

An Efficient Class of Traub-Steffensen-Like Seventh Order Multiple-Root Solvers with Applications

Janak Raj Sharma ^{1,*} , Deepak Kumar ¹  and Ioannis K. Argyros ^{2,*}

¹ Department of Mathematics, Sant Longowal Institute of Engineering and Technology Longowal Sangrur 148106, India; deepak.babbi@gmail.com

² Department of Mathematical Sciences, Cameron University, Lawton, OK 73505, USA

* Correspondence: jrshira@yahoo.co.in (J.R.S.); iargyros@cameron.edu (I.K.A.)

Received: 20 March 2019; Accepted: 8 April 2019; Published: 10 April 2019



Abstract: Many higher order multiple-root solvers that require derivative evaluations are available in literature. Contrary to this, higher order multiple-root solvers without derivatives are difficult to obtain, and therefore, such techniques are yet to be achieved. Motivated by this fact, we focus on developing a new family of higher order derivative-free solvers for computing multiple zeros by using a simple approach. The stability of the techniques is checked through complex geometry shown by drawing basins of attraction. Applicability is demonstrated on practical problems, which illustrates the efficient convergence behavior. Moreover, the comparison of numerical results shows that the proposed derivative-free techniques are good competitors of the existing techniques that require derivative evaluations in the iteration.

Keywords: nonlinear equations; multiple-root solvers; Traub–Steffensen method; fast algorithms

MSC: 65H05; 41A25; 49M15

1. Introduction

Solving nonlinear equations is an important task in numerical analysis and has numerous applications in engineering, mathematical biology, physics, chemistry, medicine, economics, and other disciplines of applied sciences [1–3]. Due to advances in computer hardware and software, the problem of solving the nonlinear equations by computational techniques has acquired an additional advantage of handling the lengthy and cumbersome calculations. In the present paper, we consider iterative techniques for computing multiple roots, say α , with multiplicity m of a nonlinear equation $f(x) = 0$, that is $f^{(j)}(\alpha) = 0, j = 0, 1, 2, \dots, m-1$ and $f^{(m)}(\alpha) \neq 0$. The solution α can be calculated as a fixed point of some function $M : D \subset \mathbb{C} \rightarrow \mathbb{C}$ by means of the fixed point iteration:

$$x_{n+1} = M(x_n), \quad n \geq 0 \quad (1)$$

where $x \in D$ is a scalar.

Many higher order techniques, based on the quadratically-convergent modified Newton's scheme (see [4]):

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

have been proposed in the literature; see, for example, [5–20] and the references therein. The techniques based on Newton's or the Newton-like method require the evaluations of derivatives of first order. There is another class of multiple-root techniques involving derivatives of both the first and second order; see [5,21]. However, higher order derivative-free techniques to handle the case of multiple roots

are yet to be explored. The main problem of developing such techniques is the difficulty in finding their convergence order. Derivative-free techniques are important in the situations when the derivative of the function f is difficult to compute or is expensive to obtain. One such derivative-free technique is the classical Traub–Steffensen method [22]. The Traub–Steffensen method actually replaces the derivative in the classical Newton’s method with a suitable approximation based on the 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 writing more concisely:

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

where $t_n = x_n + \beta f(x_n)$ and $f[x_n, t_n] = \frac{f(t_n) - f(x_n)}{t_n - x_n}$ is a first order divided difference. In this way, the modified Newton’s scheme (2) assumes the form of the modified Traub–Steffensen scheme:

$$x_{n+1} = x_n - m \frac{f(x_n)}{f[x_n, t_n]}. \quad (3)$$

The Traub–Steffensen scheme (3) is a noticeable improvement of Newton’s scheme, since it maintains the quadratic convergence without using any derivative.

The aim of the present contribution is to develop derivative-free multiple-root iterative techniques with high computational efficiency, which means the techniques that may attain a high convergence order using as small a number of function evaluations as possible. Consequently, we develop a family of derivative-free iterative methods of seventh order convergence that requires only four function evaluations per full iteration. The scheme is composed of three steps, out of which the first step is the classical Traub–Steffensen iteration (3) and the last two steps are Traub–Steffensen-like iterations. The methodology is based on the simple approach of using weight functions in the scheme. Many special cases of the family can be generated depending on the different forms of weight functions. The efficacy of the proposed methods is tested on various numerical problems of different natures. In the comparison with existing techniques requiring derivative evaluations, the new derivative-free methods are observed to be computationally more efficient.

We summarize the contents of the rest of paper. In Section 2, the scheme of the seventh order multiple-root solvers is developed and its order of convergence is determined. In Section 3, the basins of attractors are presented to check the stability of new methods. To demonstrate the performance and comparison with existing techniques, the new techniques are applied to solve some practical problems in Section 4. Concluding remarks are given in Section 5.

2. Development of the Family of Methods

Given a known multiplicity $m \geq 1$, we consider a three-step iterative scheme with the first step as the Traub–Steffensen iteration (3) as follows:

$$\begin{cases} y_n = x_n - m \frac{f(x_n)}{f[x_n, t_n]} \\ z_n = y_n - muH(u) \frac{f(x_n)}{f[x_n, t_n]} \\ x_{n+1} = z_n - mvG(u, w) \frac{f(x_n)}{f[x_n, t_n]} \end{cases} \quad (4)$$

where $u = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$, $v = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$, and $w = \left(\frac{f(z_n)}{f(y_n)}\right)^{\frac{1}{m}}$. The function $H(u) : \mathbb{C} \rightarrow \mathbb{C}$ is analytic in a neighborhood of 0, and the function $G(u, w) : \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{C}$ is holomorphic in a neighborhood of

(0, 0). Note that second and third steps are weighted by the factors $H(u)$ and $G(u, w)$, so these factors are called weight factors or weight functions.

We shall find conditions under which the scheme (4) achieves convergence order as high as possible. In order to do this, let us prove the following theorem:

Theorem 1. Let $f : \mathbb{C} \rightarrow \mathbb{C}$ be an analytic function in a region enclosing a multiple zero α with multiplicity m . Assume that initial guess x_0 is sufficiently close to α , then the iteration scheme defined by (4) possesses the seventh order of convergence, provided that the following conditions are satisfied:

$$\begin{cases} H(0) = 1, & H'(0) = 2, & H''(0) = -2, & \text{and} & |H'''(0)| < \infty \\ G(0,0) = 1, & G_{10}(0,0) = 2, & G_{01}(0,0) = 1, & G_{20}(0,0) = 0 & \text{and} & |G_{11}(0,0)| < \infty \end{cases}$$

where $G_{ij}(0,0) = \frac{\partial^{i+j}}{\partial u^i \partial w^j} G(u,w)|_{(0,0)}$.

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

$$\begin{aligned} f(x_n) = & \frac{f^{(m)}(\alpha)}{m!} e_n^m + \frac{f^{(m+1)}(\alpha)}{(m+1)!} e_n^{m+1} + \frac{f^{(m+2)}(\alpha)}{(m+2)!} e_n^{m+2} + \frac{f^{(m+3)}(\alpha)}{(m+3)!} e_n^{m+3} + \frac{f^{(m+4)}(\alpha)}{(m+4)!} e_n^{m+4} \\ & + \frac{f^{(m+5)}(\alpha)}{(m+5)!} e_n^{m+5} + \frac{f^{(m+6)}(\alpha)}{(m+6)!} e_n^{m+6} + \frac{f^{(m+7)}(\alpha)}{(m+7)!} e_n^{m+7} + O(e_n^{m+8}) \end{aligned}$$

or:

$$f(x_n) = \frac{f^{(m)}(\alpha)}{m!} e_n^m (1 + C_1 e_n + C_2 e_n^2 + C_3 e_n^3 + C_4 e_n^4 + C_5 e_n^5 + C_6 e_n^6 + C_7 e_n^7 + O(e_n^8)) \quad (5)$$

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

Using (5) in $t_n = x_n + \beta f(x_n)$, we obtain that:

$$\begin{aligned} t_n - \alpha &= x_n - \alpha + \beta f(x_n) \\ &= e_n + \frac{\beta f^{(m)}(\alpha)}{m!} e_n^m (1 + C_1 e_n + C_2 e_n^2 + C_3 e_n^3 + C_4 e_n^4 + C_5 e_n^5 + C_6 e_n^6 + C_7 e_n^7 + O(e_n^8)). \end{aligned} \quad (6)$$

Taylor's expansion of $f(t_n)$ about α is given as:

$$\begin{aligned} f(t_n) = & \frac{f^{(m)}(\alpha)}{m!} (t_n - \alpha)^m (1 + C_1 (t_n - \alpha) + C_2 (t_n - \alpha)^2 + C_3 (t_n - \alpha)^3 + C_4 (t_n - \alpha)^4 \\ & + C_5 (t_n - \alpha)^5 + C_6 (t_n - \alpha)^6 + C_7 (t_n - \alpha)^7 + O((t_n - \alpha)^8)). \end{aligned} \quad (7)$$

By using Equations (5)–(7) in the first step of (4), after some simple calculations, it follows that:

$$y_n - \alpha = \frac{C_1}{m} e_n^2 + \frac{2mC_2 - (m+1)C_1^2}{m^2} e_n^3 + \frac{1}{m^3} ((m+1)^2 C_1^2 + m(4+3m)C_1 C_2 - 3m^2 C_3) e_n^4 + \sum_{i=1}^3 \omega_i e_n^{i+4} + O(e_n^8) \quad (8)$$

where $\omega_i = \omega_i(m, C_1, C_2, \dots, C_7)$ are given in terms of m, C_1, C_2, \dots, C_7 . The expressions of ω_i are very lengthy, so they are not written explicitly.

Taylor's expansion of $f(y_n)$ about α is given by:

$$\begin{aligned} f(y_n) = & \frac{f^{(m)}(\alpha)}{m!} \left(\frac{C_1}{m}\right)^m e_n^{2m} \left(1 + \frac{2C_2 m - C_1^2(m+1)}{C_1} e_n + \frac{1}{2mC_1^2} ((3+3m+3m^2+m^3)C_1^4 \right. \\ & \left. - 2m(2+3m+2m^2)C_1^2 C_2 + 4(-1+m)m^2 C_2^2 + 6m^2 C_1 C_3) e_n^2 + \sum_{i=1}^4 \bar{\omega}_i e_n^{i+2} + O(e_n^8) \right) \end{aligned} \quad (9)$$

where $\bar{\omega}_i = \bar{\omega}_i(m, C_1, C_2, \dots, C_7)$.

By using (5) and (9), we get the expression of u as:

$$u = \frac{C_1}{m} e_n + \frac{2C_2m - C_1^2(m+2)}{m^2} e_n^2 + \sum_{i=1}^5 \eta_i e_n^{i+2} + O(e_n^8) \quad (10)$$

where $\eta_i = \eta_i(m, C_1, C_2, \dots, C_7)$ are given in terms of m, C_1, C_2, \dots, C_7 with one explicitly-written coefficient $\eta_1 = \frac{1}{2m^3} (C_1^3(2m^2 + 7m + 7) + 6C_3m^2 - 2C_2C_1m(3m + 7))$.

We expand the weight function $H(u)$ in the neighborhood of 0 by the Taylor series, then we have that:

$$H(u) \approx H(0) + uH'(0) + \frac{1}{2}u^2H''(0) + \frac{1}{6}u^3H'''(0). \quad (11)$$

Inserting Equations (5), (9), and (11) in the second step of Scheme (4) and simplifying,

$$\begin{aligned} z_n - \alpha = & -\frac{A}{m}C_1e_n^2 + \frac{1}{m^2}(-2mAC_2 + C_1^2(-1 + mA + 3H(0) - H'(0)))e_n^3 \\ & + \frac{1}{2m^3}(-6Am^2C_3 + 2mC_1C_2(-4 + 3Am + 11H(0) - 4H'(0)) + C_1^3(2 - 2Am^2 \\ & - 13H(0) + 10H'(0) + m(4 - 11H(0) + 4H'(0)) - H''(0)))e_n^4 + \sum_{i=1}^3 \gamma_i e_n^{i+4} + O(e_n^8) \end{aligned} \quad (12)$$

where $A = -1 + H(0)$ and $\gamma_i = \gamma_i(m, H(0), H'(0), H''(0), H'''(0), C_1, C_2, \dots, C_7)$.

In order to attain higher order convergence, the coefficients of e_n^2 and e_n^3 should be simultaneously equal to zero. That is possible only for the following values of $H(0)$ and $H'(0)$:

$$H(0) = 1, \quad H'(0) = 2. \quad (13)$$

By using the above values in (12), we obtain that:

$$z_n - \alpha = \frac{-2mC_1C_2 + C_1^3(9 + m - H''(0))}{2m^3} e_n^4 + \sum_{i=1}^3 \gamma_i e_n^{i+4} + O(e_n^8). \quad (14)$$

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

$$f(z_n) = \frac{f^{(m)}(\alpha)}{m!} (z_n - \alpha)^m (1 + C_1(z_n - \alpha) + C_2(z_n - \alpha)^2 + O((z_n - \alpha)^3)). \quad (15)$$

From (5), (9), and (15), we get the expressions of v and w as:

$$v = \frac{(9 + m)C_1^3 - 2mC_1C_2}{2m^3} e_n^3 + \sum_{i=1}^4 \tau_i e_n^{i+3} + O(e_n^8) \quad (16)$$

and:

$$w = \frac{(9 + m - H''(0))C_1^2 - 2mC_2}{2m^3} e_n^2 + \sum_{i=1}^5 \varsigma_i e_n^{i+2} + O(e_n^8) \quad (17)$$

where τ_i and ς_i are some expressions of $m, H''(0), H'''(0), C_1, C_2, \dots, C_7$.

Expanding the function $G(u, w)$ in the neighborhood of origin $(0, 0)$ by Taylor series:

$$G(u, w) \approx G_{00}(0, 0) + uG_{10}(0, 0) + \frac{1}{2}u^2G_{20}(0, 0) + w(G_{01}(0, 0) + uG_{11}(0, 0)) \quad (18)$$

where $G_{ij} = \frac{\partial^{i+j}}{\partial u^i \partial w^j} G(u, w)|_{(0,0)}$.

Then by substituting (5), (16), (17), and (18) into the last step of Scheme (4), we obtain the error equation:

$$e_{n+1} = \frac{1}{2m^3} ((-1 + G_{00}(0,0))C_1(2mC_1 - (9 + m - H''(0))C_1^2))e_n^4 + \sum_{i=1}^3 \xi_i e_n^{i+4} + O(e_n^8) \quad (19)$$

where $\xi_i = \xi_i(m, H''(0), H'''(0), G_{00}, G_{01}, G_{10}, G_{20}, G_{11}, C_1, C_2, \dots, C_7)$.

It is clear from Equation (19) that we will obtain at least fifth order convergence if we have $G_{00}(0,0) = 1$. Moreover, we can use this value in $\xi_1 = 0$ to obtain:

$$G_{10}(0,0) = 2. \quad (20)$$

By using $G_{00} = 1$ and (20) in $\xi_2 = 0$, the following equation is obtained:

$$C_1(2mC_2 - C_1^2(9 + m - H''(0)))(-2mC_2(-1 + G_{01}(0,0)) + C_1^2(-11 + m(-1 + G_{01}(0,0)) - (-9 + H''(0))G_{01}(0,0) + G_{20}(0,0))) = 0$$

which further yields:

$$G_{01}(0,0) = 1, \quad G_{20}(0,0) = 0 \quad \text{and} \quad H''(0) = -2.$$

Using the above values in (19), the final error equation is given by:

$$\begin{aligned} e_{n+1} = & \frac{1}{360m^6} \left(360m^3((47 + 5m)C_2^3 - 6mC_3^2 - 10mC_2C_4) + 120m^3C_1((623 + 78m)C_2C_3 \right. \\ & - 12mC_5) - 60m^2C_1^3C_3(1861 + 1025m + 78m^2 + 12H'''(0)) + 10mC_1^4C_2(32383 \\ & + 9911m^2 + 558m^3 + 515H'''(0) + 396G_{11}(0,0) + 36m(900 + 6H'''(0) + G_{11}(0,0))) \\ & - 60m^2C_1^2(-6m(67 + 9m)C_4 + C_2^2(3539 + 1870m + 135m^2 + 24H'''(0) + 6G_{11}(0,0))) \\ & - C_1^6(95557 + 20605m + 978m^4 + 2765H'''(0) + 10890G_{11}(0,0) + m^2(90305 + 600H'''(0) \\ & \left. + 90G_{11}(0,0)) + 5m(32383 + 515H'''(0) + 396G_{11}(0,0))) \right) e_n^7 + O(e_n^8). \end{aligned}$$

Hence, the seventh order convergence is established. \square

Forms of the Weight Function

Numerous special cases of the family (4) are generated based on the forms of weight functions $H(u)$ and $G(u, w)$ that satisfy the conditions of Theorem 1. However, we restrict ourselves to simple forms, which are given as follows:

I. Some particular forms of $H(u)$

Case I(a). When $H(u)$ is a polynomial weight function, e.g.,

$$H(u) = A_0 + A_1u + A_2u^2.$$

By using the conditions of Theorem 1, we get $A_0 = 1$, $A_1 = 2$ and $A_2 = -1$. Then, $H(u)$ becomes:

$$H(u) = 1 + 2u - u^2.$$

Case I(b). When $H(u)$ is a rational weight function, e.g.,

$$H(u) = \frac{1 + A_0 u}{A_1 + A_2 u}.$$

Using the conditions of Theorem 1, we get that $A_0 = \frac{5}{2}$, $A_1 = 1$, and $A_2 = \frac{1}{2}$. Therefore,

$$H(u) = \frac{2 + 5u}{2 + u}.$$

Case I(c). When $H(u)$ is a rational weight function, e.g.,

$$H(u) = \frac{1 + A_0 u + A_1 u^2}{1 + A_2 u}.$$

Using the conditions of Theorem 1, then we get $A_0 = 3$, $A_1 = 2$, and $A_2 = 1$. $H(u)$ becomes:

$$H(u) = \frac{1 + 3u + u^2}{1 + u}.$$

Case I(d). When $H(u)$ is a rational function of the form:

$$H(u) = \frac{1 + A_0 u}{1 + A_1 u + A_2 u^2}.$$

Using the conditions of Theorem 1, we get $A_0 = 1$, $A_1 = -1$, and $A_2 = 1$. Then,

$$H(u) = \frac{1 + u}{1 - u + 3u^2}.$$

II. Some particular forms of $G(u, w)$

Case II(a). When $G(u, w)$ is a polynomial weight function, e.g.,

$$G(u, w) = A_0 + A_1 u + A_2 u^2 + (A_3 + A_4 u + A_5 u^2)w.$$

Using the conditions of Theorem 1, then we get $A_0 = 1$, $A_1 = 2$, $A_2 = 0$, and $A_3 = 1$. Therefore, $G(u, w)$ becomes:

$$G(u, w) = 1 + 2u + (1 + A_4 u + A_5 u^2)w$$

where A_4 and A_5 are free parameters.

Case II(b). When $G(u, w)$ is a rational weight function, e.g.,

$$G(u, w) = \frac{B_0 + B_1 u + B_2 w + B_3 u w}{1 + A_1 u + A_2 w + A_3 u w}.$$

Using the conditions of Theorem 1, we have $B_0 = 1$, $A_1 = 2$, $B_1 = 2$, $A_2 = 0$, and $B_2 = 1$. Then,

$$G(u, w) = \frac{1 + 2u + w + B_3 u w}{1 + A_3 u w}$$

where A_3 and B_3 are free parameters.

Case II(c). When $G(u, w)$ is the sum of two weight functions $H_1(u)$ and $H_2(w)$. Let $H_1(u) = A_0 + u A_1 + u^2 A_2$ and $H_2(w) = B_0 + w B_1 + w^2 B_2$, then $G(u, w)$ becomes:

$$G(u, w) = A_0 + A_1 u + A_2 u^2 + B_0 + B_1 w + B_2 w^2.$$

By using the conditions of Theorem 1, we get:

$$G(u, w) = 1 + 2u + w + B_2 w^2$$

where B_2 is a free parameter.

Case II(d). When $G(u, w)$ is the sum of two rational weight functions, that is:

$$G(u, w) = \frac{A_0 + A_1 u}{1 + A_2 u} + \frac{B_0 + B_1 w}{1 + B_2 w}.$$

By using the conditions of Theorem 1, we obtain that:

$$G(u, w) = 2u + \frac{1}{1 - w}.$$

Case II(e). When $G(u, w)$ is product of two weight functions, that is:

$$G(u, w) = \frac{A_0 + A_1 u}{1 + A_2 u} \times \frac{B_0 + B_1 w}{1 + B_2 w}.$$

Using the conditions of Theorem 1, then we get:

$$G(u, w) = (1 + 2u)(1 + w).$$

3. Complex Dynamics of Methods

Here, our aim is to analyze the complex dynamics of the proposed methods based on the visual display of the basins of attraction of the zeros of a polynomial $p(z)$ in the complex plane. Analysis of the complex dynamical behavior gives important information about the convergence and stability of an iterative scheme. Initially, Vrscay and Gilbert [23] introduced the idea of analyzing complex dynamics. Later on, many researchers used this concept in their work, for example (see [24–26] and the references therein). We choose some of the special cases (corresponding to the above forms of $H(u)$ and $G(u, w)$) of family (4) to analyze the basins. Let us choose the combinations of special Cases II(c) (for $B_2 = 1$) and II(d) with I(a), I(b), I(c), and I(d) in (4) and denote the corresponding new methods by NM-i(j), $i = 1, 2$ and $j = a, b, c, d$.

We take the initial point as $z_0 \in D$, where D is a rectangular region in \mathbb{C} containing all the roots of $p(z) = 0$. The iterative methods starting at a point z_0 in a rectangle either converge to the zero of the function $p(z)$ or eventually diverge. The stopping criterion considered for convergence is 10^{-3} up to a maximum of 25 iterations. If the desired tolerance is not achieved in 25 iterations, we do not continue and declare that the iterative method starting at point z_0 does not converge to any root. The strategy adopted is the following: A color is assigned to each starting point z_0 in the basin of attraction of a zero. If the iteration starting from the initial point z_0 converges, then it represents the basins of attraction with that particular color assigned to it, and if it fails to converge in 25 iterations, then it shows the black color.

To view complex geometry, we analyze the attraction basins for the methods NM-1(a–d) and NM-2(a–d) on the following two polynomials:

Problem 1. In the first example, we consider the polynomial $p_1(z) = (z^2 - 1)^2$, which has zeros $\{\pm 1\}$ with multiplicity two. In this case, we use a grid of 400×400 points in a rectangle $D \in \mathbb{C}$ of size $[-2, 2] \times [-2, 2]$ and assign 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 obtained for the methods NM-1(a–d) and NM-2(a–d) are shown in Figures 1–4 corresponding to $\beta = 0.01, 0.002$. Observing the behavior of the methods, we see that the method NM-2(d) possesses a lesser number of divergent points and therefore has better stability than the

remaining methods. Notice that there is a small difference in the basins for the rest of the methods with the same value of β . Notice also that the basins are becoming wider as parameter β assumes smaller values.

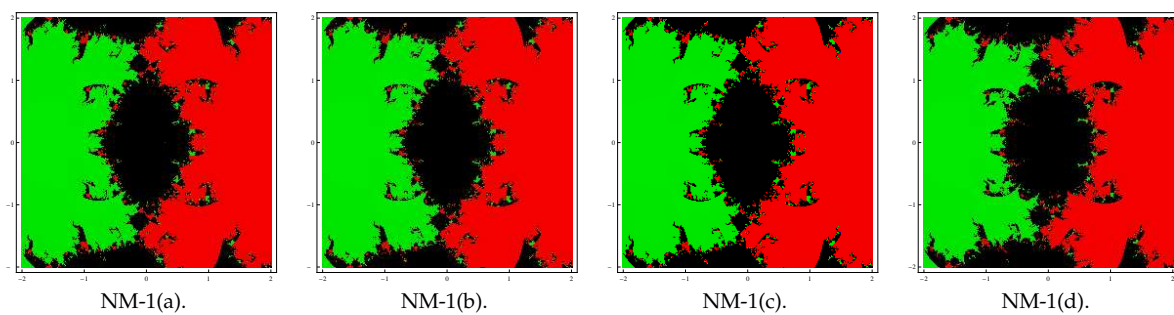


Figure 1. Basins of attraction for NM-1 (for $\beta = 0.01$) in polynomial $p_1(z)$.

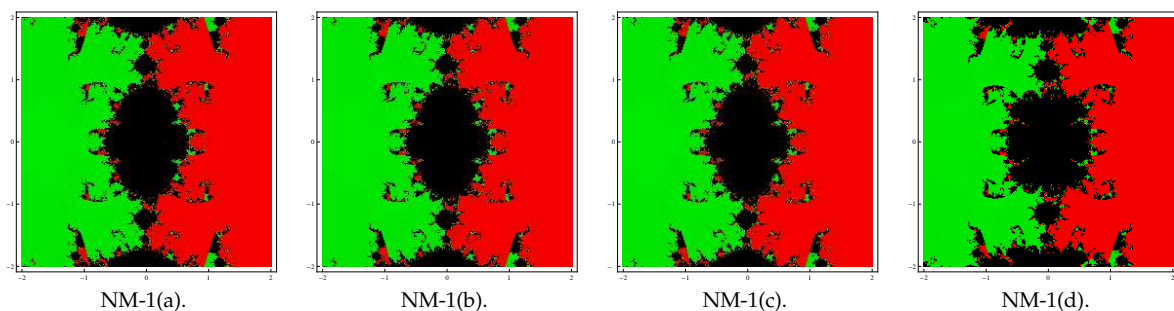


Figure 2. Basins of attraction for NM-1 (for $\beta = 0.002$) in polynomial $p_1(z)$.

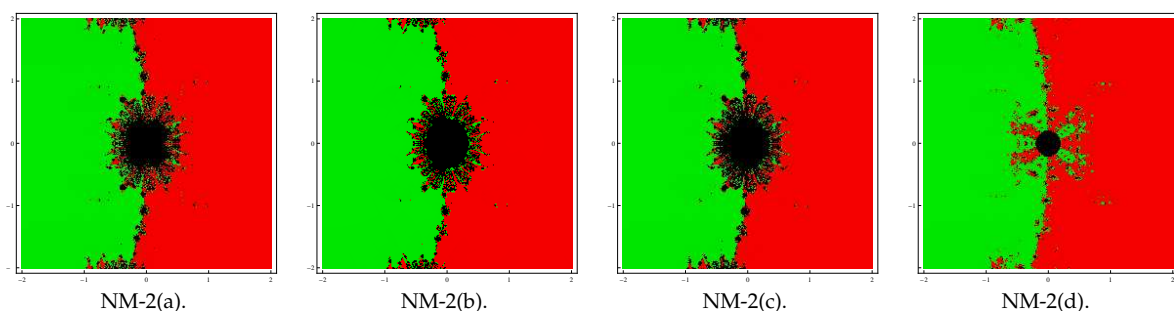


Figure 3. Basins of attraction for NM-2 (for $\beta = 0.01$) in polynomial $p_1(z)$.

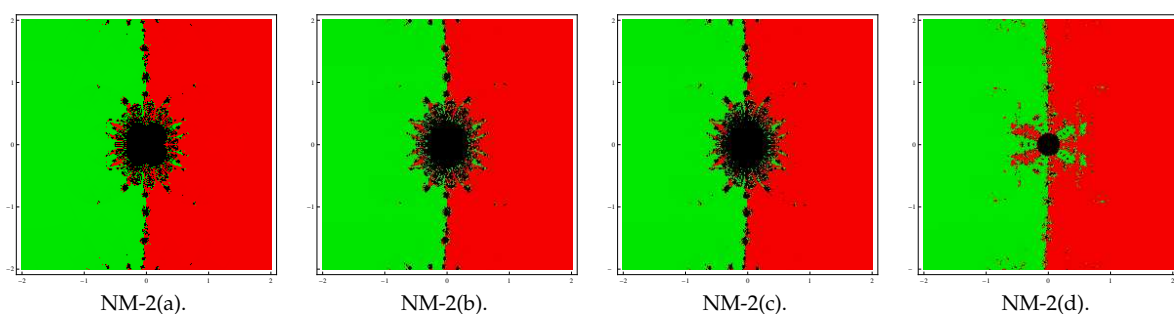


Figure 4. Basins of attraction for NM-2 (for $\beta = 0.002$) in polynomial $p_1(z)$.

Problem 2. Let us take the polynomial $p_2(z) = (z^3 + z)^3$ having zeros $\{0, \pm i\}$ with multiplicity three. 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, green, and blue to each point in the basin of attraction of $-i$, 0 , and i , respectively. Basins for this problem are shown in Figures 5–8 corresponding to parameter choices $\beta = 0.01, 0.002$ in the proposed methods. Observing the behavior, we see that again, the method NM-2(d) has better convergence behavior due to a lesser number of divergent points. Furthermore, observe that in each case, the basins are becoming larger with the smaller values of β . The basins in the remaining methods other than NM-2(d) are almost the same.

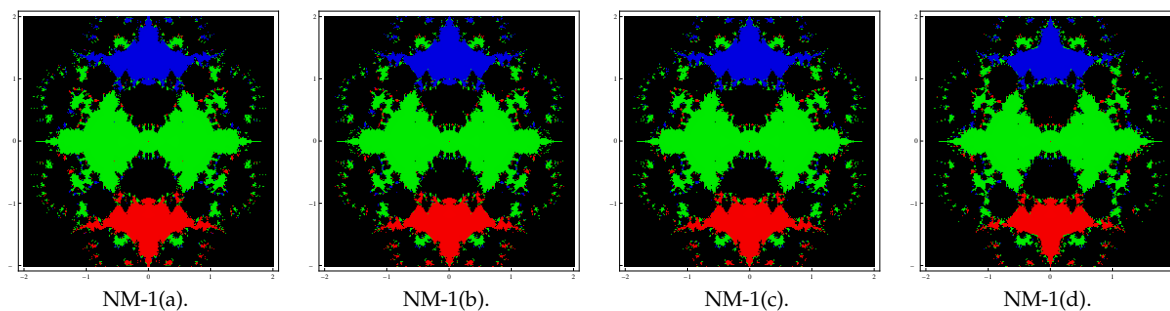


Figure 5. Basins of attraction for NM-1 (for $\beta = 0.01$) in polynomial $p_2(z)$.

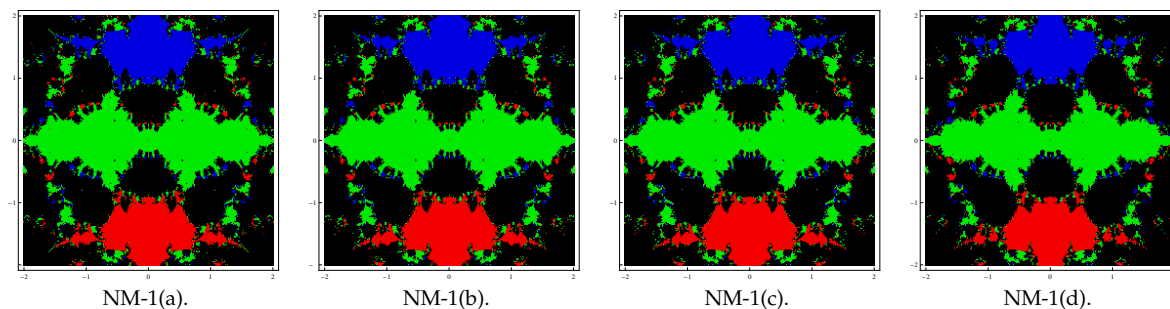


Figure 6. Basins of attraction for NM-1 (for $\beta = 0.002$) in polynomial $p_2(z)$.

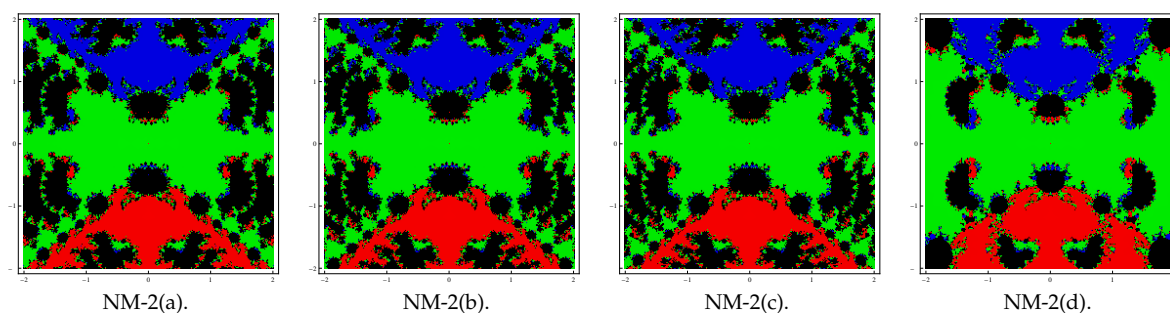


Figure 7. Basins of attraction for NM-2 (for $\beta = 0.01$) in polynomial $p_2(z)$.

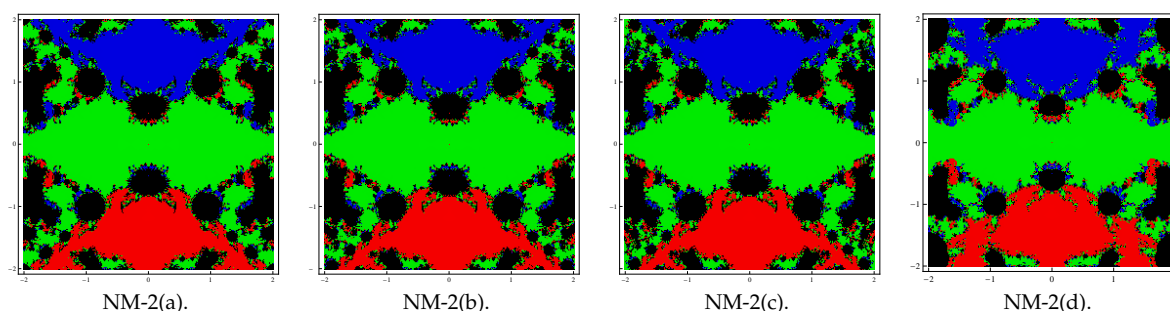


Figure 8. Basins of attraction for NM-2 (for $\beta = 0.002$) in polynomial $p_2(z)$.

From the graphics, we can easily observe the behavior and applicability of any method. If we choose an initial guess z_0 in a region wherein different basins of attraction touch each other, it is difficult to predict which root is going to be attained by the iterative method that starts from z_0 . Therefore, the choice of z_0 in such a region is not a good one. Both black regions and the regions with different colors are not suitable to assume the initial guess as z_0 when we are required to achieve a particular root. The most intricate geometry is between the basins of attraction, and this corresponds to the cases where the method is more demanding with respect to the initial point. We conclude this section with a remark that the convergence behavior of the proposed techniques depends on the value of parameter β . The smaller the value of β is, the better the convergence of the method.

4. Numerical Examples and Discussion

In this section, we implement the special cases NM-1(a–d) and NM-2(a–d) that we have considered in the previous section of the family (4), to obtain zeros of nonlinear functions. This will not only illustrate the methods' practically, but also serve to test the validity of theoretical results. The theoretical order of convergence is also confirmed by calculating the computational order of convergence (COC) using the formula (see [27]):

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

The performance is compared with some well-known higher order multiple-root solvers such as the sixth order methods by Geum et al. [8,9], which are expressed below:

First method by Geum et al. [8]:

$$\begin{cases} y_n = x_n - m \frac{f(x_n)}{f'(x_n)} \\ x_{n+1} = y_n - Q_f(u, s) \frac{f(y_n)}{f'(y_n)} \end{cases} \quad (21)$$

where $u = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$ and $s = \left(\frac{f'(y_n)}{f'(x_n)}\right)^{\frac{1}{m-1}}$, and $Q_f : \mathbb{C}^2 \rightarrow \mathbb{C}$ is a holomorphic function in the neighborhood of origin (0, 0). The authors have also studied various forms of the function Q_f leading to sixth order convergence of (21). We consider the following four special cases of function $Q_f(u, s)$ in the formula (21) and denote the corresponding methods by GKN-1(j), $j = a, b, c, d$:

- (a) $Q_f(u, s) = m(1 + 2(m-1)(u-s) - 4us + s^2)$
- (b) $Q_f(u, s) = m(1 + 2(m-1)(u-s) - u^2 - 2us)$
- (c) $Q_f(u, s) = \frac{m+au}{1+bu+cs+dus}$,
where $a = \frac{2m}{m-1}$, $b = 2-2m$, $c = \frac{2(2-2m+m^2)}{m-1}$, and $d = -2m(m-1)$
- (d) $Q_f(u, s) = \frac{m+a_1u}{1+b_1u+c_1u^2+d_1s}$,
where $a_1 = \frac{2m(4m^4-16m^3+31m^2-30m+13)}{(m-1)(4m^2-8m+7)}$, $b_1 = \frac{4(2m^2-4m+3)}{(m-1)(4m^2-8m+7)}$,
 $c_1 = -\frac{4m^2-8m+3}{4m^2-8m+7}$, and $d_1 = 2(m-1)$.

Second method by Geum et al. [9]:

$$\begin{cases} y_n = x_n - m \frac{f(x_n)}{f'(x_n)} \\ z_n = x_n - mG_f(u) \frac{f(x_n)}{f'(x_n)} \\ x_{n+1} = x_n - mK_f(u, v) \frac{f(x_n)}{f'(x_n)} \end{cases} \quad (22)$$

where $u = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$ and $v = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$. The function $G_f : \mathbb{C} \rightarrow \mathbb{C}$ is analytic in a neighborhood of 0, and $K_f : \mathbb{C}^2 \rightarrow \mathbb{C}$ is holomorphic in a neighborhood of (0, 0). Numerous cases of G_f and K_f have been proposed in [9]. We consider the following four special cases and denote the corresponding methods by GKN-2(j), $j = a, b, c, d$:

- (a) $Q_f(u) = \frac{1+u^2}{1-u}$, $K_f(u, v) = \frac{1+u^2-v}{1-u+(u-2)v}$

- (b) $Q_f(u) = 1 + u + 2u^2$, $K_f(u, v) = 1 + u + 2u^2 + (1 + 2u)v$
 (c) $Q_f(u) = \frac{1 + u^2}{1 - u}$, $K_f(u, v) = 1 + u + 2u^2 + 2u^3 + 2u^4 + (2u + 1)v$
 (d) $Q_f(u) = \frac{(2u - 1)(4u - 1)}{1 - 7u + 13u^2}$, $K_f(u, v) = \frac{(2u - 1)(4u - 1)}{1 - 7u + 13u^2 - (1 - 6u)v}$

Computations were carried out in the programming package *Mathematica* with multiple-precision arithmetic. Numerical results shown in Tables 1–4 include: (i) the number of iterations (n) required to converge to the solution, (ii) the values of the last three consecutive errors $e_n = |x_{n+1} - x_n|$, (iii) the computational order of convergence (COC), and (iv) the elapsed CPU time (CPU-time). The necessary iteration number (n) and elapsed CPU time are calculated by considering $|x_{n+1} - x_n| + |f(x_n)| < 10^{-350}$ as the stopping criterion.

The convergence behavior of the family of iterative methods (4) is tested on the following problems:

Example 1. (Eigenvalue problem). Finding eigenvalues of a large square matrix is one of the difficult tasks in applied mathematics and engineering. Finding even the roots of the characteristic equation of a square matrix of order greater than four is a big challenge. Here, we consider the following 6×6 matrix:

$$M = \begin{bmatrix} 5 & 8 & 0 & 2 & 6 & -6 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 6 & 18 & -1 & 1 & 13 & -9 \\ 3 & 6 & 0 & 4 & 6 & -6 \\ 4 & 14 & -2 & 0 & 11 & -6 \\ 6 & 18 & -2 & 1 & 13 & -8 \end{bmatrix}.$$

The characteristic equation of the above matrix (M) is given as follows:

$$f_1(x) = x^6 - 12x^5 + 56x^4 - 130x^3 + 159x^2 - 98x + 24.$$

This function has one multiple zero at $\alpha = 1$ of multiplicity three. We choose initial approximation $x_0 = 0.25$. Numerical results are shown in Table 1.

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

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-Time
GKN-1(a)	4	5.46×10^{-3}	2.40×10^{-14}	1.78×10^{-82}	6.0000	0.05475
GKN-1(b)	4	5.65×10^{-3}	3.22×10^{-14}	1.13×10^{-81}	6.0000	0.05670
GKN-1(c)	4	5.41×10^{-3}	2.80×10^{-14}	5.59×10^{-82}	6.0000	0.05856
GKN-1(d)	4	7.52×10^{-3}	4.85×10^{-13}	3.78×10^{-74}	6.0000	0.05504
GKN-2(a)	4	2.85×10^{-3}	1.57×10^{-16}	4.32×10^{-96}	6.0000	0.07025
GKN-2(b)	4	9.28×10^{-3}	1.58×10^{-12}	4.13×10^{-71}	6.0000	0.05854
GKN-2(c)	4	7.11×10^{-3}	1.87×10^{-13}	6.53×10^{-77}	6.0000	0.06257
GKN-2(d)	5	1.03×10^{-5}	3.87×10^{-30}	1.07×10^{-176}	6.0000	0.07425
NM-1(a)	4	1.62×10^{-3}	1.79×10^{-19}	3.58×10^{-131}	7.0000	0.04675
NM-1(b)	4	1.62×10^{-3}	1.85×10^{-19}	4.63×10^{-131}	6.9990	0.05073
NM-1(c)	4	1.62×10^{-3}	1.96×10^{-19}	5.92×10^{-131}	6.9998	0.05355
NM-1(d)	4	1.60×10^{-3}	1.02×10^{-19}	4.36×10^{-133}	6.9990	0.05077
NM-2(a)	4	1.37×10^{-3}	5.56×10^{-20}	1.02×10^{-134}	6.9997	0.05435
NM-2(b)	4	1.37×10^{-3}	5.77×10^{-20}	1.35×10^{-134}	6.9998	0.05454
NM-2(c)	4	1.38×10^{-3}	5.98×10^{-20}	1.77×10^{-134}	6.9996	0.05750
NM-2(d)	4	1.34×10^{-3}	2.97×10^{-20}	8.00×10^{-137}	6.9998	0.05175

Example 2. (Kepler's equation). Let us consider Kepler's equation:

$$f_2(x) = x - \alpha \sin(x) - K = 0, \quad 0 \leq \alpha < 1 \text{ and } 0 \leq K \leq \pi.$$

A numerical study, for different values of the parameters α and K , has been performed in [28]. As a particular example, let us take $\alpha = \frac{1}{4}$ and $K = \frac{\pi}{5}$. Consider this particular case four times with same values of the parameters, then the required nonlinear function is:

$$f_2(x) = \left(x - \frac{1}{4} \sin x - \frac{\pi}{5}\right)^4.$$

This function has one multiple zero at $\alpha = 0.80926328 \dots$ of multiplicity four. The required zero is calculated using initial approximation $x_0 = 1$. Numerical results are displayed in Table 2.

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

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-Time
GKN-1(a)	5	1.90×10^{-25}	2.55×10^{-76}	1.19×10^{-228}	3.0018	2.1405
GKN-1(b)	5	1.74×10^{-25}	1.94×10^{-76}	5.28×10^{-229}	3.0018	2.1445
GKN-1(c)	5	2.31×10^{-25}	4.56×10^{-76}	6.79×10^{-228}	3.0018	2.1835
GKN-1(d)	5	1.75×10^{-25}	1.97×10^{-76}	5.51×10^{-229}	3.0018	2.1797
GKN-2(a)	5	3.52×10^{-19}	5.28×10^{-117}	1.22×10^{-233}	1.1923	1.8047
GKN-2(b)	4	9.33×10^{-9}	6.67×10^{-53}	8.88×10^{-318}	6.0000	1.4452
GKN-2(c)	4	3.74×10^{-9}	9.33×10^{-56}	2.27×10^{-335}	6.0000	1.4415
GKN-2(d)	4	1.64×10^{-8}	2.04×10^{-51}	7.50×10^{-309}	6.0000	1.4492
NM-1(a)	3	1.91×10^{-1}	5.70×10^{-10}	6.59×10^{-70}	7.0000	0.9845
NM-1(b)	3	1.91×10^{-1}	6.02×10^{-10}	1.04×10^{-69}	7.0000	0.9650
NM-1(c)	3	1.91×10^{-1}	6.32×10^{-10}	1.58×10^{-69}	7.0000	0.9570
NM-1(d)	3	1.90×10^{-1}	9.62×10^{-11}	2.48×10^{-76}	7.0540	0.8590
NM-2(a)	3	1.91×10^{-1}	5.70×10^{-10}	6.59×10^{-70}	7.0000	0.9842
NM-2(b)	3	1.91×10^{-1}	6.02×10^{-10}	1.04×10^{-69}	7.0000	0.9607
NM-2(c)	3	1.91×10^{-1}	6.32×10^{-10}	1.58×10^{-69}	7.0000	0.9767
NM-2(d)	3	1.91×10^{-1}	9.68×10^{-11}	2.63×10^{-76}	7.0540	0.7460

Example 3. (Manning's equation). Consider the isentropic supersonic flow around 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 [3]):

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

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

As a particular case study, the equation is solved for M_2 given that $M_1 = 1.5$, $\gamma = 1.4$, and $\delta = 10^0$. Then, we have that:

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

where $x = M_2$.

Consider this particular case three times with the same values of the parameters, then the required nonlinear function is:

$$f_3(x) = \left(\tan^{-1} \left(\frac{\sqrt{5}}{2} \right) - \tan^{-1} (\sqrt{x^2 - 1}) + \sqrt{6} \left(\tan^{-1} \left(\sqrt{\frac{x^2 - 1}{6}} \right) - \tan^{-1} \left(\frac{1}{2} \sqrt{\frac{5}{6}} \right) \right) - \frac{11}{63} \right)^3.$$

This function has one multiple zero at $\alpha = 1.8411027704 \dots$ of multiplicity three. The required zero is calculated using initial approximation $x_0 = 1.5$. Numerical results are shown in Table 3.

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

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-Time
GKN-1(a)	4	2.17×10^{-8}	4.61×10^{-25}	1.01×10^{-151}	6.0000	1.3047
GKN-1(b)	4	2.17×10^{-8}	4.60×10^{-25}	2.27×10^{-151}	6.0000	1.2852
GKN-1(c)	4	2.11×10^{-8}	4.21×10^{-25}	1.03×10^{-151}	6.0000	1.3203
GKN-1(d)	4	1.77×10^{-8}	2.48×10^{-25}	2.68×10^{-151}	6.0000	1.2970
GKN-2(a)	4	4.83×10^{-7}	1.36×10^{-41}	6.84×10^{-249}	6.0000	1.2382
GKN-2(b)	4	4.90×10^{-7}	2.89×10^{-41}	1.21×10^{-246}	6.0000	1.2440
GKN-2(c)	4	4.88×10^{-7}	2.22×10^{-41}	1.98×10^{-247}	6.0000	1.2422
GKN-2(d)	4	4.89×10^{-7}	3.22×10^{-41}	2.62×10^{-246}	6.0000	1.2577
NM-1(a)	4	7.85×10^{-9}	1.56×10^{-60}	0	7.0000	1.0274
NM-1(b)	4	7.85×10^{-9}	1.58×10^{-60}	0	7.0000	1.0272
NM-1(c)	4	7.89×10^{-9}	1.60×10^{-60}	0	7.0000	1.0231
NM-1(d)	4	7.84×10^{-9}	1.31×10^{-60}	0	7.0000	1.0235
NM-2(a)	4	7.69×10^{-9}	1.35×10^{-60}	0	7.0000	1.0398
NM-2(b)	4	7.69×10^{-9}	1.37×10^{-60}	0	7.0000	1.0742
NM-2(c)	4	7.69×10^{-9}	1.38×10^{-60}	0	7.0000	1.0467
NM-2(d)	4	7.68×10^{-9}	1.13×10^{-60}	0	7.0000	1.0192

Example 4. Next, consider the standard nonlinear test function:

$$f_4(x) = \left(-\sqrt{1-x^2} + x + \cos \frac{\pi x}{2} + 1 \right)^4$$

which has a multiple zero at $\alpha = -0.72855964390156 \dots$ of multiplicity four. Numerical results are shown in Table 4 with initial guess $x_0 = -0.5$.

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

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-Time
GKN-1(a)	4	7.20×10^{-6}	1.80×10^{-30}	4.39×10^{-178}	6.0000	0.1017
GKN-1(b)	4	7.21×10^{-6}	1.85×10^{-30}	5.32×10^{-178}	5.9999	0.0977
GKN-1(c)	4	7.42×10^{-6}	2.52×10^{-30}	3.84×10^{-177}	5.9999	0.1055
GKN-1(d)	4	8.83×10^{-6}	1.30×10^{-29}	1.34×10^{-172}	5.9999	0.1015
GKN-2(a)	4	2.15×10^{-5}	8.22×10^{-28}	2.60×10^{-162}	5.9999	0.1132
GKN-2(b)	4	2.39×10^{-5}	4.22×10^{-27}	1.27×10^{-157}	5.9999	0.1052
GKN-2(c)	4	2.33×10^{-5}	2.57×10^{-27}	4.61×10^{-159}	5.9999	0.1055
GKN-2(d)	4	2.43×10^{-5}	5.31×10^{-27}	5.83×10^{-157}	5.9999	0.1095
NM-1(a)	4	2.87×10^{-6}	1.03×10^{-37}	8.12×10^{-258}	6.9999	0.0720
NM-1(b)	4	2.88×10^{-6}	1.06×10^{-37}	9.60×10^{-258}	6.9999	0.0724
NM-1(c)	4	2.88×10^{-6}	1.08×10^{-37}	1.13×10^{-257}	6.9999	0.0722
NM-1(d)	4	2.83×10^{-6}	7.39×10^{-38}	6.09×10^{-259}	6.9999	0.0782
NM-2(a)	4	2.80×10^{-6}	8.55×10^{-38}	2.15×10^{-258}	6.9999	0.0732
NM-2(b)	4	2.80×10^{-6}	8.74×10^{-37}	2.54×10^{-258}	6.9999	0.0723
NM-2(c)	4	2.80×10^{-6}	8.93×10^{-38}	3.00×10^{-258}	6.9999	0.0746
NM-2(d)	4	2.76×10^{-6}	6.09×10^{-38}	1.56×10^{-259}	6.9999	0.0782

Example 5. Consider the standard function, which is given as (see [8]):

$$f_5(x) = \left(x - \sqrt{3}x^3 \cos \frac{\pi x}{6} + \frac{1}{x^2+1} - \frac{11}{5} + 4\sqrt{3} \right) (x-2)^4.$$

The multiple zero of function f_5 is $\alpha = 2$ with multiplicity five. We choose the initial approximation $x_0 = 1.5$ for obtaining the zero of the function. Numerical results are exhibited in Table 5.

Table 5. Comparison of the performance of methods for Example 5.

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-Time
GKN-1(a)	4	1.20×10^{-5}	6.82×10^{-31}	2.31×10^{-182}	6.0000	0.5820
GKN-1(b)	4	1.20×10^{-5}	6.86×10^{-31}	2.40×10^{-182}	6.0000	0.5860
GKN-1(c)	4	1.21×10^{-5}	7.72×10^{-31}	5.18×10^{-182}	6.0000	0.5937
GKN-1(d)	4	1.58×10^{-5}	1.00×10^{-29}	6.51×10^{-175}	6.0000	0.5832
GKN-2(a)	4	3.17×10^{-5}	1.64×10^{-28}	3.21×10^{-168}	6.0000	0.7120
GKN-2(b)	4	3.50×10^{-5}	6.90×10^{-28}	4.05×10^{-164}	6.0000	0.6992
GKN-2(c)	4	3.41×10^{-5}	4.42×10^{-28}	2.09×10^{-165}	6.0000	0.6915
GKN-2(d)	4	3.54×10^{-5}	8.45×10^{-28}	1.56×10^{-163}	6.0000	0.6934
NM-1(a)	4	2.35×10^{-6}	1.81×10^{-40}	2.92×10^{-279}	7.0000	0.3712
NM-1(b)	4	2.35×10^{-6}	1.84×10^{-40}	3.31×10^{-279}	7.0000	0.3360
NM-1(c)	4	2.35×10^{-6}	1.87×10^{-40}	3.74×10^{-279}	7.0000	0.3555
NM-1(d)	4	2.33×10^{-6}	1.41×10^{-40}	4.23×10^{-280}	7.0000	0.3633
NM-2(a)	4	2.25×10^{-6}	1.34×10^{-40}	3.65×10^{-280}	7.0000	0.3585
NM-2(b)	4	2.25×10^{-6}	1.37×10^{-40}	4.15×10^{-280}	7.0000	0.3592
NM-2(c)	4	2.25×10^{-6}	1.39×10^{-40}	4.70×10^{-280}	7.0000	0.3791
NM-2(d)	4	2.24×10^{-6}	1.05×10^{-40}	5.20×10^{-281}	7.0000	0.3467

Example 6. Consider another standard function, which is given as:

$$f_6(x) = \sin\left(\frac{x\pi}{3}\right)(e^{x^2-2x-3} - \cos(x-3) + x^2 - 9)\left(\frac{27e^{2(x-3)} - x^3}{28(x^3+1)} + x \cos \frac{x\pi}{6}\right)$$

which has a zero $\alpha = 3$ of multiplicity three. Let us choose the initial approximation $x_0 = 3.5$ for obtaining the zero of the function. Numerical results are shown in Table 6.

Table 6. Comparison of the performance of methods for Example 6.

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-Time
GKN-1(a)	4	5.04×10^{-4}	6.20×10^{-22}	2.15×10^{-129}	6.0000	3.8670
GKN-1(b)	4	9.53×10^{-4}	4.36×10^{-20}	3.98×10^{-118}	6.0000	4.1287
GKN-1(c)	4	1.37×10^{-4}	2.87×10^{-25}	2.43×10^{-149}	5.9999	3.8866
GKN-1(d)	4	2.53×10^{-3}	5.53×10^{-17}	6.03×10^{-99}	6.0000	4.5195
GKN-2(a)	5	4.22×10^{-7}	8.51×10^{-41}	9.95×10^{-81}	5.4576	5.5310
GKN-2(b)	4	7.24×10^{-3}	4.58×10^{-14}	2.94×10^{-81}	6.0000	3.9647
GKN-2(c)	4	4.43×10^{-3}	1.12×10^{-15}	2.90×10^{-91}	5.9995	3.7772
GKN-2(d)	8	8.78×10^{-10}	1.75×10^{-55}	1.09×10^{-329}	6.0000	6.2194
NM-1(a)	4	8.78×10^{-3}	1.35×10^{-15}	2.76×10^{-105}	7.0000	1.9372
NM-1(b)	4	3.50×10^{-6}	4.38×10^{-41}	2.10×10^{-285}	7.0000	1.5625
NM-1(c)	4	3.57×10^{-6}	5.15×10^{-41}	6.69×10^{-285}	7.0000	1.5662
NM-1(d)	4	1.83×10^{-6}	2.66×10^{-43}	3.70×10^{-301}	7.0000	1.5788
NM-2(a)	4	3.42×10^{-6}	3.63×10^{-41}	5.51×10^{-286}	7.0000	1.5900
NM-2(b)	4	3.50×10^{-6}	4.36×10^{-41}	2.05×10^{-285}	7.0000	1.5585
NM-2(c)	4	3.57×10^{-6}	5.13×10^{-41}	6.53×10^{-285}	7.0000	1.6405
NM-2(d)	4	1.82×10^{-6}	2.62×10^{-43}	3.30×10^{-301}	7.0000	1.3444

Example 7. Finally, considering yet another standard function:

$$f_7(x) = (\cos(x^2 + 1) - x \log(x^2 - \pi + 2) + 1)^3(x^2 + 1 - \pi).$$

The zero of function f_7 is $\alpha = 1.4632625480850 \dots$ with multiplicity four. We choose the initial approximation $x_0 = 1.3$ to find the zero of this function. Numerical results are displayed in Table 7.

Table 7. Comparison of the performance of methods for Example 7.

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-Time
GKN-1(a)	4	6.61×10^{-5}	8.80×10^{-25}	4.90×10^{-144}	6.0000	1.7305
GKN-1(b)	4	6.87×10^{-5}	1.15×10^{-24}	2.57×10^{-143}	6.0000	1.7545
GKN-1(c)	4	6.35×10^{-5}	7.67×10^{-25}	2.38×10^{-144}	6.0000	1.7150
GKN-1(d)	4	1.15×10^{-4}	8.83×10^{-23}	1.82×10^{-131}	6.0000	1.7852
GKN-2(a)	4	5.57×10^{-6}	8.57×10^{-32}	1.14×10^{-186}	6.0000	1.6405
GKN-2(b)	4	1.27×10^{-4}	1.23×10^{-22}	1.02×10^{-130}	6.0000	1.7813
GKN-2(c)	4	7.49×10^{-5}	2.89×10^{-24}	9.62×10^{-141}	6.0000	1.7382
GKN-2(d)	4	1.18×10^{-3}	9.34×10^{-17}	2.31×10^{-95}	6.0000	1.9150
NM-1(a)	4	5.19×10^{-5}	1.05×10^{-28}	1.42×10^{-194}	7.0000	1.0077
NM-1(b)	4	5.29×10^{-5}	1.23×10^{-28}	4.63×10^{-194}	7.0000	0.9062
NM-1(c)	4	5.37×10^{-5}	1.41×10^{-28}	1.23×10^{-193}	7.0000	1.0040
NM-1(d)	4	2.73×10^{-5}	7.07×10^{-31}	5.57×10^{-210}	7.0000	1.0054
NM-2(a)	4	5.14×10^{-5}	9.79×10^{-29}	8.91×10^{-195}	7.0000	0.8867
NM-2(b)	4	5.24×10^{-5}	1.16×10^{-28}	3.02×10^{-194}	7.0000	0.9802
NM-2(c)	4	5.33×10^{-5}	1.34×10^{-28}	8.30×10^{-194}	7.0000	0.9412
NM-2(d)	4	2.60×10^{-5}	5.06×10^{-31}	5.39×10^{-211}	7.0000	0.9142

It is clear from the numerical results shown in Tables 1–7 that the accuracy in the successive approximations increases as the iterations proceed. This shows the stable nature of the methods. Moreover, the present methods like that of existing methods show consistent convergence behavior. We display the value zero of $|e_n|$ in the iteration at which $|x_{n+1} - x_n| + |F(x_n)| < 10^{-350}$. The values of the computational order of convergence exhibited in the penultimate column in each table verify the theoretical order of convergence. However, this is not true for the existing methods GKN-1(a–d) and GKN-2(a) in Example 2. The entries in the last column in each table show that the new methods use less computing time than the time used by existing methods. This verifies the computationally-efficient nature of the new methods. Similar numerical tests, performed for many problems of different types, have confirmed the aforementioned conclusions to a large extent.

We conclude the analysis with an important problem regarding the choice of initial approximation x_0 in the practical application of iterative methods. The required convergence speed of iterative methods can be achieved in practice if the selected initial approximation is sufficiently close to the root. Therefore, when applying the methods for solving nonlinear equations, special care must be given for guessing close initial approximations. Recently, an efficient procedure for obtaining sufficiently close initial approximation has been proposed in [29]. For example, the procedure when applied to the function of Example 1 in the interval $[0, 1.5]$ using the statements:

```
f[x_]=x^6-12x^5+56x^4-130x^3+159x^2-98x+24; a=0; b=1.5;
k=1; x0=0.5*(a+b+Sign[f[a]]*NIntegrate[Tanh[k*f[x]],{x,a,b}])
```

in programming package *Mathematica* yields a close initial approximation $x_0 = 1.04957$ to the root $\alpha = 1$.

5. Conclusions

In the present work, we have designed a class of seventh order derivative-free iterative techniques for computing multiple zeros of nonlinear functions, with known multiplicity. The analysis of convergence shows the seventh order convergence under standard assumptions for the nonlinear function, the zeros of which we have searched. Some special cases of the class were stated. They were applied to solve some nonlinear equations and also compared with existing techniques. Comparison of the numerical results showed that the presented derivative-free methods are good competitors of the existing sixth order techniques that require derivative evaluations. The paper is concluded with the remark that unlike the methods with derivatives, the methods without derivatives are rare in the

literature. Moreover, such algorithms are good options to Newton-like iterations in the situation when derivatives are difficult to compute or expensive to obtain.

Author Contributions: Methodology, J.R.S.; writing, review and editing, J.R.S.; investigation, D.K.; data curation, D.K.; conceptualization, I.K.A.; formal analysis, I.K.A.

Funding: This research received no external funding.

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.
- Argyros, I.K.; Magreñán, Á.A. *Iterative Methods and Their Dynamics with Applications*; CRC Press: New York, NY, USA, 2017.
- Hoffman, J.D. *Numerical Methods for Engineers and Scientists*; McGraw-Hill Book Company: New York, NY, USA, 1992.
- Schröder, E. Über unendlich viele Algorithmen zur Auflösung der Gleichungen. *Math. Ann.* **1870**, *2*, 317–365. [[CrossRef](#)]
- Hansen, E.; Patrick, M. A family of root finding methods. *Numer. Math.* **1977**, *27*, 257–269. [[CrossRef](#)]
- 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](#)]
- 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. Algorithms* **2016**, *71*, 775–796. [[CrossRef](#)]
- 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](#)]
- Geum, Y.H.; Kim, Y.I.; Neta, B. A sixth-order family of three-point modified Newton-like multiple-root finders and the dynamics behind their extraneous fixed points. *Appl. Math. Comput.* **2016**, *283*, 120–140. [[CrossRef](#)]
- 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](#)]
- 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.
- Liu, B.; Zhou, X. A new family of fourth-order methods for multiple roots of nonlinear equations. *Nonlinear Anal. Model. Control* **2013**, *18*, 143–152.
- Neta, B. Extension of Murakami's high-order nonlinear solver to multiple roots. *Int. J. Comput. Math.* **2010**, *87*, 1023–1031. [[CrossRef](#)]
- 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](#)]
- Sharma, J.R.; Sharma, R. Modified Jarratt method for computing multiple roots. *Appl. Math. Comput.* **2010**, *217*, 878–881. [[CrossRef](#)]
- Soleymani, F.; Babajee, D.K.R. Computing multiple zeros using a class of quartically convergent methods. *Alex. Eng. J.* **2013**, *52*, 531–541. [[CrossRef](#)]
- 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](#)]
- Victory, H.D.; Neta, B. A higher order method for multiple zeros of nonlinear functions. *Int. J. Comput. Math.* **1983**, *12*, 329–335. [[CrossRef](#)]
- 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](#)]
- Zhou, X.; Chen, X.; Song, Y. Families of third and fourth order methods for multiple roots of nonlinear equations. *Appl. Math. Comput.* **2013**, *219*, 6030–6038. [[CrossRef](#)]
- Osada, N. An optimal multiple root-finding method of order three. *J. Comput. Appl. Math.* **1994**, *51*, 131–133. [[CrossRef](#)]
- Traub, J.F. *Iterative Methods for the Solution of Equations*; Chelsea Publishing Company: New York, NY, USA, 1982.

23. Vrscaj, 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](#)]
24. Varona, J.L. Graphic and numerical comparison between iterative methods. *Math. Intell.* **2002**, *24*, 37–46. [[CrossRef](#)]
25. Scott, M.; Neta, B.; Chun, C. Basin attractors for various methods. *Appl. Math. Comput.* **2011**, *218*, 2584–2599. [[CrossRef](#)]
26. Lotfi, T.; Sharifi, S.; Salimi, M.; Siegmund, S. A new class of three-point methods with optimal convergence order eight and its dynamics. *Numer. Algorithms* **2015**, *68*, 261–288. [[CrossRef](#)]
27. 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](#)]
28. Danby, J.M.A.; Burkardt, T.M. The solution of Kepler's equation. I. *Celest. Mech.* **1983**, *40*, 95–107. [[CrossRef](#)]
29. Yun, B.I. A non-iterative method for solving non-linear equations. *Appl. Math. Comput.* **2008**, *198*, 691–699. [[CrossRef](#)]



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).