_6(t)$	4

Table 2. Results of the methods for problem $\Phi_1(t)$.

Methods	k	$\lvert t_2 - t_1\rvert$	$\lvert t_3 - t_2\rvert$	$\lvert t_4 - t_3\rvert$	COC	CPU
LLC	6	6.59(−2)	4.67(−3)	3.77(−6)	4.000	0.0941
LCN	6	6.59(−2)	4.67(−3)	3.77(−6)	4.000	0.0939
SSM	6	6.72(−2)	5.05(−3)	5.32(−6)	4.000	0.0933
ZCM	6	6.99(−2)	5.90(−3)	1.09(−6)	4.000	0.0872
SBM	6	6.59(−2)	4.67(−3)	3.77(−6)	4.000	0.0927
KKB	6	6.50(−2)	4.39(−3)	2.49(−6)	4.000	0.0814
SKJ	6	7.17(−2)	6.50(−3)	1.86(−5)	4.000	0.0821
BAM	6	5.78(−2)	2.74(−3)	2.96(−7)	4.000	0.0836
KKS	5	5.59(−2)	2.36(−3)	1.22(−7)	4.000	0.0801
NM	5	5.40(−2)	2.00(−3)	3.81(−8)	4.000	0.0799

Table 3. Results of the methods for problem $\Phi_2(t)$.

Methods	k	$\lvert t_2 - t_1 \rvert$	$\lvert t_3 - t_2 \rvert$	$\lvert t_4 - t_3 \rvert$	COC	CPU
LLC	4	1.95(−5)	1.17(−22)	1.51(−91)	4.000	0.8192
LCN	4	1.95(−5)	1.17(−22)	1.51(−91)	4.000	0.9367
SSM	4	1.95(−5)	1.17(−22)	1.53(−91)	4.000	0.9523
ZCS	4	1.96(−5)	1.18(−22)	1.58(−91)	4.000	0.9681
SBM	4	1.95(−5)	1.18(−22)	1.54(−91)	4.000	1.1543
KKB	4	1.95(−5)	1.16(−22)	1.44(−91)	4.000	0.9367
SKJ	3	2.76(−6)	8.00(−27)	0	4.000	0.3371
BAM	3	2.41(−6)	3.85(−27)	0	4.000	0.5150
KKS	3	2.42(−6)	3.93(−27)	0	4.000	0.3435
NM	3	2.43(−6)	3.98(−27)	0	4.000	0.3281

Table 4. Results of the methods for problem $\Phi_3(t)$.

Methods	k	$\lvert t_2 - t_1 \rvert$	$\lvert t_3 - t_2 \rvert$	$\lvert t_4 - t_3 \rvert$	COC	CPU
LLC	4	1.07(−3)	1.14(−14)	1.46(−58)	4.000	1.6220
LCN	4	1.07(−3)	1.13(−14)	1.43(−58)	4.000	1.7322
SSM	4	1.07(−3)	1.12(−14)	1.35(−58)	4.000	1.6847
ZCS	4	1.07(−3)	1.10(−14)	1.23(−58)	4.000	1.7000
SBM	4	1.07(−3)	1.08(−14)	1.16(−58)	4.000	1.9821
KKB	4	1.07(−3)	1.19(−14)	1.82(−58)	4.000	1.7476
SKJ	4	2.64(−5)	6.95(−21)	3.34(−83)	4.000	1.3955
BAM	4	2.63(−5)	4.59(−21)	4.23(−84)	4.000	2.2621
KKS	4	2.63(−5)	4.57(−21)	4.18(−84)	4.000	1.3401
NM	4	2.63(−5)	4.56(−21)	4.13(−84)	4.000	1.3219

Table 5. Results of the methods for problem $\Phi_4(t)$.

Methods	k	$\lvert t_2 - t_1 \rvert$	$\lvert t_3 - t_2 \rvert$	$\lvert t_4 - t_3 \rvert$	COC	CPU
LLC	-	-	-	-	-	-
LCN	-	-	-	-	-	-
SSM	10	64.4	35.5	14.9	4.000	0.0323
ZCS	6	2.18	4.55(−1)	3.46(−4)	4.000	0.0622
SBM	5	1.36	2.80(−2)	3.78(−9)	4.000	0.0454
KKB	261	666	396	162	4.000	0.5319
SKJ	5	3.39(−1)	9.76(−5)	1.12(−18)	4.000	0.0187
BAM	5	5.97(−1)	4.45(−4)	1.90(−16)	4.000	0.0310
KKS	5	6.50(−1)	5.19(−4)	3.11(−16)	4.000	0.0183
NM	5	7.00(−1)	5.92(−4)	4.79(−16)	4.000	0.0180

Table 6. Results of the methods for problem $\Phi_5(t)$.

Methods	k	$\lvert t_2 - t_1 \rvert$	$\lvert t_3 - t_2 \rvert$	$\lvert t_4 - t_3 \rvert$	COC	CPU
LLC	4	3.04(−4)	3.16(−15)	3.68(−59)	4.000	1.4352
LCN	4	3.04(−4)	3.16(−15)	3.70(−59)	4.000	2.1376
SSM	4	3.04(−4)	3.17(−15)	3.76(−59)	4.000	2.1383
ZCS	4	3.04(−4)	3.18(−15)	3.84(−59)	4.000	2.1845
SBM	4	3.04(−4)	3.23(−15)	4.14(−59)	4.000	2.6374
KKB	4	3.04(−4)	3.11(−15)	3.40(−59)	4.000	2.1690
SKJ	4	3.23(−5)	2.14(−19)	4.16(−76)	4.000	0.4578
BAM	4	3.34(−5)	1.52(−19)	6.41(−77)	4.000	0.6860
KKS	4	3.09(−5)	1.11(−19)	1.83(−77)	4.000	0.4527
NM	4	2.75(−5)	6.96(−20)	2.85(−78)	4.000	0.4498

Table 7. Results of the methods for problem $\Phi_6(t)$.

Methods	k	$\|t_2 - t_1\|$	$\|t_3 - t_2\|$	$\|t_4 - t_3\|$	COC	CPU
LLC	-	-	-	-	-	-
LCN	-	-	-	-	-	-
SSM	-	-	-	-	-	-
ZCS	-	-	-	-	-	-
SBM	7	$2.06(-1)$	1.68	$1.68(-1)$	4.000	0.5169
KKB	-	-	-	-	-	-
SKJ	-	-	-	-	-	-
BAM	-	-	-	-	-	-
KKS	10	2.42	1.98	$8.02(-1)$	2.000	0.2663
NM	7	2.27	1.94	$6.17(-1)$	4.000	0.1725

Remark 4. *Tables 2–7 demonstrate that the suggested technique exhibits a constant convergence behavior and follows the conclusions made in Sections 2 and 3. In all of the issues, it appears that the provided algorithm goes through k iterations, as few as or as many as the approaches under consideration. The estimated errors of the algorithm described are shown in Tables 2–7 to be comparable to those of other approaches. Our new approach is more consistent than other approaches and produces results that are reliable and reproducible. Our approach produces consistent results throughout the considered problem. Additionally, the results demonstrate the new method's high efficiency when compared to the CPU time of approaches in the same domain that were also under consideration. Figure 1 through Figure 6, which demonstrate how efficiently the new algorithm outperforms the existing one, also indicate the time consumption. Similar numerical studies on a wide range of other problems also support this result. By reducing the amount of time needed to finish, our new method has a clear benefit over other methods. By using our method, researchers can benefit from improved productivity and expedited results, which ultimately improves their work flow.*

5. Conclusions

This research study concludes by presenting the creation of an ideal derivative-free method for locating multiple roots. The method demonstrates a remarkable convergence rate of order four, as proven by a theorem. Moreover, the practical applications of this method extend to various real-life problems, including the Van der Waals equation, Planck's radiation, the Manning problem, and the blood rheology model. The numerical results obtained from applying the proposed method surpass those of existing iterative methods. The improved accuracy and efficiency of the derivative-free approach make it a valuable tool for solving problems with multiple roots.

Overall, this research contributes to the field of numerical analysis by providing a robust and efficient method for finding multiple roots. Its applications in diverse scientific and engineering domains highlight its versatility and effectiveness. The promising results obtained pave the way for the further exploration and implementation of this method in various practical scenarios.

Author Contributions: S.K.: Conceptualization; Methodology; Validation; Writing—Original Draft Preparation; Writing—Review and Editing. R.B.: Conceptualization; Methodology; Validation; Writing—Original Draft Preparation; Writing—Review and Editing. A.A.: Conceptualization; Writing—Review and Editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was funded by Institutional Fund Projects under grant no. (IFPIP: 1517-247-1443).

Data Availability Statement: Not applicable.

Acknowledgments: This research work was funded by Institutional Fund Projects under grant no. (IFPIP: 1517-247-1443). The authors gratefully acknowledge technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

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

References

1. Schröder, E. Über unendlich viele Algorithmen zur Auflösung der Gleichungen. *Math. Ann.* **1870**, *2*, 317–365. [CrossRef]
2. Kung, H.T.; Traub, J.F. Optimal order of one-point and multipoint iteration. *J. Assoc. Comput. Mach.* **1974**, 21, 643–651. [CrossRef]
3. Sharma, J.R.; Kumar, S.; Jäntschi, L. On a class of optimal fourth order multiple root solvers without using derivatives. *Symmetry* **2019**, *11*, 1452. [CrossRef]
4. Behl, R.; Alharbi, S.K.; Mallawi, F.O.; Salimi, M. An Optimal Derivative-Free Ostrowski's Scheme for Multiple Roots of Nonlinear Equations. *Mathematics* **2020**, *8*, 1809. [CrossRef]
5. Kumar, S.; Kumar, D.; Sharma, J.R.; Cesarano, C.; Agarwal, P.; Chu, Y.M. An Optimal fourth order derivative-free numerical algorithm for multiple roots. *Symmetry* **2020**, *12*, 1038. [CrossRef]
6. Qudsi, R.; Imran, M.; Syamsudhuha. A sixth-order iterative method free from derivative for solving multiple roots of a nonlinear equation. *Appl. Math. Sci.* **2014**, *8*, 5721–5730. [CrossRef]
7. Qudsi, R.; Imran, M.; Syamsudhuha. Another sixth-order iterative method free from derivative for solving multiple roots of a nonlinear equation. *Appl. Math. Sci.* **2017**, *11*, 2121–2129. [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.; Liao, X.; Cheng, L. A new fourth-order iterative method for finding multiple roots of nonlinear equations. *Appl. Math. Comput.* **2009**, *215*, 1288–1292.
10. 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]
11. Sharma, J.R.; Sharma, R. Modified Jarratt method for computing multiple roots. *Appl. Math. Comput.* **2010**, *217*, 878–881. [CrossRef]
12. 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]
13. 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]
14. Cordero, A.; Torregrosa, J.R. Variants of Newton's method using fifth-order quadrature formulas. *Appl. Math. Comput.* **2007**, *190*, 686–698. [CrossRef]
15. Wolfram, S. *The Mathematica Book*, 5th ed.; Wolfram Media: Champaign, IL, USA, 2003.
16. Bradie, B. *A Friendly Introduction to Numerical Analysis*; Pearson Education Inc.: New Delhi, India, 2006.
17. Hoffman, J.D. *Numerical Methods for Engineers and Scientists*; McGraw-Hill Book Company: New York, NY, USA, 1992.