

Article

An Efficient Class of Weighted-Newton Multiple Root Solvers with Seventh Order Convergence

Janak Raj Sharma ¹, Deepak Kumar ¹ and Carlo Cattani ^{2,3,*}

¹ Department of Mathematics, Sant Longowal Institute of Engineering & Technology, Longowal, Sangrur 148106, India

² Engineering School (DEIM), University of Tuscia, 01100 Viterbo, Italy

³ Ton Duc Thang University, Ho Chi Minh City (HCMC) 758307, Vietnam

* Correspondence: cattani@unitus.it

Received: 27 June 2019; Accepted: 13 August 2019; Published: 16 August 2019



Abstract: In this work, we construct a family of seventh order iterative methods for finding multiple roots of a nonlinear function. The scheme consists of three steps, of which the first is Newton's step and last two are the weighted-Newton steps. Hence, the name of the scheme is 'weighted-Newton methods'. Theoretical results are studied exhaustively along with the main theorem describing convergence analysis. Stability and convergence domain of the proposed class are also demonstrated by means of using a graphical technique, namely, basins of attraction. Boundaries of these basins are fractal like shapes through which basins are symmetric. Efficacy is demonstrated through numerical experimentation on variety of different functions that illustrates good convergence behavior. Moreover, the theoretical result concerning computational efficiency is verified by computing the elapsed CPU time. The overall comparison of numerical results including accuracy and CPU-time shows that the new methods are strong competitors for the existing methods.

Keywords: nonlinear equations; multiple roots; higher order methods; attraction basins

MSC: 65H05; 41A25; 49M15

1. Introduction

Finding numerically a root of an equation is an interesting and challenging problem. It is also very important in many diverse areas such as Mathematical Biology, Physics, Chemistry, Economics and Engineering, to name a few [1–4]. This is due to the fact that many problems from these disciplines are ultimately reduced to finding the root of an equation. Researchers are using iterative methods for approximating root since closed form solutions cannot be obtained in general. In particular, here we consider the problem of computing multiple roots of equation $f(x) = 0$ by iterative methods. A root (say, α) of $f(x) = 0$ is called multiple root with multiplicity m , if $f^{(j)}(\alpha) = 0$, $j = 0, 1, 2, \dots, m - 1$ and $f^{(m)}(\alpha) \neq 0$.

A basic and widely used iterative method is the well-known modified Newton's method

$$x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n)} \quad \forall n = 0, 1, 2, \dots \quad (1)$$

This method efficiently locates the required multiple root with quadratic order of convergence provided that the initial value x_0 is sufficiently close to root [5]. In terms of Traub's classification (see [1]), Newton's method (1) is called one-point method. Some other important methods that belong to this class have been developed in [6–9].

Recently, numerous higher order methods, either independent or based on the modified Newton's method (1), have been proposed and analyzed in the literature, see e.g., [10–23] and references cited therein. Such methods belong to the category of multipoint methods [1]. Multipoint iterative methods compute new approximations to root α by sampling the function $f(x)$, and its derivatives at several points of the independent variable, per each step. These methods have the strategy similar to Runge–Kutta methods for solving differential equations and Gaussian quadrature integration rules in the sense that they possess free parameters which can be used to ensure that the convergence speed is of a certain order, and that the sampling is done at some suitable points.

In particular, Geum et al. in [22,23] have proposed two- and three-point Newton-like methods with convergence order six for finding multiple roots. The two-point method [22], applicable for $m > 1$, is given as

$$\begin{aligned} y_n &= x_n - m \frac{f(x_n)}{f'(x_n)} \\ x_{n+1} &= y_n - Q(u, s) \frac{f(y_n)}{f'(y_n)} \end{aligned} \quad (2)$$

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 $Q : \mathbb{C}^2 \rightarrow \mathbb{C}$ is a holomorphic function in some neighborhood of origin $(0,0)$. The three-point method [23] for $m \geq 1$ is given as

$$\begin{aligned} y_n &= x_n - m \frac{f(x_n)}{f'(x_n)} \\ z_n &= x_n - m Q_f(u) \frac{f(x_n)}{f'(x_n)} \\ x_{n+1} &= x_n - m K_f(u, v) \frac{f(x_n)}{f'(x_n)} \end{aligned} \quad (3)$$

wherein $u = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$ and $v = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$. The function $Q_f : \mathbb{C} \rightarrow \mathbb{C}$ is analytic in a neighborhood of 0 and $K_f : \mathbb{C}^2 \rightarrow \mathbb{C}$ is holomorphic in a neighborhood of $(0,0)$. Both schemes (2) and (3) require four function evaluations to obtain sixth order convergence with the efficiency index (see [24]), $6^{1/4} \approx 1.565$.

The goal and motivation in constructing iterative methods is to attain convergence of order as high as possible by using function evaluations as small as possible. With these considerations, here we propose a family of three-point methods that attain seventh order of convergence for locating multiple roots. The methodology is based on Newton's and weighted-Newton iterations. The algorithm requires four evaluations of function per iteration and, therefore, possesses the efficiency index $7^{1/4} \approx 1.627$. This shows that the proposed methods have better efficiency (1.627) than the efficiency (1.565) of existing methods (2) and (3). Theoretical results concerning convergence order and computational efficiency are verified by performing numerical tests. In the comparison of numerical results with existing techniques, the proposed methods are observed computationally more efficient since they require less computing time (CPU-time) to achieve the solution of required accuracy.

Contents of the article are summarized as follows. In Section 2, we describe the approach to develop new methods and prove their seventh order convergence. In Section 3, stability of the methods is checked by means of using a graphical technique called basins of attraction. In Section 4, some numerical tests are performed to verify the theoretical results by implementing the methods on some examples. Concluding remarks are reported in Section 5.

2. Formulation of Method

Let $m \geq 1$ be the multiplicity of a root of the equation $f(x) = 0$. To compute the root let us consider the following three-step iterative scheme:

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

where $u = \left(\frac{f(y_n)}{f(x_n)}\right)^{\frac{1}{m}}$, $v = \left(\frac{f(z_n)}{f(x_n)}\right)^{\frac{1}{m}}$, $w = \left(\frac{f(z_n)}{f(y_n)}\right)^{\frac{1}{m}}$, and the function $H : \mathbb{C} \rightarrow \mathbb{C}$ is analytic in some neighborhood of 0 and $G : \mathbb{C}^2 \rightarrow \mathbb{C}$ is holomorphic in a neighborhood of $(0, 0)$. Notice that the first step is Newton iteration (1) whereas second and third steps are weighted by employing the factors $H(u)$ and $G(u, w)$, and so we call the algorithm (4) by the name weighted-Newton method. Factors H and G are called weight factors or more appropriately weight functions.

In the sequel we shall find conditions under which the algorithm (4) achieves high convergence order. Thus, the following theorem is stated and proved:

Theorem 1. Assume that $f : \mathbb{C} \rightarrow \mathbb{C}$ is an analytic function in a domain enclosing a root α with multiplicity m . Suppose that initial point x_0 is closer enough to the root α , then the iterative formula defined by (4) has seventh order of convergence, if the functions $H(u)$ and $G(u, w)$ verify the conditions: $H(0) = 1$, $H'(0) = 2$, $H''(0) = -2$, $G(0, 0) = 1$, $G_{10}(0, 0) = 2$, $G_{01}(0, 0) = 1$, $G_{20}(0, 0) = 0$, $|H'''(0)| < \infty$ and $|G_{11}(0, 0)| < \infty$, where $G_{ij}(0, 0) = \frac{\partial^{i+j}}{\partial u^i \partial w^j} G(u, w)|_{(0,0)}$.

Proof. Let $e_n = x_n - \alpha$ be the error at n -th iteration. Taking into account that $f^{(j)}(\alpha) = 0$, $j = 0, 1, 2, \dots, m - 1$, we have by the Taylor’s expansion of $f(x_n)$ about α

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

or:

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

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

also

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

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

Using (5) and (6) in first step of (4), it follows that

$$y_n - \alpha = \frac{C_1}{m} e_n^2 + \sum_{i=1}^5 \omega_i e_n^{i+2} + O(e_n^8), \tag{7}$$

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 with explicitly written two coefficients $\omega_1 = \frac{2mC_2 - (m+1)C_1^2}{m^2}$, $\omega_2 = \frac{1}{m^3} (3m^2C_3 + (m+1)^2C_1^3 - m(4+3m)C_1C_2)$. Here, rest of the expressions of ω_i are not being produced explicitly since they are very lengthy.

Expansion of $f(y_n)$ about α yields

$$f(y_n) = \frac{f^{(m)}(\alpha)}{m!} \left(\frac{C_1}{m}\right)^m e_n^{2m} \left(1 + \frac{2C_2m - C_1^2(m+1)}{C_1} e_n + \frac{1}{2mC_1^2} ((3 + 3m + 3m^2 + m^3)C_1^4 - 2m(2 + 3m + 2m^2)C_1^2C_2 + 4(-1 + m)m^2C_2^2 + 6m^2C_1C_3)e_n^2 + \sum_{i=1}^4 \bar{\omega}_i e_n^{i+2} + O(e_n^8)\right), \quad (8)$$

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

Using (5) and (8) in the expression of u , it follows that

$$u = \frac{C_1}{m} e_n + \frac{2C_2m - C_1^2(m+2)}{m^2} e_n^2 + \sum_{i=1}^5 \eta_i e_n^{i+2} + O(e_n^8), \quad (9)$$

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

Developing weight function $H(u)$ in neighborhood 0,

$$H(u) \approx H(0) + uH'(0) + \frac{1}{2!}u^2H''(0) + \frac{1}{2!}u^3H'''(0). \quad (10)$$

Inserting Equations (5), (8) and (10) in the second step of (4), after some simplifications we have that

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

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 accelerate convergence, the coefficients of e_n^2 and e_n^3 should be equal to zero. That is possible only if we have

$$H(0) = 1, \quad H'(0) = 2. \quad (12)$$

By using the above values in (11), we obtain that

$$z_n - \alpha = \frac{-2mC_1C_2 + C_1^3(9 + m - H''(0))}{2m^3} e_n^4 + \sum_{i=1}^3 \gamma_i e_n^{i+4} + O(e_n^8). \quad (13)$$

Expansion of $f(z_n)$ about α yields

$$f(z_n) = \frac{f^{(m)}(\alpha)}{m!} (z_n - \alpha)^m (1 + C_1(z_n - \alpha) + C_2(z_n - \alpha)^2 + O((z_n - \alpha)^3)). \quad (14)$$

From (5), (8) and (14), we obtain forms of v and w as

$$v = \frac{(9 + m)C_1^3 - 2mC_1C_2}{2m^3} e_n^3 + \sum_{i=1}^4 \tau_i e_n^{i+3} + O(e_n^8), \quad (15)$$

where $\tau_i = \tau_i(m, H''(0), H'''(0), C_1, C_2, \dots, C_7)$ and

$$w = \frac{(9 + m - H''(0))C_1^2 - 2mC_2}{2m^3} e_n^2 + \sum_{i=1}^5 \zeta_i e_n^{i+2} + O(e_n^8), \tag{16}$$

where $\zeta_i = \zeta_i(m, H''(0), H'''(0), C_1, C_2, \dots, C_7)$.

Expanding $G(u, w)$ in neighborhood of origin $(0, 0)$ by Taylor series, it follows that

$$G(u, w) \approx G_{00}(0, 0) + uG_{10}(0, 0) + \frac{1}{2}u^2G_{20}(0, 0) + w(G_{01}(0, 0) + uG_{11}(0, 0)), \tag{17}$$

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

Then by substituting (5), (6), (15)–(17) into the last step of scheme (4), we obtain that

$$e_{n+1} = \frac{1}{2m^3} ((-1 + G_{00}(0, 0))C_1(2mC_1 - (9 + m - H''(0))C_1^2))e_n^4 + \sum_{i=1}^3 \xi_i e_n^{i+4} + O(e_n^8), \tag{18}$$

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

From Equation (18) it is clear that we can obtain at least fifth order convergence when $G_{00}(0, 0) = 1$. In addition, using this value in $\xi_1 = 0$, we will obtain that

$$G_{10}(0, 0) = 2. \tag{19}$$

By using $G_{00} = 1$ and (19) in $\xi_2 = 0$, the following equation is obtained

$$\begin{aligned} & C_1(2mC_2 - C_1^2(9 + m - H''(0))) \\ & (-2mC_2(-1 + G_{01}(0, 0)) + C_1^2(-11 + m(-1 + G_{01}(0, 0)) \\ & - (-9 + H''(0))G_{01}(0, 0) + G_{20}(0, 0))) \\ & = 0, \end{aligned} \tag{20}$$

which further yields

$$G_{01}(0, 0) = 1, \quad G_{20}(0, 0) = 0 \quad \text{and} \quad H''(0) = -2. \tag{21}$$

Using the above values in (18), we obtain the error equation

$$\begin{aligned} e_{n+1} = & \frac{1}{360m^6} (360m^3((39 + 5m)C_2^3 - 6mC_3^2 - 10mC_2C_4) + 120m^3C_1((515 + 78m)C_2C_3 \\ & - 12mC_5) - 60m^2C_1^3C_3(1383 + 845m + 78m^2 + 12H'''(0)) + 10mC_1^4C_2(21571 \\ & + 8183m^2 + 558m^3 + 515H'''(0) + 324G_{11}(0, 0) + 36m(667 + 6H'''(0) + G_{11}(0, 0))) \\ & - 60m^2C_1^2(-6m(55 + 9m)C_4 + C_2^2(2619 + 1546m + 135m^2 + 24H'''(0) + 6G_{11}(0, 0))) \\ & - C_1^6(55017 + 17005m + 978m^4 + 2775H'''(0) + 7290G_{11}(0, 0) + 15m^2(4463 + 40H'''(0) \\ & + 6G_{11}(0, 0)) + 5m(21571 + 515H'''(0) + 324G_{11}(0, 0))) e_n^7 + O(e_n^8). \end{aligned} \tag{22}$$

Thus, the seventh order convergence is established. \square

Based on the conditions on $H(u)$ and $G(u, w)$ as shown in Theorem 1, we can generate numerous methods of the family (4). However, we restrict to the following simple forms:

2.1. Some Concrete Forms of $H(u)$

Case 1. Considering $H(u)$ a polynomial function, i.e.,

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

Using the conditions of Theorem 1, we get $A_0 = 1$, $A_1 = 2$ and $A_2 = -1$. Then

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

Case 2. When $H(u)$ is a rational function, i.e.,

$$H(u) = \frac{1 + A_0u}{A_1 + A_2u}.$$

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

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

Case 3. Consider $H(u)$ as another rational weight function, e.g.,

$$H(u) = \frac{1 + A_0u + A_1u^2}{1 + A_2u}.$$

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

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

Case 4. When $H(u)$ is a yet another rational function of the form

$$H(u) = \frac{1 + A_0u}{1 + A_1u + A_2u^2}.$$

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

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

2.2. Some Concrete Forms of $G(u, w)$

Case 5. Considering $G(u, w)$ a polynomial function, e.g.,

$$G(u, w) = A_0 + A_1u + A_2u^2 + A_3w.$$

From the conditions of Theorem 1, we get $A_0 = 1$, $A_1 = 2$, $A_2 = 0$ and $A_3 = 1$. So

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

Case 6. Considering $G(u, w)$ a sum of two rational functions, that is

$$G(u, w) = \frac{A_0 + 2u}{1 + A_1u} + \frac{B_0}{1 + B_1w}.$$

By using the conditions of Theorem 1, we find that $A_0 = 0$, $A_1 = 0$, $B_0 = 1$ and $B_1 = -1$. $G(u, w)$ becomes

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

Case 7. When $G(u, w)$ is a product of two rational functions, that is

$$G(u, w) = \frac{1 + A_0u}{1 + A_1u} \times \frac{B_0}{1 + B_1w}.$$

Then the conditions of Theorem 1 yield $A_0 = 2$, $A_1 = 0$, $B_0 = 1$ and $B_1 = -1$. So

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

3. Complex Dynamics of Methods

Here we analyze the complex dynamics of new methods based on a graphical tool ‘the basins of attraction’ of the roots of a polynomial $p(z)$ in Argand plane. Analysis of the basins gives an important information about the stability and convergence region of iterative methods. Wider is the convergence region (i.e., basin), better is the stability. The idea of complex dynamics was introduced initially by Vrscay and Gilbert [25]. In recent times, many authors have used this concept in their work, see, for example [26,27] and references therein. We consider some of the cases corresponding to the previously obtained forms of $H(u)$ and $G(u, w)$ of family (4) to assess the basins of attraction. Let us select the combinations: cases 1 and 2 of $H(u)$ with the cases 5, 6 and 7 of $G(u, w)$ in the scheme (4), and denote the corresponding methods by NM-i(j), $i = I, II$ and $j = a, b, c$.

To start with we take the initial point z_0 in a rectangular region $R \in \mathbb{C}$ that contains all the roots of a polynomial $p(z)$. The iterative method when starts from point z_0 in a rectangle either converges to the root $P(z)$ or eventually diverges. The stopping criterion for convergence is considered to be 10^{-3} up to a maximum of 25 iterations. If the required accuracy is not achieved in 25 iterations, we conclude that the method with initial point z_0 does not converge to any root. The strategy adopted is as follows: A color is allocated to each initial point z_0 lying in the basin of attraction of a root. If the iteration initiating at z_0 converges, it represents the attraction basin painted with assigned color to it, otherwise, the non-convergent cases are painted by the black color.

To view the geometry in complex plane, we characterize attraction basins associated with the methods NM-I(a–c) and NM-II(a–c) considering the following four polynomials:

Problem 1. Consider the polynomial $p_1(z) = (z^2 - 1)^3$, which has roots $\{-1, 1\}$ with multiplicity three. We use a grid of 400×400 points in a rectangle $R \in \mathbb{C}$ of size $[-3, 3] \times [-3, 3]$ and assign red color to each initial point in the attraction basin of root -1 and green color to each point in the attraction basin of root 1 . The basins so plotted for NM-I(a–c) and NM-II(a–c) are displayed in Figure 1. Looking at these graphics, we conclude that the method NM-II(c) possesses better stability followed by NM-I(c) and NM-II(b). Black zones in the figures show the divergent nature of a method when it starts assuming initial point from such zones.

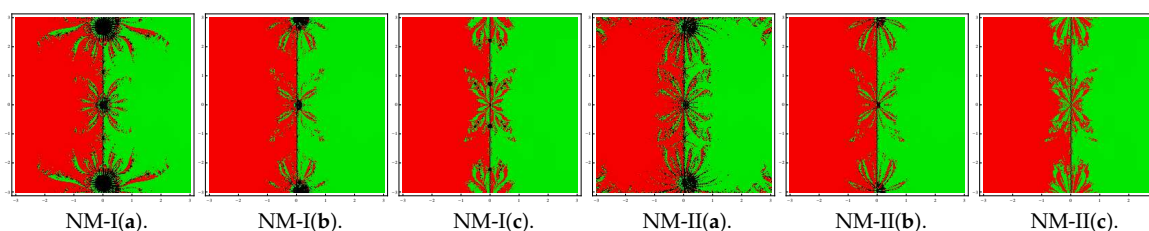


Figure 1. Basins of attraction for NM-I(a–c) and NM-II(a–c) in polynomial $p_1(z)$.

Problem 2. Let $p_2(z) = (z^3 - 1)^2$ that has three roots $\{-0.5 \pm 0.866025i, 1\}$ each with multiplicity two. To plot the graphics, we use a grid of 400×400 points in a rectangle $R \in \mathbb{C}$ of size $[-3, 3] \times [-3, 3]$ and assign the colors blue, green and red corresponding to each point in the basins of attraction of $1, -0.5 + 0.866025i$ and $-0.5 - 0.866025i$. Basins drawn for the methods NM-I(a–c) and NM-II(a–c) are shown in Figure 2. As can be observed from the pictures, the method NM-I(c) and NM-II(c) possess a small number of divergent points and therefore have better convergence than the remaining methods.

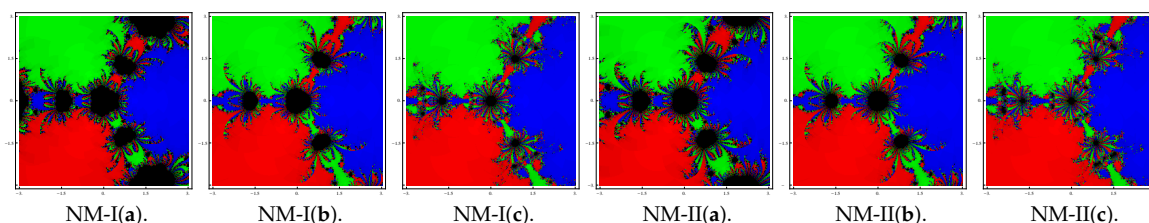


Figure 2. Basins of attraction for NM-I(a–c) and NM-II(a–c) in polynomial $p_2(z)$.

Problem 3. Let $p_3(z) = (z^6 - 1)^3$ with six roots $\{\pm 1, -0.5 \pm 0.866025i, 0.5 \pm 0.866025i\}$ each with multiplicity $m = 3$. Basins obtained for the considered methods are presented in Figure 3. To draw the pictures, the red, blue, green, pink, cyan and magenta colors have been assigned to the attraction basins of the six roots. We observe from the graphics that the method NM-I(c) and NM-II(c) have better convergence behavior since they have lesser number of divergent points. On the other hand NM-I(a) and NM-II(a) contain large black regions followed by NM-I(b) and NM-II(b) indicating that the methods do not converge in 25 iterations starting at those points.

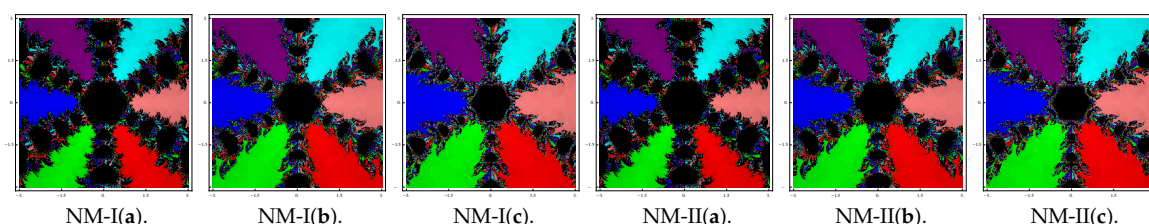


Figure 3. Basins of attraction for NM-I(a–c) and NM-II(a–c) in polynomial $p_3(z)$.

Problem 4. Consider the polynomial $p_4(z) = z^4 - 6z^2 + 8$ that has four simple roots $\{\pm 2, \pm 1.414\dots\}$. In this case also, we use a grid of 400×400 points in a rectangle $R \in \mathbb{C}$ of size $[-3, 3] \times [-3, 3]$ and allocate the red, blue, green and yellow colors to the basins of attraction of these four roots. Basins obtained for the methods are shown in Figure 4. Observing the basins, we conclude that the method NM-II(c) possesses better stability followed by NM-I(c). Remaining methods show chaotic nature along the boundaries of the attraction basins.

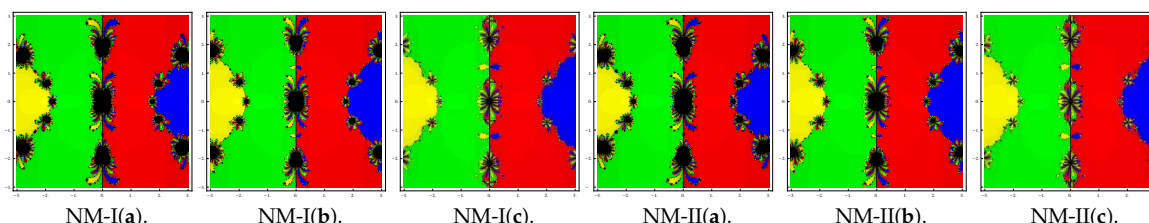


Figure 4. Basins of attraction for NM-I(a–c) and NM-II(a–c) in polynomial $p_4(z)$.

Looking at the graphics, one can easily judge the stable behavior and so the better convergence of any method. We reach to a root, if we start the iteration choosing z_0 anywhere in the basin of that root. However, if we choose an initial guess z_0 in a region wherein different basins of attraction meet

each other, it is difficult to predict which root is going to be attained by the iterative method that starts from z_0 . So, 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, which corresponds to the cases where the method is more demanding with respect to the initial point. From the basins, one can conclude that the method NM-II(c) possesses better stability followed by NM-I(c) than the remaining methods.

4. Numerical Tests

In this section, we apply the special cases NM-i(j), $i = I, II$ and $j = a, b, c$ of the scheme (4), corresponding to the combinations of $H(u)$: cases 1 and 2 with that of $G(u, v)$: cases 5, 6 and 7, to solve some nonlinear equations for validation of the theoretical results that we have derived. The theoretical seventh order of convergence is verified by calculating the computational order of convergence (COC)

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

which is given in (see [28]). Comparison of performance is also done with some existing methods such as the sixth order methods by Geum et al. [22,23], which are already expressed by (2) and (3). To represent $Q_f(u, s)$, we choose the following four special cases in the formula (2) and denote the respective methods by GKN-I(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 $a = \frac{2m}{m-1}$, $b = 2 - 2m$, $c = \frac{2(2-2m+m^2)}{m-1}$, $d = -2m(m - 1)$.
 (d) $Q_f(u, s) = \frac{m+a_1u}{1+b_1u+c_1u^2} \frac{1}{1+d_1s}$, where $a_1 = \frac{2m(4m^4-16m^3+31m^2-30m+13)}{(m-1)(4m^2-8m+7)}$, $b_1 = \frac{4(2m^2-4m+3)}{(m-1)(4m^2-8m+7)}$,
 $c_1 = -\frac{4m^2-8m+3}{4m^2-8m+7}$, $d_1 = 2(m - 1)$.

For the formula (3), considering the following four combinations of the functions $Q_f(u)$ and $K_f(u, v)$, and denoting the corresponding methods by GKN-II(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) $Q_f(u) = \frac{(2u-1)(4u-1)}{1-7u+13u^2}$, $K_f(u, v) = \frac{(2u-1)(4u-1)}{1-7u+13u^2-(1-6u)v}$.

Computational work is compiled in the programming package of Mathematica software using multiple-precision arithmetic. Numerical results as displayed in Tables 1–5 contain: (i) number of iterations (n) needed to converge to desired solution, (ii) last three successive errors $e_n = |x_{n+1} - x_n|$, (iii) computational order of convergence (COC) and (iv) CPU-time (CPU-t) in seconds elapsed during the execution of a program. Required iteration (n) and elapsed CPU-time are computed by selecting $|x_{n+1} - x_n| + |f(x_n)| < 10^{-350}$ as the stopping condition.

For numerical tests we select seven problems. The first four problems are of practical interest where as last three are of academic interest. In the problems we need not to calculate the root multiplicity m and it is set a priori, before running the algorithm.

Example 1 (Eigen value problem). *Finding Eigen values of a large sparse square matrix is a challenging task in applied mathematics and engineering sciences. Calculating the roots of a characteristic equation of matrix of order larger than 4 is even a big job. We consider the following 9×9 matrix.*

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

We calculate the characteristic polynomial of the matrix (A) as

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

This function has a multiple root $\alpha = 3$ with multiplicity 4. We select initial value $x_0 = 2.25$ and obtain the numerical results as shown in Table 1.

Table 1. Comparison of the numerical results for Example 1.

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-t (s)
GKN-I(a)	4	1.06×10^{-9}	3.86×10^{-56}	9.03×10^{-335}	6.0000	0.1567
GKN-I(b)	4	1.06×10^{-9}	3.91×10^{-56}	9.85×10^{-335}	6.0000	0.1583
GKN-I(c)	4	1.06×10^{-9}	4.34×10^{-56}	2.02×10^{-334}	6.0000	0.1525
GKN-I(d)	4	1.07×10^{-9}	1.17×10^{-55}	2.02×10^{-331}	6.0000	0.1600
GKN-II(a)	4	1.19×10^{-6}	5.39×10^{-38}	4.56×10^{-226}	5.9999	0.1835
GKN-II(b)	4	1.20×10^{-6}	1.61×10^{-37}	9.49×10^{-223}	5.9999	0.1640
GKN-II(c)	4	1.20×10^{-6}	1.12×10^{-37}	7.51×10^{-224}	5.9999	0.1718
GKN-II(d)	4	1.20×10^{-6}	1.87×10^{-37}	2.76×10^{-222}	5.9999	0.1680
NM-I(a)	3	9.83×10^{-8}	4.34×10^{-51}	0	7.0000	0.1562
NM-I(b)	3	1.16×10^{-9}	1.38×10^{-64}	0	7.0000	0.1170
NM-I(c)	3	6.30×10^{-10}	7.75×10^{-67}	0	7.0000	0.1485
NM-II(a)	3	9.83×10^{-8}	4.41×10^{-51}	0	7.0000	0.1367
NM-II(b)	3	1.16×10^{-9}	1.40×10^{-64}	0	7.0000	0.1562
NM-II(c)	3	6.30×10^{-10}	8.07×10^{-67}	0	7.0000	0.1405

Example 2 (Manning equation for fluid dynamics). Next, the problem of isentropic supersonic flow around a sharp expansion corner is chosen (see [2]). Relation among the Mach number before the corner (say M_1) and after the corner (say M_2) is given by

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

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

For a specific case, the above 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 $x = M_2$.

Let us consider this particular case seven times using same values of the involved parameters and then obtain the nonlinear function

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

The above function has one root at $\alpha = 1.8411027704\dots$ of multiplicity 7 with initial approximations $x_0 = 1.50$. Computed numerical results are shown in Table 2.

Table 2. Comparison of the numerical results for Example 2.

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-t (s)
GKN-I(a)	4	2.17×10^{-8}	4.61×10^{-25}	1.01×10^{-152}	6.0000	1.4218
GKN-I(b)	4	2.17×10^{-8}	4.60×10^{-25}	2.27×10^{-151}	6.0000	1.4923
GKN-I(c)	4	2.11×10^{-8}	4.21×10^{-25}	1.03×10^{-150}	6.0000	1.4532
GKN-I(d)	4	1.77×10^{-8}	2.48×10^{-25}	2.68×10^{-151}	6.0000	1.4960
GKN-II(a)	4	4.83×10^{-7}	1.36×10^{-41}	6.84×10^{-249}	6.0000	1.3867
GKN-II(b)	4	4.90×10^{-7}	2.89×10^{-41}	1.21×10^{-246}	6.0000	1.3790
GKN-II(c)	4	4.88×10^{-7}	2.22×10^{-41}	1.98×10^{-247}	6.0000	1.4110
GKN-II(d)	4	4.89×10^{-7}	3.22×10^{-41}	2.62×10^{-246}	6.0000	1.3982
NM-I(a)	3	1.65×10^{-8}	2.82×10^{-58}	0	7.0000	1.1367
NM-I(b)	3	7.69×10^{-9}	1.35×10^{-60}	0	7.0000	1.1915
NM-I(c)	3	3.65×10^{-9}	3.19×10^{-63}	0	7.0000	1.1407
NM-II(a)	3	1.65×10^{-9}	2.86×10^{-58}	0	7.0000	1.1290
NM-II(b)	3	7.69×10^{-9}	1.36×10^{-60}	0	7.0000	1.2540
NM-II(c)	3	3.65×10^{-9}	3.27×10^{-63}	0	7.0000	1.1445

Example 3 (Beam designing model). We consider the problem of beam positioning (see [4]) where a beam of length r unit leans against the edge of a cubical box of sides 1 unit distance each, such that one end of the beam touches the wall and the other end touches the floor, as depicted in Figure 5.

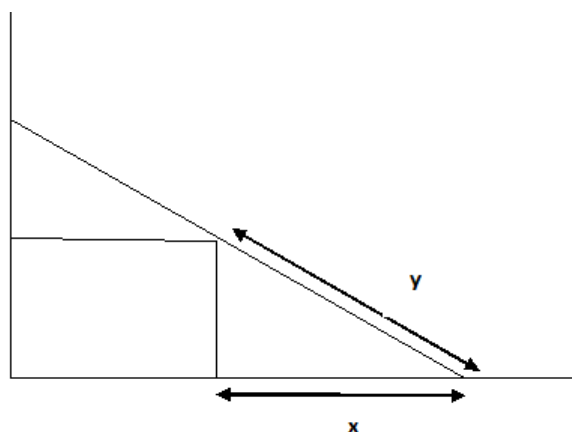


Figure 5. Beam positioning problem.

The problem is: What will be the distance alongside the floor from the base of wall to the bottom of beam? Suppose that y is distance along the beam from the floor to the edge of the box and x is the distance from the bottom of box to the bottom of beam. For a given r , we can obtain the equation

$$f_3(x) = x^4 + 4x^3 - 24x^2 + 16x + 16 = 0.$$

One of the roots of this equation is the double root $x = 2$. We select the initial guess $x_0 = 3$ to find the root. Numerical results by various methods are shown in Table 3.

Table 3. Comparison of numerical results for Example 3.

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-t (s)
GKN-I(a)	4	1.29×10^{-3}	5.18×10^{-20}	2.19×10^{-118}	6.0000	0.0313
GKN-I(b)	4	1.48×10^{-3}	1.63×10^{-19}	2.19×10^{-115}	5.9998	0.0390
GKN-I(c)	4	1.45×10^{-3}	1.76×10^{-19}	5.56×10^{-115}	5.9997	0.0352
GKN-I(d)	4	1.97×10^{-3}	1.80×10^{-18}	1.07×10^{-108}	5.9996	0.0428
GKN-II(a)	4	5.67×10^{-4}	1.20×10^{-22}	1.06×10^{-134}	5.9999	0.0314
GKN-II(b)	4	2.39×10^{-3}	5.78×10^{-18}	1.16×10^{-105}	5.9996	0.0396
GKN-II(c)	4	1.70×10^{-3}	4.26×10^{-19}	1.08×10^{-112}	5.9997	0.0392
GKN-II(d)	4	1.55×10^{-2}	5.18×10^{-13}	7.23×10^{-76}	6.0000	0.0354
NM-I(a)	4	1.13×10^{-4}	6.52×10^{-23}	1.41×10^{-157}	6.9998	0.0275
NM-I(b)	4	9.26×10^{-4}	1.63×10^{-23}	8.75×10^{-162}	6.9998	0.0313
NM-I(c)	4	4.64×10^{-4}	4.44×10^{-26}	3.23×10^{-180}	6.9998	0.0275
NM-II(a)	4	1.13×10^{-4}	6.83×10^{-23}	2.00×10^{-157}	6.9998	0.0316
NM-II(b)	4	9.33×10^{-4}	1.77×10^{-23}	1.58×10^{-161}	6.9998	0.0275
NM-II(c)	4	4.78×10^{-4}	5.86×10^{-26}	2.43×10^{-179}	6.9998	0.0354

Example 4 (van der Waals equation). Consider the Van der Waals equation

$$\left(P + \frac{a_1 n^2}{V^2}\right)(V - na_2) = nRT,$$

that describes nature of a real gas by adding in the ideal gas equation two parameters, a_1 and a_2 , which are specific for each gas. To find the volume V in terms of rest of the parameters one requires to solve the equation

$$PV^3 - (na_2P + nRT)V^2 + a_1n^2V = a_1a_2n^3.$$

Given a set of values of a_1 and a_2 of a particular gas, one can find values for n , P and T , so that this equation has three real roots. Using a particular set of values (see [3]), we have the equation

$$f_4(x) = x^3 - 5.22x^2 + 9.0825x - 5.2675 = 0,$$

where $x = V$. This equation has a multiple root $\alpha = 1.75$ with multiplicity 2. The initial guess chosen to obtain the root 1.75 is $x_0 = 2$. Numerical results are shown in Table 4.

Table 4. Comparison of numerical results for Example 4.

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-t (s)
GKN-I(a)	5	1.90×10^{-5}	9.03×10^{-22}	1.05×10^{-119}	6.0000	0.0471
GKN-I(b)	5	2.31×10^{-5}	3.69×10^{-21}	6.14×10^{-116}	6.0000	0.0472
GKN-I(c)	5	2.18×10^{-5}	3.18×10^{-21}	3.14×10^{-116}	6.0000	0.0465
GKN-I(d)	5	3.58×10^{-5}	1.01×10^{-19}	5.02×10^{-107}	6.0000	0.0483
GKN-II(a)	5	3.00×10^{-6}	4.91×10^{-27}	9.51×10^{-152}	6.0000	0.0474
GKN-II(b)	5	4.78×10^{-5}	5.42×10^{-19}	1.17×10^{-102}	6.0000	0.0472
GKN-II(c)	5	2.51×10^{-5}	6.82×10^{-21}	2.75×10^{-114}	6.0000	0.0481
GKN-II(d)	7	3.85×10^{-11}	1.78×10^{-55}	1.75×10^{-321}	6.0000	0.0625
NM-I(a)	5	1.06×10^{-5}	4.09×10^{-26}	5.33×10^{-169}	7.0000	0.0368
NM-I(b)	5	5.10×10^{-6}	2.51×10^{-28}	1.73×10^{-184}	7.0000	0.0322
NM-I(c)	5	1.15×10^{-6}	2.55×10^{-33}	6.75×10^{-220}	7.0000	0.0327
NM-II(a)	5	1.05×10^{-5}	4.13×10^{-23}	5.89×10^{-169}	7.0000	0.0316
NM-II(b)	5	5.16×10^{-6}	2.76×10^{-23}	3.48×10^{-184}	7.0000	0.0323
NM-II(c)	5	1.20×10^{-6}	3.65×10^{-26}	9.09×10^{-219}	7.0000	0.0314

Example 5. Consider now the standard nonlinear test function (see [23])

$$f_5(x) = (9 - 2x - 2x^4 + \cos 2x)(5 - x - x^4 - \sin^2 x).$$

The root $\alpha = 1.29173329244360\dots$ of multiplicity 2 is computed with initial guess $x_0 = 1.5$. Numerical results are displayed in Table 5.

Table 5. Comparison of the numerical results for Example 5.

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-t (s)
GKN-I(a)	4	1.12×10^{-4}	5.78×10^{-24}	1.10×10^{-139}	6.0000	0.2772
GKN-I(b)	4	1.55×10^{-4}	7.30×10^{-23}	8.07×10^{-133}	6.0000	0.2462
GKN-I(c)	4	1.39×10^{-4}	4.40×10^{-23}	4.43×10^{-134}	6.0000	0.2497
GKN-I(d)	4	2.32×10^{-4}	1.95×10^{-21}	6.85×10^{-124}	6.0000	0.2812
GKN-II(a)	4	3.36×10^{-5}	8.72×10^{-28}	2.66×10^{-163}	6.0000	0.3397
GKN-II(b)	4	3.39×10^{-5}	2.19×10^{-20}	1.57×10^{-117}	6.0000	0.2695
GKN-II(c)	4	2.16×10^{-5}	7.70×10^{-22}	1.58×10^{-126}	6.0000	0.2460
GKN-II(d)	4	3.51×10^{-3}	3.25×10^{-14}	2.03×10^{-80}	6.0000	0.2342
NM-I(a)	4	1.52×10^{-4}	8.45×10^{-26}	1.41×10^{-174}	6.9999	0.1445
NM-I(b)	4	1.25×10^{-4}	2.22×10^{-26}	1.23×10^{-178}	6.9999	0.1522
NM-I(c)	4	5.26×10^{-4}	1.58×10^{-29}	3.54×10^{-201}	6.9999	0.1640
NM-II(a)	4	1.52×10^{-4}	9.05×10^{-26}	2.36×10^{-174}	6.9999	0.1482
NM-II(b)	4	1.27×10^{-4}	2.49×10^{-26}	2.84×10^{-178}	6.9999	0.1492
NM-II(c)	4	5.54×10^{-4}	2.51×10^{-29}	9.82×10^{-200}	6.9999	0.1642

Example 6. Let us assume another nonlinear test function given as (see [22])

$$f_6(x) = \left(x - \sqrt{3}x^3 \cos\left(\frac{\pi x}{6}\right) + \frac{1}{x^2+1} - \frac{11}{5} + 4\sqrt{3}\right)(x-2)^4.$$

The root $\alpha = 2$ of this function is of multiplicity 5. This root is calculated assuming the initial approximation $x_0 = 1.5$. Results so obtained are shown in Table 6.

Table 6. Comparison of the numerical results for Example 6.

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-t (s)
GKN-I(a)	4	1.20×10^{-5}	6.82×10^{-31}	2.31×10^{-182}	6.0000	0.6797
GKN-I(b)	4	1.20×10^{-5}	6.86×10^{-31}	2.40×10^{-182}	6.0000	0.6680
GKN-I(c)	4	1.21×10^{-5}	7.72×10^{-31}	5.18×10^{-182}	6.0000	0.6992
GKN-I(d)	4	1.58×10^{-5}	1.00×10^{-29}	6.51×10^{-175}	6.0000	0.6720
GKN-II(a)	4	3.17×10^{-5}	1.64×10^{-28}	3.21×10^{-168}	6.0000	0.8047
GKN-II(b)	4	3.50×10^{-5}	6.90×10^{-28}	4.05×10^{-164}	6.0000	0.8280
GKN-II(c)	4	3.41×10^{-5}	4.42×10^{-28}	2.09×10^{-165}	6.0000	0.7967
GKN-II(d)	4	3.54×10^{-5}	8.45×10^{-28}	1.56×10^{-163}	6.0000	0.8242
NM-I(a)	4	5.14×10^{-6}	4.35×10^{-38}	1.35×10^{-262}	7.0000	0.5625
NM-I(b)	4	3.45×10^{-6}	2.68×10^{-39}	4.53×10^{-271}	7.0000	0.5782
NM-I(c)	4	2.05×10^{-6}	2.95×10^{-41}	3.76×10^{-285}	7.0000	0.5277
NM-II(a)	4	5.14×10^{-6}	4.42×10^{-38}	1.53×10^{-262}	7.0000	0.4805
NM-II(b)	4	3.45×10^{-6}	2.73×10^{-39}	5.24×10^{-271}	7.0000	0.4725
NM-II(c)	4	2.05×10^{-6}	3.07×10^{-41}	5.17×10^{-285}	7.0000	0.4610

Example 7. Lastly, consider the test function

$$f_7(x) = (x^2 + 1)(2xe^{x^2+1} + x^3 - x) \cosh^2\left(\frac{\pi x}{2}\right)$$

The function has multiple root $\alpha = i$ of multiplicity 4. We choose the initial approximations $x_0 = 1.25i$ for obtaining the root of the function. The results computed by various methods are shown in Table 7.

Table 7. Comparison of the numerical results for Example 7.

Methods	n	$ e_{n-3} $	$ e_{n-2} $	$ e_{n-1} $	COC	CPU-t (s)
GKN-I(a)	4	2.53×10^{-6}	3.79×10^{-35}	4.32×10^{-208}	6.0000	1.1564
GKN-I(b)	4	2.53×10^{-6}	3.92×10^{-35}	5.33×10^{-208}	6.0000	1.1577
GKN-I(c)	4	2.68×10^{-6}	6.07×10^{-35}	8.23×10^{-207}	6.0000	1.1415
GKN-I(d)	4	4.80×10^{-6}	5.34×10^{-33}	1.01×10^{-194}	6.0000	1.0473
GKN-II(a)	4	5.04×10^{-6}	1.82×10^{-33}	4.04×10^{-198}	6.0000	1.0212
GKN-II(b)	4	7.15×10^{-6}	4.23×10^{-32}	1.81×10^{-189}	6.0000	1.1215
GKN-II(c)	4	6.39×10^{-6}	1.51×10^{-32}	2.64×10^{-192}	6.0000	1.2035
GKN-II(d)	4	8.22×10^{-6}	1.41×10^{-31}	8.09×10^{-187}	6.0000	1.1416
NM-I(a)	4	1.08×10^{-6}	6.96×10^{-43}	3.13×10^{-296}	7.0000	0.5787
NM-I(b)	4	9.01×10^{-7}	1.91×10^{-43}	3.71×10^{-300}	7.0000	0.5632
NM-I(c)	4	4.64×10^{-7}	7.44×10^{-46}	2.01×10^{-317}	7.0000	0.5586
NM-II(a)	4	1.09×10^{-6}	7.21×10^{-43}	4.10×10^{-296}	7.0000	0.5478
NM-II(b)	4	9.04×10^{-7}	2.00×10^{-43}	5.10×10^{-300}	7.0000	0.5946
NM-II(c)	4	4.68×10^{-7}	8.21×10^{-46}	4.20×10^{-317}	7.0000	0.5644

From the numerical values of errors we observe the increasing accuracy in the values of successive approximations as the iteration proceed, which points to the stable nature of the methods. Like the existing methods, the convergence behavior of new methods is also consistent. At the stage when stopping criterion $|x_{n+1} - x_n| + |f(x_n)| < 10^{-350}$ has been satisfied we display the value '0' of $|x_{n+1} - x_n|$. From the calculation of computational order of convergence shown in the penultimate column in each table, we verify the theoretical convergence of seventh order. The entries of last column in each table show that the new methods consume less CPU-time during the execution of program than the time taken by existing methods. This confirms the computationally more efficient nature of the new methods. Among the new methods, the better performers (in terms of accuracy) are NM-I(c) and NM-II(c) since they produce approximations of the root with small error. However, this is not true when execution time is taken into account because if one method is better in some situations, then the other is better in some other situation. The main purpose of implementing the new methods for solving different type of nonlinear equations is purely to illustrate the better accuracy of the computed solution and the better computational efficiency than existing techniques. Similar numerical experimentation, performed on a variety of numerical problems of different kinds, confirmed the above remarks to a large extent.

5. Conclusions

In the present work, we have constructed a class of seventh order methods for solving nonlinear equations containing multiple roots. Analysis of the convergence has been carried out, which proves the seventh order of convergence under standard conditions of the function whose zeros we are looking for. Some particular cases of the family are presented. The stability of these cases are tested by means of visual display of the basins of attraction when the methods are applied on different polynomials. The methods are also implemented to solve nonlinear equations including those arising in practical problems. The performance is compared with existing methods in numerical testing. Superiority of proposed methods over the known techniques is endorsed by the numerical tests including the elapsed CPU-time in execution of program.

Author Contributions: Methodology, J.R.S.; Formal analysis, J.R.S.; Investigation, D.K.; Data Curation, D.K.; Conceptualization, C.C.; Writing—review & editing, C.C.

Funding: This research received no external funding.

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

References

1. Traub, J.F. *Iterative Methods for the Solution of Equations*; Chelsea Publishing Company: New York, NY, USA, 1982.
2. Hoffman, J.D. *Numerical Methods for Engineers and Scientists*; McGraw-Hill Book Company: New York, NY, USA, 1992.
3. Constantinides, A.; Mostoufi, N. *Numerical Methods for Chemical Engineers with MATLAB Applications*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 1999.
4. Zachary, J.L. *Introduction to Scientific Programming: Computational Problem Solving Using Maple and C*; Springer: New York, NY, USA, 2012.
5. Schröder, E. Über unendlich viele Algorithmen zur Auflösung der Gleichungen. *Math. Ann.* **1870**, *2*, 317–365. [[CrossRef](#)]
6. Chun C.; Neta, B. A third order modification of Newton's method for multiple roots. *Appl. Math. Comput.* **2009**, *211*, 474–479. [[CrossRef](#)]
7. Hansen, E.; Patrick, M. A family of root finding methods. *Numer. Math.* **1977**, *27*, 257–269. [[CrossRef](#)]
8. Neta, B. New third order nonlinear solvers for multiple roots. *App. Math. Comput.* **2008**, *202*, 162–170 [[CrossRef](#)]
9. Osada, N. An optimal multiple root-finding method of order three. *J. Comput. Appl. Math.* **1994**, *51*, 131–133. [[CrossRef](#)]
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. Liu, B.; Zhou, X. A new family of fourth-order methods for multiple roots of nonlinear equations. *Non. Anal. Model. Cont.* **2013**, *18*, 143–152.
12. 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.
13. 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](#)]
14. Sharma, J.R.; Sharma, R. Modified Jarratt method for computing multiple roots. *Appl. Math. Comput.* **2010**, *217*, 878–881. [[CrossRef](#)]
15. 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](#)]
16. Victory, H.D.; Neta, B. A higher order method for multiple zeros of nonlinear functions. *Int. J. Comput. Math.* **1983**, *12*, 329–335. [[CrossRef](#)]
17. 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](#)]
18. Soleymani, F.; Babajee, D.K.R. Computing multiple zeros using a class of quartically convergent methods. *Alex. Eng. J.* **2013**, *52*, 531–541. [[CrossRef](#)]
19. 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](#)]
20. Thukral, R. A new family of fourth-order iterative methods for solving nonlinear equations with multiple roots. *J. Numer. Math. Stoch.* **2014**, *6*, 37–44.
21. Hueso, J.L.; Martínez, E.; Teruel, C. Determination of multiple roots of nonlinear equations and applications. *J. Math. Chem.* **2015**, *53*, 880–892. [[CrossRef](#)]
22. 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](#)]
23. 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](#)]
24. Ostrowski, A.M. *Solution of Equations and Systems of Equations*; Academic Press: New York, NY, USA, 1966.

25. Vrscay, E.R.; Gilbert, W.J. Extraneous fixed points, basin boundaries and chaotic dynamics for Schröder and König rational iteration functions. *Numer. Math.* **1988**, *52*, 1–16. [[CrossRef](#)]
26. Varona, J.L. Graphic and numerical comparison between iterative methods. *Math. Intell.* **2002**, *24*, 37–46. [[CrossRef](#)]
27. Scott, M.; Neta B.; Chun, C. Basin attractors for various methods. *Appl. Math. Comput.* **2011**, *218*, 2584–2599. [[CrossRef](#)]
28. 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](#)]



© 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/>).