Derivative-Free Families of with- and without-Memory Iterative Methods for Solving Nonlinear Equations and Their Engineering Applications

Ekta Sharma 1,*, Sunil Panday 1, Shubham Kumar Mittal 1,*, Dan-Marian Joită 2,3, Lavinia Lorena Pruteanu 4 and Lorentz Jäntschi 3

1 Department of Mathematics, National Institute of Technology Manipur, Langol, Imphal 795004, Manipur, India; sunilpanday@hotmail.co.in
2 Chemistry Doctoral School, Babeș-Bolyai University, 400084 Cluj, Romania; joita.danmariant@gmail.com
3 Department of Physics and Chemistry, Technical University of Cluj-Napoca, B-dul Muncii nr. 103-105, 400641 Cluj-Napoca, Romania; lorentz.jantschi@gmail.com
4 Department of Chemistry and Biology, North University Center at Baia Mare, Technical University of Cluj-Napoca, 430122 Baia Mare, Romania; lavinia.pruteanu@cb.utcluj.ro
* Correspondence: ektsaharanit@gmail.com (E.S.); shubhammehara2@gmail.com (S.K.M.)

Abstract: In this paper, we propose a new fifth-order family of derivative-free iterative methods for solving nonlinear equations. Numerous iterative schemes found in the existing literature either exhibit divergence or fail to work when the function derivative is zero. However, the proposed family of methods successfully works even in such scenarios. We extended this idea to memory-based iterative methods by utilizing self-accelerating parameters derived from the current and previous approximations. As a result, we increased the convergence order from five to ten without requiring additional function evaluations. Analytical proofs of the proposed family of derivative-free methods, both with and without memory, are provided. Furthermore, numerical experimentation on diverse problems reveals the effectiveness and good performance of the proposed methods when compared with well-known existing methods.

Keywords: nonlinear equation; roots; efficiency index; iterative method; with-memory methods

MSC: 41A25; 65H05; 65D99

1. Introduction

Developing iterative methods to solve nonlinear equations poses an interesting and significant challenge in the fields of applied mathematics and engineering. In practice, analytical techniques often fall short in determining the roots of nonlinear problems. As a result, researchers have developed a variety of iterative methods to solve nonlinear equations.


Multi-point iterations surpass the limitations of one-point algorithms, demonstrating superior convergence rates and computational efficiency, thereby emerging as the most powerful technique for root finding. The development of iterative methods for finding the roots of nonlinear equations holds a crucial position in the field of numerical analysis, generating considerable interest and significance. The Newton–Raphson method stands as a widely renowned iterative approach that operates without memory, defined as follows [20]:

\[ f(x) = 0 \]
\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \]
\[ s_{n+1} = s_n - \frac{\Theta(s_n)}{\Theta'(s_n)}, \quad n = 0, 1, 2, \ldots \] (1)

where \( \Theta \) is the function and \( \Theta' \) is its derivative.

The Newton–Raphson method requires the evaluation of two functions in each iteration and exhibits a second-order convergence rate. It aligns with Kung–Traub’s hypothesis [21], which asserts that a memory-less multi-point method can achieve a maximum order of \( 2^{\gamma - 1} \) by performing \( \gamma \) function evaluations per iteration. The Chebyshev–Halley [21] and Ostrowski [22] methods are two iterative techniques devised for solving nonlinear equations of third and fourth orders, respectively. Researchers commonly strive to enhance the convergence rate of iterative methods. However, as the convergence rate improves, the associated increase in the number of required function evaluations may lead to a reduction in the efficiency index of these methods. The efficiency index of an iterative method quantifies its performance, as defined by [21,22]:

\[ E = \rho^{1/\gamma} \] (2)

where \( \rho \) symbolizes the convergence rate of the iterative method and \( \gamma \) denotes the number of function and derivative evaluations performed per iteration. Recent advancements in the field have witnessed remarkable contributions to the development of iterative techniques for solving nonlinear equations. Kumar et al. [23] introduced a derivative-free fifth-order method, while Choubey et al. [24] presented an eighth-order approach that removes the derivatives by employing techniques such as divided differences and weight functions. Sharma et al. [25] proposed an optimal fourth-order iterative method that incorporates derivatives, and Panday et al. [26] formulated both fourth and eighth-order optimal iterative approaches. Singh and Singh [27] devised an optimal eighth-order approach employing the modified Halley’s method. Chanu and Panday [29] contributed a non-optimal tenth-order method for solving nonlinear equations. The methods developed in [22,30,31] required evaluation of the first and second-order derivatives, which is a cumbersome task. Also, when the derivative value is zero, the application of these methods is not possible. The exploration of nonlinear equation solving has also led to the formulation of derivative-free methods with memory, as showcased by B. Neta [32], who utilized Traub’s method and Newton’s method. Furthermore, Chanu et al. [33] proposed optimal memory-less techniques of fourth and eighth orders, extending them to incorporate memory. In the pursuit of resolving nonlinear equations with multiple roots, Thangkhempau et al. [34] introduced a novel scheme that offers both with- and without-memory-based variants.

In this research article, we present a novel fifth-order derivative-free iterative method for solving simple roots of nonlinear equations. Moreover, it excels in scenarios where the derivative is zero at initial or successive iterative approximations. The method exhibits an efficiency index of \( 5^{1/4} = 1.4953 \). Building upon this foundation, we extend the technique to a tenth-order method with memory by incorporating self-accelerating parameters without requiring any additional function evaluation. The efficiency index of the tenth-order method with memory is \( 10^{1/4} = 1.7783 \). The subsequent sections of this document are meticulously organized to provide comprehensive insights. Section 2 delves into the utilization of divided difference and weight function techniques in the formulation of the methods, while also analyzing the convergence rates for both with- and without-memory approaches. Section 3 presents a thorough examination of numerical tests, meticulously comparing the proposed method with other well-established techniques. Finally, Section 4 concludes the study, offering a comprehensive summary of the findings and their implications.
2. Construction of New Iterative Schemes and Their Convergence Analysis

In this section, we have developed novel iterative techniques of both fifth and tenth orders, which are specifically designed for solving nonlinear equations without the need for derivatives. The new three-step fifth-order iterative without-memory method is outlined below:

\[ y_n = s_n - \frac{\Theta(s_n)}{\Theta(s_n, w_n) + \alpha \Theta(w_n)}, \quad w_n = s_n + \beta \Theta(s_n), \quad \alpha, \beta \in R, \]

\[ z_n = y_n - \left( 1 + \frac{\Theta(y_n)}{\Theta(s_n) - 2 \Theta(y_n)} \right) \left( \frac{\Theta(y_n)}{\Theta(s_n, w_n)} \right), \]

\[ s_{n+1} = z_n - \frac{\Theta(z_n)}{\Theta(s_n, w_n)} (P(t_n) + K(u_n, v_n)), \quad (3) \]

where \( \Theta(s, w) = \frac{\Theta(s) - \Theta(w)}{s - w} \) and \( P : \mathbb{C} \to \mathbb{C} \) is an analytic function in the neighbourhood of \( 0 \) with \( t_n = \Theta(z_n) \) and \( K : \mathbb{C} \times \mathbb{C} \to \mathbb{C} \) is another analytic function in the neighbourhood of \( (0,0) \) with \( u_n = \Theta(y_n) \) and \( v_n = \Theta(z_n) \). This new family \((3)\) requires four function evaluations at each iteration.

**Theorem 1.** Suppose that \( \Theta : D \subseteq \mathbb{C} \to \mathbb{C} \) is an analytic function that is sufficiently differentiable. Let \( \xi \in D \) be a simple root of \( \Theta \), and \( \xi_0 \) be a value that is sufficiently near to \( \xi \). If \( P(t_0) \) and \( K(u_0, v_0) \) satisfy the conditions outlined below, the iterative process defined in \((3)\) is of fifth-order convergence. The conditions are as follows: \( P(0) = 0, P'(0) = 1, P''(0) = 2, P'''(0) = 6, P''''(0) = 24, P''''\prime(0) = 120, K(0,0) = 1, K^{(1,0)}(0,0) = 1, \) and \( K^{(0,1)}(0,0) = 2 \). Furthermore, the iterative process \((3)\) fulfills the error equation below:

\[ e_{n+1} = -\alpha(1 + \Theta'(\xi)\beta)^3(a + \Theta'(\xi)\alpha \beta - c_2)(a + c_2)^2 e_n^3 + O(e_n^5) \quad (4) \]

where \( c_j = \frac{\Theta^{(j)}(\xi)}{\Theta'(\xi)^j}, j = 2, 3, 4, \ldots \)

**Proof of Theorem 1.** Let \( \xi \) be the simple root of \( \Theta(s) = 0 \) and let \( e_n = s_n - \xi \) be the error of \( n^{th} \) iteration. Using Taylor expansion, we obtain

\[ \Theta(s_n) = \Theta'(\xi) \left[ e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + O(e_n^6) \right] \quad (5) \]

where \( e_n = s_n - \xi \) and \( c_j = \frac{\Theta^{(j)}(\xi)}{\Theta'(\xi)^j}, j = 2, 3, 4, \ldots \)

Now, using Equation \((5)\) in the first step of the method given by \((3)\), we have

\[ w_n - \xi = (1 + \Theta'(\xi)\beta)e_n + \Theta'(\xi)\beta c_2 e_n^2 + \Theta'(\xi)\beta c_3 e_n^3 + \Theta'(\xi)\beta c_4 e_n^4 + \Theta'(\xi)\beta c_5 e_n^5 + O(e_n^6) \quad (6) \]

and

\[ \Theta(w_n) = \Theta'(\xi) \left( 1 \right) e_n + \Theta'(\xi)\beta c_2 e_n^2 + \Theta'(\xi)\beta c_3 e_n^3 + \Theta'(\xi)\beta c_4 e_n^4 + \Theta'(\xi)\beta c_5 e_n^5 + O(e_n^6) \quad \text{(7)} \]
Now, using Equations (5)–(7), the divided difference $\Theta[n, m]$ can be expressed as:

$$
\Theta[n, m] = \Theta'(\xi) + \Theta''(\xi)\frac{c_2}{2} + c_m + \Theta'(\xi)(\Theta''(\xi)\frac{c_2^2}{2} + c_3(\Theta''(\xi)\beta(\Theta'\xi)\beta + 3))c_n^2 + c_6(\Theta''(\xi)\beta(\Theta'\xi)\beta + 3)\beta + 2 + (2\Theta''(\xi)\beta c_2 + c_4(\Theta''(\xi)\beta(\Theta'\xi)\beta + 2))c_n^2 + c_6(\Theta''(\xi)\beta(\Theta'\xi)\beta(\Theta''(\xi)\beta + 8) + 7 + c_5(\Theta''(\xi)\beta(\Theta'\xi)\beta(\Theta''(\xi)\beta + 5) + 10) + 10)) + 5c_3\epsilon_n^4
$$

By using Equations (5)–(6) in the second step of the method (3), we obtain

$$
y_n - \xi = (1 + \Theta'(\xi)\beta)(a + c_2)c_n^2 + (-a + \Theta'(\xi)\beta(2 + \Theta''(\xi)\beta))c_2(a + c_2) + 2c_3 + \Theta'(\xi)(3 + \Theta''(\xi)\beta)c_3c_n^2 + \ldots + O(\epsilon_n^6)
$$

and

$$
\Theta(y_n) = \Theta'(\xi)(1 + \Theta'(\xi)\beta)(a + c_2)c_n^2 + \Theta''(\xi)(-a + \Theta'(\xi)\beta\beta)^2 - (2 + \Theta''(\xi)\beta(2 + \Theta'(\xi)\beta))c_2(a + c_2)
$$

Now, after using Equations (5) to (9), we obtain the third step $z_n$ and $\Theta(z_n)$ as

$$
z_n - \xi = - (1 + \Theta'(\xi)\beta)(a + \Theta'(\xi)\beta - c_2)(a + c_2)c_n^2 + \ldots + O(\epsilon_n^6)
$$

$$
\Theta(z_n) = \Theta'(\xi)(1 + \Theta'(\xi)\beta)(a + \Theta'(\xi)\beta - c_2)(a + c_2)c_n^2 + \ldots + O(\epsilon_n^6)
$$

Now, we obtain $t_n = \frac{\Theta(z_n)}{\Theta(z_m)}$, $u_n = \frac{\Theta(y_n)}{\Theta(z_m)}$, and $v_n = \frac{\Theta(y_n)}{\Theta(z_n)}$ by using Equations (5), (10) and (12); thus, we have

$$
t_n = \left(-a(1 + \Theta'(\xi)\beta)\beta + c_2\epsilon_n^2 + (-a + \Theta'(\xi)\beta\beta)^2(3 + 2\Theta''(\xi)\beta)c_2
$$

$$
- (2 + \Theta'(\xi)\beta)(2 + \Theta'(\xi)\beta))c_2 + c_3)c_n^2 + \ldots + O(\epsilon_n^6)
$$

and

$$
v_n = - (1 + \Theta'(\xi)\beta)(a + \Theta'(\xi)\beta - c_2)(a + c_2)c_n^2 + \ldots + O(\epsilon_n^6)
$$

Now, by using Equations (5)–(15), we obtain the fourth step as

$$
s_{n+1} - \xi = c_{n+1} = (1 + \Theta'(\xi)\beta)P(0)(a + \Theta'(\xi)\beta - c_2)(a + c_2)c_n^2 + \gamma_4\epsilon_n^4 + \gamma_5\epsilon_n^4 + O(\epsilon_n^6)
$$

where

$$
\gamma_4 = \frac{1}{2}(c_2(c_2^2 - 2P(0)(a^2\Theta'(\xi)\beta^2(\Theta''(\xi)\beta)(\Theta''(\xi)\beta(\Theta''(\xi)\beta + 11) + 11 + 9) + c_3(\Theta''(\xi)\beta + 1))\Theta''(\xi)\beta(\Theta''(\xi)\beta + 1)\Theta''(\xi)\beta (\Theta''(\xi)\beta + 1)\Theta''(\xi)\beta (\Theta''(\xi)\beta + 3) + 2(P(0) - 1) + 3(P(0) - 1)) + 2(0 + 2P(0) - 1) + a(\Theta''(\xi)\beta + 1) + c_3P(0)(\Theta''(\xi)\beta(\Theta''(\xi)\beta + 3) + 1 - (P(0) - 1) + a + a(\Theta''(\xi)\beta^2)) + c_2(\Theta''(\xi)\beta P(0)(\Theta''(\xi)\beta(\Theta''(\xi)\beta + 5) + 9) - P(0) + 1) - P(0) + 6P(0) + 1)
$$

and

$$
\gamma_5 = \frac{1}{2}(c_2^2 - 2P(0)(a^2\Theta'(\xi)\beta^2(\Theta''(\xi)\beta)(\Theta''(\xi)\beta(\Theta''(\xi)\beta + 11) + 11 + 9) + c_3(\Theta''(\xi)\beta + 1))\Theta''(\xi)\beta(\Theta''(\xi)\beta + 1)\Theta''(\xi)\beta (\Theta''(\xi)\beta + 1)\Theta''(\xi)\beta (\Theta''(\xi)\beta + 3) + 2(P(0) - 1) + 3(P(0) - 1)) + 2(0 + 2P(0) - 1) + a(\Theta''(\xi)\beta + 1) + c_3P(0)(\Theta''(\xi)\beta(\Theta''(\xi)\beta + 3) + 1 - (P(0) - 1) + a + a(\Theta''(\xi)\beta^2)) + c_2(\Theta''(\xi)\beta P(0)(\Theta''(\xi)\beta(\Theta''(\xi)\beta + 5) + 9) - P(0) + 1) - P(0) + 6P(0) + 1)
$$
which evolve as the iteration progresses according to the formulas

\[
(4\Theta'(\xi)\beta + 5)P'(0) + 36\Theta'(\xi)\beta P(0) + 3P(0) + 3) + (\Theta'(\xi)\beta - 2)(\Theta'(\xi)\beta + 1)P''(0) + 4P'(0) - 2 - 2c_4P(0)(\Theta'(\xi)\beta(\Theta'(\xi)\beta + 3) + 4) + 4c_3(\Theta'(\xi)\beta(\Theta'(\xi)\beta + 3) + 4)) + 4c_3(\Theta'(\xi)\beta(\Theta'(\xi)\beta + 3)(\Theta'(\xi)\beta P(0)(\Theta'(\xi)\beta + 2) + P'(0) = 1)) + 2P'(0) = P'(0) - 2) + 2P'(0) = P'(0) - 2)) + (\Theta'(\xi)\beta + 1)(2\alpha c_3(\Theta'(\xi)\beta(\Theta'(\xi)\beta(\Theta'(\xi)\beta - (\Theta'(\xi)\beta + 4)P'(0) + \Theta'(\xi)\beta P(0) + 2P(0) + 4) - 3P'(0) - P'(0) + 3) - 4P(0) + 2c_4P(0)(\Theta'(\xi)\beta(\Theta'(\xi)\beta (\Theta'(\xi)\beta + 4) + 2) - 2c_4P(0)(\Theta'(\xi)\beta + 2) + \alpha^2(\Theta'(\xi)\beta + 1)^2 - 3P'(0) + P''(0) + 2P(0) - 2)) + \alpha^2(2\Theta'(\xi)\beta(\Theta'(\xi)\beta(\Theta'(\xi)\beta - 2(\Theta'(\xi)\beta + 3)P'(0) - 10P'(0) + 7) - 10P'(0) - 33P(0) + 10) - 11P'(0) + 37P(0) - 8(\Theta'(\xi)\beta + 1)(3\Theta'(\xi)\beta + 2)P''(0) - 8P'(0) - 32P(0) + 6 - (c_4(\Theta'(\xi)\beta(2 \Theta'(\xi)\beta(\Theta'(\xi)\beta - P(0)(2\Theta'(\xi)\beta - 15) - 2P'(0) = 2)) - 8P'(0) + 39P(0) + 8) - 26P'(0) + P''(0) + 96P(0) + 24) - 16P'(0) + P''(0) + 50P(0) + 14) ) \).  

After substituting the values \( P(0) = 0, P'(0) = 1 \) and \( P''(0) = 2 \), we have derived the error equation below:

\[
s_{n+1} - \xi = e_{n+1} = -a(1 + \Theta'(\xi)\beta)^3(a + \Theta'(\xi)\beta - c_2)(a + c_2)^3 + O(e_n^5) \quad (17)
\]

By examining Equation (17), we can infer that the method described by Equation (3) exhibits a fifth-order convergence. This conclusion serves as the completion of the proof.

Based on the conditions for \( P(t_n) \) and \( K(u_n, v_n) \), as presented in Theorem 1 we are adopting the particular forms for the weight functions: \( P(t_n) = t_n^2 + t_n^2 + t_n^2 + t_n \) and \( K(u_n, v_n) = u_n + 2v_n + 1 \) within the newly proposed method described by Equation (3).

Parametric Family of Three-Point with-Memory Method and its Convergence Analysis

We shall now proceed to enhance the method described by Equation (3) by incorporating the with-memory feature, introducing two additional parameters. Upon analyzing Equation (4), it becomes evident that the convergence order of the method (3) reaches ten when \( \alpha = -c_4 \) and \( \beta = -\frac{1}{2\Theta'(\xi)} \). By selecting \( \alpha = -c_4 = -\Theta''(\xi) \) and \( \beta = -\frac{1}{2\Theta'(\xi)} \), we can transform the error Equation (4) into the following form:

\[
e_{n+1} = c_2^3(5c_2^2 - c_3)c_2^2c_2^10 + O(e_n^{11})
\]

To derive the method with memory, we introduce the parameters \( \alpha_n = \alpha_n \) and \( \beta_n = \beta_n \), which evolve as the iteration progresses according to the formulas \( \alpha_n = -\frac{\Theta''(\xi)}{2\Theta'(\xi)} \) and \( \beta_n = -\frac{1}{\Theta'(\xi)} \). In the method (3), we make use of the following approximation:

\[
\begin{align*}
\alpha_n &= -\frac{\Theta''(\xi)}{2\Theta'(\xi)} \approx \frac{N'_n(w_n)}{2N'_n(w_n)} \\
\beta_n &= -\frac{1}{\Theta'(\xi)} \approx -\frac{1}{N'_n(s_n)}
\end{align*}
\]

Let us define Newton’s interpolating polynomials of fourth and fifth degrees as follows:

\[
N_4(u) = N_4(u; s_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1}) \quad \text{and} \quad N_5(u) = N_5(u; w_n, s_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1})
\]

Now, we can express the iterative method with memory as follows:

\[
\begin{align*}
w_n &= s_n + \beta_n \Theta(s_n) \\
y_n &= s_n - \frac{\Theta(s_n)}{\Theta(s_n, w_n) + \alpha_n \Theta(w_n)} \\
z_n &= y_n - \frac{1}{\Theta(s_n) - 2\Theta(y_n)} \left( \frac{\Theta(y_n)}{\Theta(s_n, w_n)} \right) \\
s_{n+1} &= z_n - \frac{\Theta(z_n)}{\Theta(s_n, w_n)} (P(t_n) + K(u_n, v_n)).
\end{align*}
\]
Remark 1. It is important to note that the approach of iteratively calculating independent parameters, as employed in this method, is commonly known as a self-accelerating method. Prior to initiating the iterative process, it is crucial to determine the initial values of \(a_0\) and \(b_0\), as highlighted in [35].

Our objective is to analyze the convergence properties of the method with memory. Specifically, we are interested in investigating the behavior of the sequence \(s_n\) as it converges to the root \(\xi\) of \(\Theta\), as well as determining the rate at which this convergence occurs. To quantify the convergence rate, we define the difference between the initial estimate and \(s\), which is given by \(e_n = s_n - \xi\). If the sequence \(s_n\) converges to the root \(\xi\), we can express it as \(e_{n+1} \sim e_n^p\). In order to establish the convergence order of the method (20), we can make use of the following lemma, which has been presented in [36].

**Lemma 1.** If \(s_n = -\frac{N_n}{N_{n+1}}\) and \(b_n = -\frac{1}{N_{n+1}(s_n)}\), then the estimates \(1 + \Theta'(\xi)\) hold.

To derive the following lemma, which has been presented in [36].

**Theorem 2.** If the initial estimate \(s_0\) is in close vicinity to the unique root \(\xi\) of the real and suitably smooth function \(\Theta(s) = 0\), method (20) will possess a convergence rate of at least 10.8151.

**Proof of Theorem 2.** Let us assume that the iterative process described in (20) produces a sequence of estimations denoted as \(s_n\). If this sequence converges to the root \(\xi\) of the function \(\Theta\) with a convergence order of \(q\), we can deduce the following:

\[
e_{n+1} \sim e_n^q, \quad \text{where} \quad e_n = s_n - \xi
\]  

\[
e_{n+1} \sim (e_n^q) = e_{n-1}^{q+1}
\]  

Let us assume that the iterative sequences \(w_n, y_n\) and \(z_n\) have the order \(q_1, q_2\) and \(q_3\), respectively. Then, Equations (21) and (22) give the following:

\[
e_{n,w} \sim (e_n^{q_1}) = e_{n-1}^{q_1+1}
\]  

\[
e_{n,y} \sim (e_n^{q_2}) = e_{n-1}^{q_2+1}
\]  

\[
e_{n,z} \sim (e_n^{q_3}) = e_{n-1}^{q_3+1}
\]  

By Theorem 1, we can write

\[
e_{n,w} \sim (1 + \Theta'(\xi)\beta) e_n
\]  

\[
e_{n,y} \sim (1 + \Theta'(\xi)\beta) (\alpha + c_2) e_n^2
\]  

\[
e_{n,z} \sim (1 + \Theta'(\xi)\beta) (\alpha + \Theta'(\xi)\alpha\beta - c_2) (\alpha + c_2) e_n^3
\]  

\[
e_{n+1} \sim -\alpha(1 + \Theta'(\xi)\beta) (\alpha + \Theta'(\xi)\alpha\beta - c_2) (\alpha + c_2)^2 e_n^5
\]  

Using Lemma 1, we obtain the following:

\[
e_{n,w} \sim (1 + \Theta'(\xi)\beta) e_n \sim (e_{n-1,w}e_{n-1,y}e_{n-1,z}) e_n \sim e_{n-1}^{q_1+q_2+q_3+1}
\]


$$e_{n,y} \sim (1 + \Theta'(\xi)\beta)(a + c_2)e_{n-1}^2 \sim \left(e_{n-1}e_{n-1}e_{n-1}e_{n-1}e_{n-1}\right)^2 e_{n-1}^2 \sim e_{n-1}^{2q+2q_1+2q_2+2q_3+2}$$ (31)

$$e_{n,z} \sim -\left(1 + \Theta'(\xi)\beta\right)(a + \Theta'(\xi)\alpha - c_2)(a + c_2)e_{n-1}^3 \sim \left(e_{n-1}e_{n-1}e_{n-1}e_{n-1}e_{n-1}\right)^3 e_{n-1}^3 \sim e_{n-1}^{3q+3q_1+3q_2+3q_3+3}$$ (32)

$$e_{n+1} \sim -\alpha\left(1 + \Theta'(\xi)\beta\right)^3(a + \Theta'(\xi)\alpha - c_2)(a + c_2)^2 e_{n}^5 \sim \left(e_{n-1}e_{n-1}e_{n-1}e_{n-1}e_{n-1}\right)^4 e_{n}^5 \sim e_{n-1}^{5q+4q_1+4q_2+4q_3+4}$$ (33)

Comparing the power of $e_{n-1}$ of Equations (23)–(30), (24)–(31), (25)–(32) and (22)–(33), we obtain the following system of equations:

$$qq_1 - q - q_1 - q_2 - q_3 - 1 = 0$$
$$qq_2 - 2q - 2q_1 - 2q_2 - 2q_3 - 2 = 0$$
$$qq_3 - 3q - 3q_1 - 3q_2 - 3q_3 - 3 = 0$$
$$q^2 - 5q - 4q_1 - 4q_2 - 4q_3 - 4 = 0$$ (34)

After, solving the above system of equations, we obtain $q_1 = 2.4538$, $q_2 = 4.9075$, $q_3 = 7.3613$ and $q = 10.8151$. Thus, the proof is complete. □

3. Numerical Discussion

In this section, our objective is to elucidate the efficacy of recently introduced iterative families of methods through their application to various nonlinear equations. We will compare the results with some well-known existing methods available in the literature. In particular, the following iterative methods, in addition to Newton’s method, are considered for comparison.

The well-known fourth-order multipoint without-memory Ostrowski’s method (OM) is given as [22]:

$$y_n = s_n - \frac{\Theta(s_n)}{\Theta'(s_n)}$$

$$z_n = y_n + \frac{\Theta(s_n)}{\Theta(s_n) - 2\Theta(y_n)} \frac{\Theta(y_n)}{\Theta'(s_n)}.$$ (35)

In 2020, Nouri et al. [30] developed the following fifth-order method (NRTM):

$$y_n = s_n - \frac{\Theta(s_n)}{\Theta'(s_n)}$$

$$z_n = s_n + \frac{\Theta(s_n)}{\Theta'(s_n)},$$

$$s_{n+1} = s_n - \frac{(s_n - y_n)\Theta(s_n)^2\Theta(y_n) + \Theta(z_n)}{\Theta(s_n)^2(\Theta(z_n) - \Theta(y_n)) - 4\Theta(s_n)\Theta(y_n)^2 - 6\Theta(y_n)^3}.$$ (36)
The fifth-order method (GM) developed by Grau et al. [31]:

\[
\begin{align*}
    y_n &= s_n - \frac{\Theta(s_n)}{\Theta'(s_n)^2}, \\
    z_n &= s_n - \left(1 + \frac{\Theta''(s_n) \Theta(s_n)}{\Theta'(s_n)^2}\right) \frac{\Theta(s_n)}{\Theta'(s_n)^2}, \\
    s_{n+1} &= s_n - \left(1 + \frac{\Theta''(s_n) (\Theta(s_n) + \Theta(z_n))}{2\Theta'(s_n)^2}\right) \frac{\Theta(s_n) + \Theta(z_n)}{\Theta'(s_n)}. \tag{37}
\end{align*}
\]

We selected the method [30,31] for comparison so as to have a fair and uniform comparison. This is because, like our proposed methods, they have the same order of convergence and the same number of function evaluations per iteration.

The nonlinear test functions utilized for comparative analysis, along with their corresponding initial approximations, are provided below.

**Example 1:**

\[\Theta_1(s) = e^{-s^2} (1 + s^3 + s^6)(s - 2), \quad s_0 = 2.1\]

**Example 2:**

\[\Theta_2(s) = \sin^2 s + s, \quad s_0 = -0.07\]

**Example 3:**

\[\Theta_3(s) = 10s e^{-s^2} - 1, \quad s_0 = 1.7\]

**Example 4:**

\[\Theta_4(s) = e^{-s^2 + s + 2} - 1, \quad s_0 = 2.001\]

**Example 5:**

\[\Theta_5(s) = s^2 + 1, \quad s_0 = \frac{2}{3}\]

**Example 6:**

\[\Theta_6(s) = s^2 + s + 1, \quad s_0 = \sqrt{3}i - 2\]

**Example 7:** We consider the following Planck’s radiation law problem, which calculates the energy density within an isothermal blackbody and is given by [37]:

\[v(\lambda) = \frac{8\pi c P \lambda^{-5}}{e^{\frac{c P}{\lambda^2}} - 1}\]

where \(\lambda\) is the wavelength of the radiation, \(T\) is the absolute temperature of the blackbody, \(B\) is the Boltzmann constant, \(P\) is the Planck constant and \(c\) is the speed of light. We are interested in determining the wavelength \(\lambda\) that corresponds to maximum energy density \(v(\lambda)\).

From (38), we obtain

\[v'(\lambda) = \left(\frac{8\pi c P \lambda^{-6}}{e^{\frac{c P}{\lambda^2}} - 1}\right) \left(\frac{e^{\frac{c P}{\lambda^2}} e^{\frac{s^2}{\lambda^2}}}{e^{\frac{c P}{\lambda^2}} - 1} - 5\right)\]

so that the maxima of \(v\) occur when

\[\frac{e^{\frac{c P}{\lambda^2}} e^{\frac{s^2}{\lambda^2}}}{e^{\frac{c P}{\lambda^2}} - 1} = 5\]

After that, if \(s = \frac{e^{\frac{c P}{\lambda^2}}}{e^{\frac{c P}{\lambda^2}} - 1}\), then (40) is satisfied if

\[\Theta_7(s) = e^{-s} + \frac{s}{5} - 1; \quad s_0 = 4.965\]

Therefore, the solutions of \(\Theta_7(s) = 0\) give the maximum wavelength of radiation \(\lambda\) by means of the following formula:

\[\lambda \approx \frac{c P}{s^* B T}\]

where \(s^*\) is a solution of (41).

**Example 8:** In the study of the multi-factor effect, the trajectory of an electron in the air gap
between two parallel plates is given by [37]

\[ s(t) = s_0 + \left( v_0 + e \frac{E_0}{m \omega} \sin(\omega t_0 + a) \right) (t - t_0) \\
+ e \frac{E_0}{m \omega^2} \left( \cos(\omega t + a) + \sin(\omega + a) \right) \]  \hspace{1cm} (43)

where \( e \) and \( m \) are the charge and the mass of the electron at rest, \( s_0 \) and \( v_0 \) are the position and velocity of the electron at the time \( t_0 \) and \( E_0 \sin(\omega t + a) \) is the RF electric field between the plates. We choose the particular parameters in the expression in order to deal with a simpler expression, which is defined as follows:

\[ \Theta_8(s) = s - \frac{1}{2} \cos(s) + \frac{\pi}{4} \]  \hspace{1cm} (44)

with \( s_0 = -0.309 \).

The comparative results for all the methods, including NM (1), NDM1 (3), NDM2 (20), NTRM (36) and GM (37), are summarized in Tables 1–8. In these tables, we have presented the following metrics for each of the compared methods after the first three full iterations: the approximated roots \( (s_n) \), absolute residual error \(|\Theta(s_n)|\), the difference between the last two successive iterations \(|s_n - s_{n-1}|\) and the computational rate of convergence (COC). Also in Figure 1, we provide a comparison of methods based on the error in the consecutive iterations, \(|s_n - s_{n-1}|\), after the first three iterations. The determination of COC is achieved using the following equation [38]:

\[ \text{COC} = \frac{\log|\Theta(s_n)/\Theta(s_{n-1})|}{\log|\Theta(s_{n-1})/\Theta(s_{n-2})|} \]  \hspace{1cm} (45)

For all numerical calculations, the programming software Mathematica 12.2 was utilized. For the with-memory method NDM2, we have selected the parameter values \( \alpha_0 = 1 \) and \( \beta_0 = 0.01 \) to start the initial iteration.

### Table 1. Comparisons of without-memory and with-memory methods after first three \((n = 3)\) iterations for \( \Theta_1(s) \).

| Method   | \(|s_1 - s_0|\)       | \(|s_2 - s_1|\)       | \(|s_3 - s_2|\)       | \(|\Theta(s_3)|\)     | COC     |
|----------|----------------------|----------------------|----------------------|----------------------|---------|
| NM       | 0.11779 1.7434 \times 10^{-1} | 3.5679 \times 10^{-4} | 2.0508 \times 10^{-7} | 2.0000              |
| OM       | 0.10048 4.8190 \times 10^{-4} | 1.4662 \times 10^{-13} | 1.6842 \times 10^{-51} | 4.0000              |
| NTRM     | 0.099122 8.7833 \times 10^{-4} | 8.2482 \times 10^{-15} | 7.9485 \times 10^{-70} | 5.0000              |
| GM       | 0.098037 1.9631 \times 10^{-3} | 1.2383 \times 10^{-12} | 1.6115 \times 10^{-58} | 5.0000              |
| NDM1 \((\alpha = 1, \beta = 1)\) | 0.10105 1.0533 \times 10^{-3} | 1.3616 \times 10^{-15} | 7.0363 \times 10^{-75} | 5.0000              |
| NDM2     | 0.99749 2.5101 \times 10^{-4} | 6.0861 \times 10^{-35} | 5.7271 \times 10^{-341} | 10.0000             |

### Table 2. Comparisons of without-memory and with-memory methods after first three \((n = 3)\) iterations for \( \Theta_2(s) \).

| Method   | \(|s_1 - s_0|\)       | \(|s_2 - s_1|\)       | \(|s_3 - s_2|\)       | \(|\Theta(s_3)|\)     | COC     |
|----------|----------------------|----------------------|----------------------|----------------------|---------|
| NM       | 0.075667 5.6350 \times 10^{-3} | 3.1750 \times 10^{-5} | 1.0081 \times 10^{-9} | 2.0000              |
| OM       | 0.070030 3.0385 \times 10^{-5} | 8.5226 \times 10^{-19} | 5.2758 \times 10^{-73} | 4.0000              |
| NTRM     | 0.069979 2.1309 \times 10^{-5} | 2.7830 \times 10^{-23} | 1.0574 \times 10^{-112} | 5.0000              |
| GM       | 0.069961 3.9227 \times 10^{-5} | 1.1149 \times 10^{-21} | 2.0673 \times 10^{-104} | 5.0000              |
| NDM1 \((\alpha = 1, \beta = 1)\) | 0.070006 6.1200 \times 10^{-6} | 1.3737 \times 10^{-24} | 7.8257 \times 10^{-118} | 5.0000              |
| NDM2     | 0.069984 1.5860 \times 10^{-5} | 3.9187 \times 10^{-56} | 3.9040 \times 10^{-577} | 10.0000             |
Table 3. Comparisons of without-memory and with-memory methods after first three (n = 3) iterations for $\Theta_3(s)$.

| Method       | $|s_1 - s_0|$ | $|s_2 - s_1|$ | $|s_3 - s_2|$ | $|\Theta(s_3)|$ | COC  |
|--------------|--------------|--------------|--------------|----------------|------|
| NM           | 0.020781     | 4.1098 × 10^{-4} | 1.6149 × 10^{-7} | 6.8908 × 10^{-14} | 2.0000 |
| OM           | 0.020370     | 1.4498 × 10^{-7} | 3.5630 × 10^{-28} | 3.5916 × 10^{-110} | 4.0000 |
| NRTM         | 0.020369     | 1.8841 × 10^{-8} | 1.0645 × 10^{-38} | 1.6942 × 10^{-189} | 5.0000 |
| GM           | 0.020369     | 3.7534 × 10^{-8} | 6.9025 × 10^{-37} | 4.0127 × 10^{-180} | 5.0000 |
| NDM1 (\(\alpha = 1, \beta = 1\)) | 0.020369  | 1.0344 × 10^{-10} | 1.1052 × 10^{-52} | 4.2540 × 10^{-262} | 5.0000 |
| NDM2         | 0.020369     | 2.7829 × 10^{-9} | 2.4637 × 10^{-64} | 2.0132 × 10^{-834} | 10.0000 |

Table 4. Comparisons of without-memory and with-memory methods after first three (n = 3) iterations for $\Theta_4(s)$.

| Method       | $|s_1 - s_0|$ | $|s_2 - s_1|$ | $|s_3 - s_2|$ | $|\Theta(s_3)|$ | COC  |
|--------------|--------------|--------------|--------------|----------------|------|
| NM           | 0.0010012    | 1.168 × 10^{-6} | 1.5927 × 10^{-12} | 8.8779 × 10^{-24} | 2.0000 |
| OM           | 0.0010000    | 1.0065 × 10^{-12} | 1.0311 × 10^{-48} | 3.4070 × 10^{-192} | 4.0000 |
| NRTM         | 0.0010000    | 9.7700 × 10^{-15} | 8.6879 × 10^{-70} | 1.4493 × 10^{-344} | 5.0000 |
| GM           | 0.0010000    | 1.4655 × 10^{-14} | 1.0014 × 10^{-68} | 4.4761 × 10^{-339} | 5.0000 |
| NDM1 (\(\alpha = 1, \beta = 1\)) | 0.0010000 | 1.2108 × 10^{-16} | 3.2123 × 10^{-81} | 1.2667 × 10^{-403} | 5.0000 |
| NDM2         | 0.0010000    | 1.5040 × 10^{-15} | 9.7884 × 10^{-147} | 4.0028 × 10^{-1458} | 10.0000 |

Table 5. Comparisons of without-memory and with-memory methods after first three (n = 3) iterations for $\Theta_5(s)$.

| Method       | $|s_1 - s_0|$ | $|s_2 - s_1|$ | $|s_3 - s_2|$ | $|\Theta(s_3)|$ | COC  |
|--------------|--------------|--------------|--------------|----------------|------|
| NM           | 0.41667      | 8.0128 × 10^{-2} | 3.2000 × 10^{-3} | 1.0240 × 10^{-4} | 1.9990 |
| OM           | 0.33654      | 3.2051 × 10^{-3} | 1.3107 × 10^{-11} | 7.3787 × 10^{-45} | 4.0000 |
| NRTM         | 0.31962      | 1.3711 × 10^{-2} | 2.2606 × 10^{-10} | 5.1662 × 10^{-49} | 5.0000 |
| GM           | 0.30867      | 2.4662 × 10^{-2} | 7.7382 × 10^{-9} | 4.1619 × 10^{-41} | 5.0000 |
| NDM1 (\(\alpha = 0.1, \beta = 0.01\)) | 0.33269   | 1.3872 × 10^{-3} | 7.1489 × 10^{-16} | 1.3430 × 10^{-78} | 5.1293 |
| NDM2         | 0.33216      | 1.8261 × 10^{-3} | 4.8530 × 10^{-29} | 1.6984 × 10^{-284} | 10.0000 |

Table 6. Comparisons of without-memory and with-memory methods after first three (n = 3) iterations for $\Theta_6(s)$.

| Method       | $|s_1 - s_0|$ | $|s_2 - s_1|$ | $|s_3 - s_2|$ | $|\Theta(s_3)|$ | COC  |
|--------------|--------------|--------------|--------------|----------------|------|
| NM           | 1.1339       | 5.4470 × 10^{-1} | 1.9126 × 10^{-1} | 3.6580 × 10^{-2} | 2.0935 |
| OM           | 1.6641       | 1.8131 × 10^{-1} | 2.6397 × 10^{-4} | 1.6190 × 10^{-15} | 4.0000 |
| NRTM         | 1.6753       | 1.7655 × 10^{-1} | 2.2683 × 10^{-4} | 8.0963 × 10^{-19} | 5.0000 |
| GM           | 1.6385       | 2.1410 × 10^{-1} | 1.0058 × 10^{-3} | 2.3909 × 10^{-15} | 5.0003 |
| NDM1 (\(\alpha = 0.1, \beta = 0.01\)) | 1.7591   | 1.3677 × 10^{-1} | 1.9137 × 10^{-5} | 1.0706 × 10^{-24} | 5.1009 |
| NDM2         | 1.7514       | 1.3873 × 10^{-1} | 1.8773 × 10^{-9} | 3.2210 × 10^{-88} | 10.0000 |

From all the numerical results in Tables 1–8 and from Figure 1, it is concluded that the proposed family of methods NDM1 and NDM2 is highly competitive and possesses fast convergence toward the roots with minimal absolute residual error and a minimum error value in consecutive iteration as compared to the other existing methods. Additionally, the numerical results indicate that the computational order of convergence supports the theoretical convergence order of the newly presented family of methods in the test functions.
Methods for solving nonlinear equations. The first is a memoryless, derivative-free family of fifth-order methods, while the second is a three-point family of with-memory methods. We obtained the fifth-order family of methods by employing a composition technique along with a modified Newton’s method and a weighted function approach. The memory-based extension of the fifth-order family of methods employs two acceleration parameters calculated using Newton interpolating polynomials and enhancing convergence from the 5th to 10th order without requiring additional function evaluations and subsequently increasing its efficiency index from 1.495 to 1.778. Analysis of the numerical results has revealed the effectiveness and enhanced capabilities of the newly proposed methods in terms of minimal absolute residual error and minimal error in consecutive iterations. The results demonstrate that the proposed methods NDM1 and NDM2 exhibit faster convergence with smaller asymptotic constant values compared to other existing methods. Moreover, the overall performance of the

4. Conclusions

In this paper, we have presented two families of iterative methods for solving nonlinear equations. The first is a memoryless, derivative-free family of fifth-order methods, while the second is a three-point family of with-memory methods. We obtained the fifth-order family of methods by employing a composition technique along with a modified Newton’s method and a weighted function approach. The memory-based extension of the fifth-order family of methods employs two acceleration parameters calculated using Newton interpolating polynomials and enhancing convergence from the 5th to 10th order without requiring additional function evaluations and subsequently increasing its efficiency index from 1.495 to 1.778. Analysis of the numerical results has revealed the effectiveness and enhanced capabilities of the newly proposed methods in terms of minimal absolute residual error and minimal error in consecutive iterations. The results demonstrate that the proposed methods NDM1 and NDM2 exhibit faster convergence with smaller asymptotic constant values compared to other existing methods. Moreover, the overall performance of the

| Method         | \(|s_1 - s_0|\) | \(|s_2 - s_1|\) | \(|s_3 - s_2|\) | COC     |
|----------------|----------------|----------------|----------------|---------|
| NM             | 0.00011423     | 2.3586 × 10^{-10} | 1.0054 × 10^{-21} | 3.5263 × 10^{-45} | 2.0000 |
| OM             | 0.00011423     | 2.8491 × 10^{-16} | 7.5636 × 10^{-67} | 7.2512 × 10^{-270} | 4.0000 |
| NRTM           | 0.00011423     | 2.8491 × 10^{-16} | 1.1839 × 10^{-82} | 2.8305 × 10^{-415} | 5.0000 |
| GM             | 0.00011423     | 2.8491 × 10^{-16} | 2.5115 × 10^{-82} | 2.5802 × 10^{-413} | 5.0000 |
| NDM1 (\(\alpha = 1, \beta = 1\)) | 0.00011423 | 1.2212 × 10^{-19} | 1.7056 × 10^{-94} | 1.7490 × 10^{-469} | 5.0000 |
| NDM2           | 0.00011423     | 1.0938 × 10^{-23} | 1.4380 × 10^{-233} | 2.4917 × 10^{-2341} | 10.0000 |

| Method         | \(|s_1 - s_0|\) | \(|s_2 - s_1|\) | \(|s_3 - s_2|\) | COC     |
|----------------|----------------|----------------|----------------|---------|
| NM             | 0.00093269     | 2.4434 × 10^{-9} | 1.6769 × 10^{-18} | 6.6965 × 10^{-37} | 2.0000 |
| OM             | 0.00093272     | 7.7299 × 10^{-18} | 4.9129 × 10^{-71} | 6.7973 × 10^{-284} | 4.0000 |
| NRTM           | 0.00093272     | 7.7299 × 10^{-18} | 5.7906 × 10^{-88} | 1.1583 × 10^{-438} | 5.0000 |
| GM             | 0.00093272     | 7.7299 × 10^{-18} | 1.3540 × 10^{-87} | 1.8930 × 10^{-436} | 5.0000 |
| NDM1 (\(\alpha = 1, \beta = 1\)) | 0.00093272 | 4.0787 × 10^{-19} | 6.5223 × 10^{-91} | 5.7827 × 10^{-450} | 5.0000 |
| NDM2           | 0.00093272     | 2.7995 × 10^{-22} | 5.0470 × 10^{-217} | 6.7502 × 10^{-2169} | 10.0000 |
newly proposed work is quite impressive, offering a fast convergence speed, and making it a promising alternative for solving nonlinear equations.


**Funding:** This research was funded by the Technical University of Cluj-Napoca open access publication grant.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

**Abbreviations**
The following abbreviations are used in this manuscript:

- GM: Grau fifth-order method
- NM: Newton–Raphson method
- NDM1: Newly developed method 1 (without memory, Equation (3))
- NDM2: Newly developed method 2 (with memory, Equation (19))
- NRTM: Nouri fifth-order method (Equation (36))
- OM: Ostrowski’s method
- COC: Computational rate of convergence

The following constants were used in this manuscript:

- $c$: speed of light
- $P$: Planck constant

For the authors, the following abbreviations were used:

- E.S.: Ekta Sharma
- L.J.: Lorentz Jäntschi
- S.K.M.: Shubham Kumar Mittal
- S.P.: Sunil Panday
- D.-M.J.: Dan-Marian Joita
- L.L.P.: Lavinia Lorena Pruteanu

**References**


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.