mathematics

Article

A New Adaptive Eleventh-Order Memory Algorithm for Solving Nonlinear Equations

Sunil Panday [1], Shubham Kumar Mittal [1,*], Carmen Elena Stoenoiu [2] and Lorentz Jäntschi [3]

1 Department of Mathematics, National Institute of Technology Manipur, Langol, Imphal 795004, Manipur, India; sunilpanday@hotmail.co.in
2 Department of Electric Machines and Drives, Technical University of Cluj-Napoca, 26-28 Baritiu Str., 400027 Cluj-Napoca, Romania; carmen.stoenoiu@emd.utcluj.ro
3 Department of Physics and Chemistry, Technical University of Cluj-Napoca, 103-105 Muncii Blvd., 400641 Cluj-Napoca, Romania; lorentz.jantschi@chem.utcluj.ro
* Correspondence: shubhammehara2@gmail.com

Abstract: In this article, we introduce a novel three-step iterative algorithm with memory for finding the roots of nonlinear equations. The convergence order of an established eighth-order iterative method is elevated by transforming it into a with-memory variant. The improvement in the convergence order is achieved by introducing two self-accelerating parameters, calculated using the Hermite interpolating polynomial. As a result, the R-order of convergence for the proposed bi-parametric with-memory iterative algorithm is enhanced from 8 to 10.5208. Notably, this enhancement in the convergence order is accomplished without the need for extra function evaluations. Moreover, the efficiency index of the newly proposed with-memory iterative algorithm improves from 1.5157 to 1.6011. Extensive numerical testing across various problems confirms the usefulness and superior performance of the presented algorithm relative to some well-known existing algorithms.

Keywords: nonlinear equation; iterative method; efficiency index; with-memory algorithms; R-order convergence; self-accelerating

MSC: 65D99; 65H05; 41A25

Citation: Panday, S.; Mittal, S.K.; Stoenoiu, C.E.; Jäntschi, L. A New Adaptive Eleventh-Order Memory Algorithm for Solving Nonlinear Equations. *Mathematics* **2024**, *12*, 1809. https://doi.org/10.3390/math12121809

Academic Editor: Dhananjay Gopal

Received: 6 May 2024
Revised: 3 June 2024
Accepted: 7 June 2024
Published: 11 June 2024

1. Introduction

Addressing nonlinear equations is a critical challenge in science and engineering, particularly in fields such as gas dynamics and elasticity, where problems are often reduced to solving single-variable nonlinear equations $\Omega(x) = 0$, where $\Omega : D \subset \mathbb{R} \to \mathbb{R}$ acts as a scalar function within an open interval D. Traditional analytical methods frequently prove inadequate for determining the roots of these complex nonlinear equations, making iterative numerical methods indispensable, with the ongoing advancement of computational technology.

The classical one-point Newton's method [1] for a nonlinear equation is defined by the iterative formula:

$$x_{n+1} = x_n - \frac{\Omega(x_n)}{\Omega'(x_n)}, n = 0, 1, 2, \ldots \tag{1}$$

where Ω is the function and Ω' is its derivative. The Newton–Raphson method, known for its quadratic convergence near the roots, requires the evaluation of both the function and its derivative in each iteration. Researchers consistently strive to enhance the convergence rate of iterative methods.

Multipoint methods for solving nonlinear equations offer significant advantages over one-point methods due to their computational efficiency and higher convergence order. Researchers have shown considerable interest in constructing optimal multipoint methods without memory, following Kung–Traub's conjecture [2], which uses $n + 1$ functional evaluations to reach the optimal 2^n convergence order.

Recent innovations have improved a variety of numerical methods, including the Adomian decomposition Newton–Raphson [1], bisection [3], Chebyshev–Halley [4], Chun–Neta [5], collocation [6], Galerkin [7], and Jarratt methods [8], as well as the Nash–Moser iteration [9], Thukral method [10], Osada method [11], Ostrowski method [12], Picard iteration [13], diverse quadrature formulas [14,15], super-Halley method [16], and Traub–Steffensen method [17].

With the increase in the convergence rate, the number of evaluations of the required function also increases, which can lead to a reduction in the efficiency index. The efficiency index of an iterative method quantifies its performance, and it is defined as [2,18]:

$$E = \rho^{1/\gamma} \tag{2}$$

where ρ is the iterative method convergence rate and γ is the number of function and derivative evaluations performed per iteration.

On the other hand, iterative methods that incorporate memory make use of information from both recent and past iterations to boost both the convergence order and the efficiency index. Recent advancements in the field have seen significant contributions in extending without-memory methods to with-memory methods using self-accelerating parameters. In 2022, Choubey et al. [19] transformed a fourth-order without-memory iterative method into a with-memory method using one self-accelerating parameter and achieved sixth-order convergence. In 2023, Sharma et al. [20] upgraded an eighth-order without-memory iterative method to a with-memory method using two self-accelerating parameters and attained tenth-order convergence. Also in 2023, Abdullah et al. [21] developed a with-memory method by enhancing a without-memory method with one parameter, which improved its convergence order from 6 to 7.2749. Additionally, in the same year, Thangkhenpau et al. [22] developed a derivative-free without-memory iterative method with eighth-order convergence and then expanded it to a with-memory method using four self-accelerating parameters, which resulted in an increase in the convergence order from 8 to 15.5156. In their pursuit of resolving nonlinear equations with multiple roots, Thangkhenpau et al. introduced a novel scheme offering both with- and without-memory-based variants [23]. In recent years, the development of with-memory iterative methods has garnered considerable interest among researchers. For a deeper understanding, one can refer to [23–34] and the references cited therein.

In this research paper, a novel bi-parametric three-step with-memory iterative algorithm is introduced, which elevates the R-order of convergence from 8 to 10.5208. The algorithm achieves an efficiency index of 1.6011. The paper is structured to enhance understanding and analysis. Section 2 details the development of this new bi-parametric three-point with-memory iterative algorithm by integrating self-accelerating parameters into the first and third steps of an existing eighth-order without-memory iterative algorithm, accompanied by a thorough convergence analysis. Section 3 presents an extensive evaluation through numerical tests, providing a rigorous comparison of the proposed method with other well-established algorithms. Finally, Section 4 summarizes this study, offering a detailed synthesis of the results and their implications.

2. Analysis of Convergence for with-Memory Algorithm

In this section, the two parameters $\alpha, \beta \in \mathbb{R}$ used in the first and third steps, respectively, of Algorithm 2.1 in [35], proposed by Butsakorn Kong-ied in 2021, are utilized to increase its order of convergence.

$$\begin{aligned} y_n &= x_n - \frac{\Omega(x_n)}{\Omega'(x_n) + \alpha(\Omega(x_n))^2}, \\ z_n &= y_n - \frac{(\Omega(x_n))^2\Omega(y_n)}{(\Omega(x_n))^2\Omega'(x_n) - 2\Omega(x_n)\Omega'(x_n)\Omega(y_n) + \Omega'(x_n)(\Omega(y_n))^2}, \end{aligned} \tag{3}$$

$$x_{n+1} = z_n - \frac{\Omega(z_n)}{\Omega'(z_n) + \beta\Omega(z_n)}.$$

Using Taylor-series approximation, the expressions for $\Omega(x_n)$ and $\Omega'(x_n)$ can be written as:

$$\Omega(x_n) = A\big(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + c_6e_n^6 + c_7e_n^7 + c_8e_n^8\big), \tag{4}$$

$$\Omega'(x_n) = A\big(1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + 6c_6e_n^5 + 7c_7e_n^6 + 8c_8e_n^7 + 9c_9e_n^8\big), \tag{5}$$

where $A = \Omega'(\xi)$, ξ is the zero of $\Omega(x)$, $e_n = x_n - \xi$, and $c_j = \frac{\Omega^{(j)}(\xi)}{j!\Omega'(\xi)}$ for $j = 2, 3, \ldots$.

After substituting the values of Equations (4) and (5) into the first step of (3), the expression for the error of y_n is given by:

$$e_{n,y} = c_2e_n^2 + (A\alpha - 2c_2^2 + 2c_3)e_n^3 + \big(4c_2^3 - c_2(A\alpha + 7c_3) + 3c_4\big)e_n^4 + O(e_n^5). \tag{6}$$

where $e_{n,y} = y_n - \xi$.

Furthermore, the expression for $\Omega(y_n)$ can be written as:

$$\Omega(y_n) = Ac_2e_n^2 + A(A\alpha - 2c_2^2 + 2c_3)e_n^3 + A(5c_2^3 - c_2(A\alpha + 7c_3) + 3c_4)e_n^4 + O(e_n^5), \tag{7}$$

After substituting the values of Equations (4)–(7) into the second step of (3), the expression for the error of z_n is given by:

$$\begin{aligned} e_{n,z} =& c_2(-2A\alpha + 2c_2^2 - c_3)e_n^4 \\ &+ \big(-10c_2^4 + 7c_2^2(A\alpha + 2c_3) - (2A\alpha + c_3)(A\alpha + 2c_3) - 2c_2c_4\big)e_n^5 + O(e_n^6). \end{aligned} \tag{8}$$

where $e_{n,z} = z_n - \xi$.

Also, the expression for $\Omega(z_n)$ and $\Omega'(z_n)$ can be given by:

$$\begin{aligned} \Omega(z_n) =& -Ac_2(2A\alpha - 2c_2^2 + c_3)e_n^4 \\ &+ A\big(-10c_2^4 + 7c_2^2(A\alpha + 2c_3) - (2A\alpha + c_3)(A\alpha + 2c_3) - 2c_2c_4\big)e_n^5 + O(e_n^6), \end{aligned} \tag{9}$$

$$\begin{aligned} \Omega'(z_n) =& A - 2\big(Ac_2^2(2A\alpha - 2c_2^2 + c_3)\big)e_n^4 + 2Ac_2\times \\ &\big(-10c_2^4 + 7c_2^2(A\alpha + 2c_3) - (2A\alpha + c_3)(A\alpha + 2c_3) - 2c_2c_4\big)e_n^5 + O(e_n^6), \end{aligned} \tag{10}$$

Finally, after substituting the values of Equations (8)–(10) into the third step of (3), the expression for the error of x_{n+1} is given by:

$$\begin{aligned} e_{n+1} =& c_2^2(\beta + c_2)(2A\alpha - 2c_2^2 + c_3)^2e_n^8 + 2c_2(\beta + c_2)(2A\alpha - 2c_2^2 + c_3)\times \\ &(10c_2^4 - 7c_2^2(A\alpha + 2c_3) + (2A\alpha + c_3)(A\alpha + 2c_3) + 2c_2c_4)e_n^9 \\ &+ (\beta + c_2)\big(224c_2^8 + (2A\alpha + c_3)^2(A\alpha + 2c_3)^2 - 10c_2^6(34A\alpha + 63c_3) \\ &+ 124c_2^5c_4 - 18c_2^3(8A\alpha + 7c_3)c_4 + 2c_2(2A\alpha + c_3)(10A\alpha + 11c_3)c_4 \\ &+ c_2^4(173A^2\alpha^2 + 730A\alpha c_3 + 500c_3^2 - 12c_5) - 2c_2^2(18A^3\alpha^3 + 119A^2\alpha^2c_3 \\ &+ 171A\alpha c_3^2 + 58c_3^3 - 2c_4^2 - 3(2A\alpha + c_3)c_5)\big)e_n^{10} - 2\big((\beta + c_2)(458c_2^9 \\ &- c_2^7(637A\alpha + 1720c_3) + 472c_2^6c_4 - (2A\alpha + c_3)(A\alpha + 2c_3)(8A\alpha + 7c_3)c_4 \\ &+ c_2^5(311A^2\alpha^2 + 4c_3(479A\alpha + 498c_3) - 86c_5) + c_2(6A^4\alpha^4 + 79A^3\alpha^3c_3 \\ &+ 258A^2\alpha^2c_3^2 + 270A\alpha c_3^3 + 80c_3^4 - 28A\alpha c_4^2 - 20c_3c_4^2 - 2(2A\alpha + c_3)(7A\alpha + 8c_3)c_5) \\ &+ c_2^3(-68A^3\alpha^3 - 1508A\alpha c_3^2 - 792c_3^3 + 54c_4^2 + 99A\alpha c_5 + c_3(-665A^2\alpha^2 + 90c_5)) \\ &+ c_2^4(-((541A\alpha + 784c_3)c_4) + 8c_6) + 2c_2^2(c_4(89A^2\alpha^2 + 3c_3(89A\alpha + 48c_3) - 3c_5) \end{aligned}$$

$$- 2(2A\alpha + c_3)c_6)))e_n^{11} + O(e_n^{12}). \tag{11}$$

Now, by replacing α and β, which are calculated using Equations (13) and (14), respectively, with α_n and β_n in Equation (3), the following with-memory iterative scheme is obtained:

$$\begin{aligned} y_n &= x_n - \frac{\Omega(x_n)}{\Omega'(x_n) + \alpha_n(\Omega(x_n))^2}, \\ z_n &= y_n - \frac{(\Omega(x_n))^2\Omega(y_n)}{(\Omega(x_n))^2\Omega'(x_n) - 2\Omega(x_n)\Omega'(x_n)\Omega(y_n) + \Omega'(x_n)(\Omega(y_n))^2}, \\ x_{n+1} &= z_n - \frac{\Omega(z_n)}{\Omega'(z_n) + \beta_n\Omega(z_n)}. \end{aligned} \tag{12}$$

The above scheme is denoted by NWM11. At this point, from (11), it is clear that the convergence order of Algorithm (3) is 8 when $\alpha \neq \frac{2c_2^2 - c_3}{2A}$ and $\beta \neq -c_2$, respectively. To accelerate the order of convergence from 8 to 11 of Algorithm (3), one can assume that $\alpha = \frac{2c_2^2 - c_3}{2A} = \frac{3(\Omega''(\xi))^2 - \Omega'''(\xi)\Omega'(\xi)}{12(\Omega'(\xi))^3}$ and $\beta = -c_2 = -\frac{\Omega''(\xi)}{2\Omega'(\xi)}$; however, the exact values of $\Omega'(\xi), \Omega''(\xi)$, and $\Omega'''(\xi)$ are not attainable in practice. Let us assume the parameters α and β as α_n and β_n, respectively. The parameters α_n and β_n can be updated iteratively using the available data from the current and previous iterations, aiming for them to satisfy the conditions $\lim_{n\to\infty} \alpha_n = \frac{2c_2^2 - c_3}{2A} = \frac{3(\Omega''(\xi))^2 - \Omega'''(\xi)\Omega'(\xi)}{12(\Omega'(\xi))^3}$ and $\lim_{n\to\infty} \beta_n = -c_2 = -\frac{\Omega''(\xi)}{2\Omega'(\xi)}$ such that the asymptotic convergence constants for the 8th, 9th, and 10th orders in the error expression (11) will be zero. α_n and β_n are chosen as follows:

$$\alpha_n = \frac{3(H_5''(x_n))^2 - H_6'''(y_n)\Omega'(x_n)}{12(\Omega'(x_n))^3}, \tag{13}$$

$$\beta_n = -\frac{H_5''(x_n)}{2\Omega'(x_n)}, \tag{14}$$

where

$$\begin{aligned} H_6(x) &= \Omega(y_n) + (x - y_n)\Omega[y_n, x_n] + (x - y_n)(x - x_n)\Omega[y_n, x_n, x_n] \\ &\quad + (x - y_n)(x - x_n)^2\Omega[y_n, x_n, x_n, z_{n-1}] + (x - y_n)(x - x_n)^2(x - z_{n-1}) \\ &\quad \Omega[y_n, x_n, x_n, z_{n-1}, y_{n-1}] + (x - y_n)(x - x_n)^2(x - z_{n-1})(x - y_{n-1}) \\ &\quad \Omega[y_n, x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}] + (x - y_n)(x - x_n)^2(x - z_{n-1})(x - y_{n-1}) \\ &\quad (x - x_{n-1})\Omega[y_n, x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}, x_{n-1}], \\ H_5(x) &= \Omega(x_n) + (x - x_n)\Omega[x_n, x_n] + (x - x_n)^2\Omega[x_n, x_n, z_{n-1}] \\ &\quad + (x - x_n)^2(x - z_{n-1})\Omega[x_n, x_n, z_{n-1}, y_{n-1}] + (x - x_n)^2(x - z_{n-1})(x - y_{n-1}) \\ &\quad \Omega[x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}] + (x - x_n)^2(x - z_{n-1})(x - y_{n-1})(x - x_{n-1}) \\ &\quad \Omega[x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}, x_{n-1}], \\ H_6'''(y_n) &= 6\Omega[y_n, x_n, x_n, z_{n-1}] + 6\big(2(y_n - x_n) + (y_n - z_{n-1})\big) \\ &\quad \Omega[y_n, x_n, x_n, z_{n-1}, y_{n-1}] + 6\big((y_n - x_n)^2 + 2(y_n - x_n)(y_n - z_{n-1}) \\ &\quad + 2(y_n - x_n)(y_n - y_{n-1}) + (y_n - z_{n-1})(y_n - y_{n-1})\big) \\ &\quad \Omega[y_n, x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}] + 6\big((y_n - x_n)^2(y_n - z_{n-1}) \\ &\quad + (y_n - x_n)^2(y_n - y_{n-1}) + 2(y_n - x_n)(y_n - z_{n-1})(y_n - y_{n-1}) \\ &\quad + (y_n - x_n)^2(y_n - x_{n-1}) + 2(y_n - x_n)(y_n - z_{n-1})(y_n - x_{n-1}) \\ &\quad + 2(y_n - x_n)(y_n - y_{n-1})(y_n - x_{n-1}) + (y_n - z_{n-1})(y_n - y_{n-1})(y_n - x_{n-1})\big) \\ &\quad \Omega[y_n, x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}, x_{n-1}], \\ H_5''(x_n) &= 2\Omega[x_n, x_n, z_{n-1}] + 2(x_n - z_{n-1})\Omega[x_n, x_n, z_{n-1}, y_{n-1}] \\ &\quad + 2\big((x_n - z_{n-1})(x_n - y_{n-1})\big)\Omega[x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}] \end{aligned} \tag{15}$$

$$+2\big((x_n - z_{n-1})(x_n - y_{n-1})(x_n - x_{n-1})\big)\Omega[x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}, x_{n-1}].$$

It should be noted that the condition $H'_m(x_n) = \Omega'(x_n)$ is satisfied by the Hermite interpolation polynomial $H_m(x)$ for $m = 5, 6$. So, $\alpha_n = \frac{3(H''_5(x_n))^2 - H'''_6(y_n)\Omega'(x_n)}{12(\Omega'(x_n))^3}$ and $\beta_n = -\frac{H''_5(x_n)}{2\Omega'(x_n)}$ can be expressed as $\alpha_n = \frac{3(H''_5(x_n))^2 - H'''_6(y_n)H'_m(x_n)}{12(H'_m(x_n))^3}$ and $\beta_n = -\frac{H''_5(x_n)}{2H'_m(x_n)}$, respectively, for $m = 5, 6$.

Theorem 1. *Let H_m be the Hermite polynomial of degree m, interpolating the Ω function at interpolation nodes $y_n, x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}, x_{n-1}$ within an interval $I \subset D \subset \mathbb{R}$, and the derivative $\Omega^{(m+1)}$ is continuous in I with $H_m(x_n) = \Omega(x_n)$ and $H'_m(x_n) = \Omega'(x_n)$. Suppose that all nodes $y_n, x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}, x_{n-1}$ are in the neighborhood of the root ξ. Then,*

$$H'''_6(y_n) = 6\Omega'(\xi)(c_3 - c_7 e_{n-1,z} e_{n-1,y} e_{n-1}^2), \tag{16}$$

$$H''_5(x_n) = 2\Omega'(\xi)(c_2 - c_6 e_{n-1,z} e_{n-1,y} e_{n-1}^2). \tag{17}$$

and

$$\alpha_n = \frac{3(H''_5(x_n))^2 - H'''_6(y_n)\Omega'(x_n)}{12(\Omega'(x_n))^3} \sim \frac{2c_2^2 - c_3}{2A} - \frac{c_6^2}{A} e_{n-1,z}^2 e_{n-1,y}^2 e_{n-1}^4 + \frac{c_7}{2A} e_{n-1,z} e_{n-1,y} e_{n-1}^2, \tag{18}$$

$$\beta_n = -\frac{H''_5(x_n)}{2\Omega'(x_n)} \sim -c_2 + c_6 e_{n-1,z} e_{n-1,y} e_{n-1}^2. \tag{19}$$

Again, after simplification, the result is

$$\alpha_n - \frac{2c_2^2 - c_3}{2A} = 2A\alpha - 2c_2^2 + c_3 \sim \Big(-\frac{c_6^2}{A} e_{n-1,z} e_{n-1,y} e_{n-1}^2 + \frac{c_7}{2A}\Big) e_{n-1,z} e_{n-1,y} e_{n-1}^2, \tag{20}$$

$$\beta_n + c_2 \sim c_6 e_{n-1,z} e_{n-1,y} e_{n-1}^2. \tag{21}$$

Proof. The sixth-degree and fifth-degree Hermite interpolation polynomials are

$$\Omega(x) - H_6(x) = \frac{\Omega^{(7)}(\delta)}{7!}(x - y_n)(x - x_n)^2(x - z_{n-1})(x - y_{n-1})(x - x_{n-1})^2, \tag{22}$$

$$\Omega(x) - H_5(x) = \frac{\Omega^{(6)}(\delta)}{6!}(x - x_n)^2(x - z_{n-1})(x - y_{n-1})(x - x_{n-1})^2. \tag{23}$$

In order to obtain the following equations, Equation (22) is differentiated three times in $x = y_n$, and Equation (23) is differentiated two times in $x = x_n$:

$$H'''_6(y_n) = \Omega'''(y_n) - 6\frac{\Omega^{(7)}(\delta)}{7!}(y_n - z_{n-1})(y_n - y_{n-1})(y_n - x_{n-1})^2, \tag{24}$$

$$H''_5(x_n) = \Omega''(x_n) - \frac{\Omega^{(6)}(\delta)}{6!}(x_n - z_{n-1})(x_n - y_{n-1})(x_n - x_{n-1})^2. \tag{25}$$

The Taylor-series expansion of Ω' at points y_n and x_n in I and $\delta \in I$ about the simple zero ξ of Ω provides

$$\Omega'(x_n) = \Omega'(\xi)\Big(1 + 2c_2 e_n + 3c_3 e_n^2 + O(e_n^3)\Big), \tag{26}$$

$$\Omega''(x_n) = \Omega'(\xi)\Big(2c_2 + 6c_3 e_n + O(e_n^2)\Big), \tag{27}$$

Similarly,

$$\Omega'''(y_n) = \Omega'(\xi)\Big(6c_3 + 24c_4 e_{n,y} + O(e_{n,y}^2)\Big), \tag{28}$$

$$\Omega^{(6)}(\delta) = \Omega'(\xi)\Big(6!c_6 + 7!c_7 e_\delta + O(e_\delta^2)\Big), \tag{29}$$

$$\Omega^{(7)}(\delta) = \Omega'(\xi)\Big(7!c_7 + 8!c_8e_\delta + O(e_\delta^2)\Big). \tag{30}$$

where $e_\delta = \delta - \xi$. Putting (28) and (30) into (24), and (27) and (29) into (25), we obtain

$$H_6'''(y_n) = 6\Omega'(\xi)(c_3 - c_7e_{n-1,z}e_{n-1,y}e_{n-1}^2), \tag{31}$$

and

$$H_5''(x_n) = 2\Omega'(\xi)(c_2 - c_6e_{n-1,z}e_{n-1,y}e_{n-1}^2). \tag{32}$$

By using Equations (26), (31) and (32), the result is

$$\frac{3(H_5''(x_n))^2 - H_6'''(y_n)\Omega'(x_n)}{12(\Omega'(x_n))^3} \sim \frac{2c_2^2 - c_3}{2A} - \frac{c_6^2}{A}e_{n-1,z}^2e_{n-1,y}^2e_{n-1}^4 + \frac{c_7}{2A}e_{n-1,z}e_{n-1,y}e_{n-1}^2, \tag{33}$$

and

$$-\frac{H_5''(x_n)}{2\Omega'(x_n)} \sim -c_2 + c_6e_{n-1,z}e_{n-1,y}e_{n-1}^2. \tag{34}$$

Hence,

$$\alpha_n \sim \frac{2c_2^2 - c_3}{2A} + \Big(-\frac{c_6^2}{A}e_{n-1,z}e_{n-1,y}e_{n-1}^2 + \frac{c_7}{2A}\Big)e_{n-1,z}e_{n-1,y}e_{n-1}^2, \tag{35}$$

$$\beta_n \sim -c_2 + c_6e_{n-1,z}e_{n-1,y}e_{n-1}^2. \tag{36}$$

or

$$\alpha_n - \frac{2c_2^2 - c_3}{2A} = 2A\alpha - 2c_2^2 + c_3 \sim \Big(-\frac{c_6^2}{A}e_{n-1,z}e_{n-1,y}e_{n-1}^2 + \frac{c_7}{2A}\Big)e_{n-1,z}e_{n-1,y}e_{n-1}^2, \tag{37}$$

$$\beta_n + c_2 \sim c_6e_{n-1,z}e_{n-1,y}e_{n-1}^2. \tag{38}$$

This completes the proof of Theorem 1. □

R-Order of Convergence: It can be said that sequence $\{x_n\}$ converges to x^* with an R-order of convergence of at least $\tau > 1$ if there are constants $C \in (0, \infty)$ and $\theta \in (0, 1)$ such that [36]

$$||x^* - x_n|| \leq C.\theta^{\tau^n}; n = 0, 1, \ldots \tag{39}$$

Using the above definition of R-order of convergence, along with the statement in [37], provides an estimate of the order of convergence of the iterative scheme (12).

Theorem 2. *If the errors $e_j = x_j - \xi$ are evaluated through the iterative root-finding method, the following relation exists*

$$e_{k+1} \sim \prod_{i=0}^{m-2}(e_{k-i})^{m_i}, k \geq k(\{e_k\}) \tag{40}$$

and then the R-order of convergence of the IM, denoted as $O_R(IM, \xi)$, satisfies the inequality $O_R(IM, \xi) \geq s^*$, where s^* is the unique positive solution of the equation $s_{n+1} - \sum_{i=0}^{n} m_i s^{n-i} = 0$ [37].

Going further, the new iterative scheme with memory (12) is regulated by the subsequent convergence theorem.

Theorem 3. *In the iterative method (12), let α_n β_n be the varying parameters that are calculated using Equations (13) and (14). If an initial guess x_0 is near a simple zero ξ of $\Omega(x)$, then the R-order of convergence of the iterative method (12) with memory is at least 10.5208.*

Proof. Let the iterative method (IM) generate the sequence of $\{x_n\}$, which converges to the root ξ of $\Omega(x)$. By means of R-order $O_R(IM,\xi) \geq r$, it is expressed as

$$e_{n+1} \sim D_{n,r} e_n^r, \tag{41}$$

and

$$e_n \sim D_{n-1,r} e_{n-1}^r. \tag{42}$$

Next, D_r of IM tends to $D_{n,r}$. The result will be an asymptotic error constant when $n \to \infty$, and then

$$e_{n+1} \sim D_{n,r}(D_{n-1,r}e_{n-1}^r)^r = D_{n,r}D_{n-1,r}^r e_{n-1}^{r^2}. \tag{43}$$

The resulting error expression of the with-memory scheme (12) can be obtained using Equations (6), (8) and (11) and the varying parameters α_n and β_n.

$$e_{n,y} = y_n - \xi \sim c_2 e_n^2, \tag{44}$$

$$e_{n,z} = z_n - \xi \sim -c_2(2A\alpha - 2c_2^2 + c_3)e_n^4, \tag{45}$$

and

$$e_{n+1} = x_{n+1} - \xi \sim c_2^2(\beta + c_2)(2A\alpha - 2c_2^2 + c_3)^2 e_n^8. \tag{46}$$

It should be noted that in Equations (44)–(46), the higher-order terms are excluded.

Furthermore, if the R-order convergence of the iterative sequences $\{y_n\}$ and $\{z_n\}$ are p and q, respectively, then

$$e_{n,y} \sim D_{n,p}e_n^p \sim D_{n,p}(D_{n-1,r}e_{n-1}^r)^p = D_{n,p}D_{n-1,r}^p e_{n-1}^{rp}, \tag{47}$$

and

$$e_{n,z} \sim D_{n,q}e_n^q \sim D_{n,q}(D_{n-1,r}e_{n-1}^r)^q = D_{n,q}D_{n-1,r}^q e_{n-1}^{rq}. \tag{48}$$

Now, from Equations (42) and (44), the obtained result is

$$e_{n,y} \sim c_2 e_n^2 \sim c_2(D_{n-1,r}e_{n-1}^r)^2 \sim c_2 D_{n-1,r}^2 e_{n-1}^{2r}. \tag{49}$$

Also, from Equations (37), (42) and (45), the following is obtained

$$\begin{aligned} e_{n,z} &\sim -c_2(2A\alpha - 2c_2^2 - c_3)e_n^4 \\ &\sim -c_2 B_n e_{n-1,z} e_{n-1,y} e_{n-1}^2 (D_{n-1,r}e_{n-1}^r)^4 \\ &\sim -c_2 B_n (D_{n-1,q}e_{n-1}^q)(D_{n-1,p}e_{n-1}^p)e_{n-1}^2 (D_{n-1,r}e_{n-1}^r)^4 \\ &\sim -c_2 B_n D_{n-1,q} D_{n-1,p} D_{n-1,r}^4 e_{n-1}^{4r+p+q+2}. \end{aligned} \tag{50}$$

Again, from Equations (37), (38), (42) and (46), the result is

$$\begin{aligned} e_{n+1} &\sim c_2^2(\beta + c_2)(2A\alpha - 2c_2^2 - c_3)^2 e_n^8 \\ &\sim c_2^2 (c_6 e_{n-1,z} e_{n-1,y} e_{n-1}^2)(B_n e_{n-1,z} e_{n-1,y} e_{n-1}^2)^2 (D_{n-1,r}e_{n-1}^r)^8 \\ &\sim c_2^2 c_6 B_n^2 e_{n-1,z}^3 e_{n-1,y}^3 e_{n-1}^6 (D_{n-1,r}e_{n-1}^r)^8 \\ &\sim c_2^2 c_6 B_n^2 (D_{n-1,q}e_{n-1}^q)^3 (D_{n-1,p}e_{n-1}^p)^3 e_{n-1}^6 (D_{n-1,r}e_{n-1}^r)^8 \\ &\sim c_2^2 c_6 B_n^2 D_{n-1,q}^3 D_{n-1,p}^3 D_{n-1,r}^8 e_{n-1}^{8r+3p+3q+6}. \end{aligned} \tag{51}$$

where $B_n = \left(-\frac{c_6^2}{A} e_{n-1,z} e_{n-1,y} e_{n-1}^2 + \frac{c_7}{2A}\right)$.

Since $r > q > p$, equating the exponents of e_{n-1} from the set of relations (47)–(49), (48)–(50) and (43)–(51), the following system of equations is obtained:

$$\begin{aligned} rp &= 2r, \\ rq &= p + q + 4r + 2, \\ r^2 &= 3p + 3q + 8r + 6. \end{aligned} \tag{52}$$

The solution of (52) is $r = 10.5208$, $q = 4.8403$, and $p = 2$. As a result, the R-order of convergence of the with-memory iterative method (12) is at least 10.5208. □

3. Numerical Discussion

In this section, the convergence behavior of the newly developed with-memory method (NWM11) presented in (12) is explored. The goal of this section is to assess the effectiveness of a recently developed iterative method by applying it to a range of nonlinear equations. The nonlinear test functions, along with their roots and initial guesses for numerical analysis, are detailed below:

Example 1. $\Omega_1(x) = e^x \sin x + \log x^4 - 3x + 1$, $x_0 = 0.3$, $\xi \approx 0.9400$.

Example 2. $\Omega_2(x) = x^2 - (1-x)^{25}$, $x_0 = 0.2$, $\xi \approx 0.1437$.

Example 3. $\Omega_3(x) = e^{x^3 - x} - \cos(x^2 - 1) + x^3 + 1$, $x_0 = -1.6$, $\xi \approx -1.0000$.

Example 4. $\Omega_4(x) = x^6 - 10x^3 + x^2 - x + 3$, $x_0 = 0.6$, $\xi \approx 0.6586$.

Example 5. $\Omega_5(x) = (\sin x)^2 - x^2 + 1$, $x_0 = 1.3$, $\xi \approx 1.4045$.

Example 6. $\Omega_6(x) = x^3 - 10$, $x_0 = 2.1$, $\xi \approx 2.1544$.

Example 7. $\Omega_7(x) = x^5 + x^4 + 4x^2 - 15$, $x_0 = 1.3$, $\xi \approx 1.3474$.

Example 8. $\Omega_8(x) = e^{x^2 + x\cos x - 1} \sin(\pi x) + x \log(x \sin x + 1)$, $x_0 = 0.53$, $\xi \approx 0.5313$.

The proposed method is evaluated against several well-established methods documented in the literature, including BK8 (53), KP10 (54), OSO10 (55), NJ10 (56), NAJJ10 (57), XT10 (58), and NWM11 (12), which are described below.

In 2021, Butsokorn Kong-ied (BK8) [35] developed an eighth-order iterative method, which is defined as:

$$\begin{aligned} y_n &= x_n - \frac{\Omega(x_n)}{\Omega'(x_n)}, \\ z_n &= y_n - \frac{(\Omega(x_n))^2 \Omega(y_n)}{\Omega(x_n))^2 \Omega'(x_n) - 2\Omega(x_n)\Omega'(x_n)\Omega(y_n) + \Omega'(x_n)(\Omega(y_n))^2}, \\ x_{n+1} &= z_n - \frac{\Omega(z_n)}{\Omega'(z_n)}. \end{aligned} \tag{53}$$

In 2024, Devi and Maroju (KP10) [38] developed a tenth-order iterative method, defined as:

$$\begin{aligned} y_n &= x_n - \frac{\Omega(x_n)}{\Omega'(x_n)}, \\ z_n &= y_n - 5\frac{\Omega(y_n)}{\Omega'(x_n)}, \end{aligned} \tag{54}$$

$$w_n = z_n - \frac{\Omega(z_n) - 16\Omega(y_n)}{5\Omega'(x_n)},$$
$$x_{n+1} = w_n - \frac{\Omega(w_n)}{\Omega'(w_n)}.$$

In 2023, Ogbereyivwe et al. (OSO10) [39] developed a tenth-order iterative method, defined as:

$$\begin{aligned} w_n &= x_n + \alpha(\Omega(x_n))^3, \\ y_n &= x_n - \frac{\Omega(x_n)}{\Omega[x_n, w_n]}, \\ z_n &= y_n - \frac{\Omega(y_n)}{\Omega'(y_n)} + \frac{1}{2}\left(\frac{\Omega(y_n)}{\Omega'(y_n)}\right)^2 \left(\frac{\Omega'(y_n) - \Omega[x_n, w_n]}{\Omega'(y_n)}\right)\frac{\Omega[x_n, w_n]}{\Omega(x_n)}, \\ x_{n+1} &= z_n - G(u_n)\frac{\Omega(z_n)}{2\Omega[y_n, z_n] - \Omega'(y_n)}, \end{aligned} \tag{55}$$

where $u_n = \frac{\Omega(z_n)}{\Omega(x_n)}$ and $\alpha \in R$.

In 2016, Choubey and Jaiswal (NJ10) [32] developed a bi-parametric with-memory iterative method, with tenth-order convergence for solving nonlinear equations, defined as:

$$\begin{aligned} y_n &= x_n - \frac{\Omega(x_n)}{\Omega'(x_n) - T_n\Omega(x_n)}, \\ z_n &= y_n - \left(\frac{\Omega(y_n)(\Omega(x_n) + \gamma\Omega(y_n))}{(\Omega'(x_n) - 2T_n\Omega(x_n))(\Omega(x_n) + (\gamma - 2)\Omega(y_n))}\right), \\ x_{n+1} &= z_n - \frac{\Omega(z_n)}{\Omega[z_n, y_n] + \Omega[z_n, y_n, x_n](z_n - y_n) + \Omega[z_n, y_n, x_n, x_n](z_n - y_n)(z_n - x_n)}, \end{aligned} \tag{56}$$

where $T, \gamma \in$ R and T_n is calculated as $T_n = \frac{H_5''(x_n)}{2\Omega'(x_n)}$.

In 2018, Choubey et al. (NAJJ10) [33] proposed a tenth-order with-memory iterative method using two self-accelerating parameters, defined as:

$$\begin{aligned} y_n &= x_n - \frac{\Omega(x_n)}{\Omega'(x_n) - \gamma_n\Omega(x_n)}, \\ z_n &= y_n - \left(\frac{\Omega(y_n)}{-\Omega'(x_n) + 2((\Omega(y_n) - \Omega(x_n))/(y_n - x_n))}\right), \\ x_{n+1} &= z_n - \Omega(z_n)\Big(-\frac{\Omega(y_n) - \Omega(x_n)}{y_n - x_n} + \frac{\Omega(z_n) - \Omega(y_n)}{z_n - y_n} + \frac{\Omega(z_n) - \Omega(x_n)}{z_n - x_n} \\ &\quad + \lambda_n(z_n - x_n)(z_n - y_n)\Big), \end{aligned} \tag{57}$$

where γ and $\lambda \in R$ are calculated as $\gamma_n = \frac{H_5''(x_n)}{2\Omega'(x_n)}$ and $\lambda_n = \frac{H_6'''(y_n)}{6}$.

In 2013, Wang and Zhang (XT10) [34] developed a family of three-step with-memory iterative schemes for nonlinear equations, defined as:

$$\begin{aligned} y_n &= x_n - \frac{\Omega(x_n)}{\Omega'(x_n) - T_n\Omega(x_n)}, \\ z_n &= y_n - \left(\frac{\Omega(y_n)}{2\Omega[x_n, y_n] - \Omega'(x_n) + T_n\Omega(y_n)}\right), \\ x_{n+1} &= z_n - [G(s_n) + H(t_n)]\left(\frac{(\alpha + w)\Omega(z_n)}{2w\Omega[y_n, z_n] + (\alpha - w)(\Omega'(x_n) + L\Omega(z_n))}\right), \end{aligned} \tag{58}$$

where $s_n = \frac{\Omega(z_n)}{\Omega(x_n)}$, $t_n = \frac{\Omega(y_n)}{\Omega(x_n)}$, $\alpha = y_n - x_n$, $w = z_n - x_n$, and $L \in R$. Also, T_n is calculated as $T_n = -\frac{H_5''(x_n)}{2\Omega'(x_n)}$.

All the comparative results for these methods are summarized in Tables 1–8. These tables present the absolute differences between the last two consecutive iterations ($|x_n - x_{n-1}|$) and the absolute residual error ($|\Omega(x_n)|$) of up to three iterations for each function, along with the computational order of convergence (COC) for the proposed method in comparison to some well-known existing methods. The determination of the COC is achieved using the following equation [40]:

$$COC = \frac{log|\Omega(x_n)/\Omega(x_{n-1})|}{log|\Omega(x_{n-1})/\Omega(x_{n-2})|}. \quad (59)$$

For all numerical calculations, the programming software Mathematica 12.2 was used. For the newly proposed with-memory algorithm (NWM11), the parameter values $\alpha_0 = 0.001$ and $\beta_0 = -0.1$ were selected to start the initial iteration.

Table 1. Comparison of without-memory and with-memory algorithms after the first three (n = 3) iterations for $\Omega_1(x)$.

Method	$|(x_1 - x_0)|$	$|(x_2 - x_1)|$	$|(x_3 - x_2)|$	$|\Omega(x_3)|$	**COC**
BK8	0.64003	1.6713×10^{-6}	1.3856×10^{-50}	1.4946×10^{-402}	8.0000
KP10	0.63843	1.6023×10^{-3}	2.4259×10^{-32}	7.2153×10^{-320}	10.0000
OSO10	0.89217	3.7908×10^{-2}	2.5397×10^{-14}	3.3269×10^{-134}	9.9375
NJ10	0.63416	5.8788×10^{-3}	2.3098×10^{-24}	3.3435×10^{-238}	10.0221
NAJJ10	0.60909	3.0943×10^{-2}	1.9071×10^{-18}	1.4560×10^{-173}	9.6100
XT10	0.59184	4.8191×10^{-2}	2.7250×10^{-15}	3.8179×10^{-147}	10.0023
NWM11	0.64003	2.6164×10^{-6}	1.6115×10^{-57}	5.2489×10^{-609}	10.7820

Table 2. Comparison of without-memory and with-memory algorithms after the first three (n = 3) iterations for $\Omega_2(x)$.

Method	$|(x_1 - x_0)|$	$|(x_2 - x_1)|$	$|(x_3 - x_2)|$	$|\Omega(x_3)|$	**COC**
BK8	0.05626	7.2495×10^{-7}	1.3567×10^{-43}	1.8178×10^{-337}	8.0000
KP10	0.05674	4.2508×10^{-4}	4.1551×10^{-23}	9.7770×10^{-214}	9.9988
OSO10	0.05627	1.0119×10^{-5}	1.4540×10^{-41}	4.8595×10^{-400}	10.0000
NJ10	0.05631	4.8255×10^{-5}	9.9239×10^{-39}	7.0418×10^{-373}	9.9177
NAJJ10	0.05620	6.0195×10^{-5}	3.6586×10^{-36}	2.1280×10^{-347}	9.9688
XT10	0.05636	9.6954×10^{-5}	3.1227×10^{-33}	4.5026×10^{-318}	9.9953
NWM11	0.05626	7.3371×10^{-7}	2.9281×10^{-56}	1.0098×10^{-576}	10.5350

Table 3. Comparison of without-memory and with-memory algorithms after the first three (n = 3) iterations for $\Omega_3(x)$.

Method	$|(x_1 - x_0)|$	$|(x_2 - x_1)|$	$|(x_3 - x_2)|$	$|\Omega(x_3)|$	**COC**
BK8	0.60000	6.1229×10^{-11}	1.9861×10^{-83}	1.2169×10^{-662}	8.0000
KP10	0.60000	1.9866×10^{-12}	2.1861×10^{-119}	2.8467×10^{-1188}	10.0000
OSO10	*divergent*	*divergent*	*divergent*	*divergent*	*divergent*
NJ10	0.60000	2.5816×10^{-9}	1.1089×10^{-87}	3.5748×10^{-869}	9.9811
NAJJ10	0.60000	2.6174×10^{-9}	9.1514×10^{-87}	4.9636×10^{-860}	9.9923
XT10	0.60000	9.3477×10^{-9}	1.8199×10^{-81}	4.2835×10^{-807}	9.9893
NWM11	0.60000	1.4300×10^{-13}	8.1050×10^{-139}	2.9949×10^{-1451}	10.4880

Table 4. Comparison of without-memory and with-memory algorithms after the first three (n = 3) iterations for $\Omega_4(x)$.

Method	$\lvert(x_1 - x_0)\rvert$	$\lvert(x_2 - x_1)\rvert$	$\lvert(x_3 - x_2)\rvert$	$\lvert\Omega(x_3)\rvert$	COC
BK8	0.05860	6.1011×10^{-9}	4.6502×10^{-65}	6.3312×10^{-513}	8.0000
KP10	0.05860	5.6994×10^{-9}	1.0558×10^{-79}	6.0064×10^{-786}	10.0000
OSO10	0.05861	1.0003×10^{-5}	1.2746×10^{-49}	1.7573×10^{-465}	9.9998
NJ10	0.05860	1.6837×10^{-8}	1.3324×10^{-78}	1.7409×10^{-778}	9.9992
NAJJ10	0.05860	4.0950×10^{-9}	6.2510×10^{-86}	6.2244×10^{-855}	10.0252
XT10	0.05860	1.0741×10^{-8}	2.5825×10^{-80}	2.1179×10^{-795}	9.9996
NWM11	0.05860	5.6515×10^{-9}	1.5026×10^{-88}	2.5987×10^{-922}	10.4910

Table 5. Comparison of without-memory and with-memory algorithms after the first three (n = 3) iterations for $\Omega_5(x)$.

Method	$\lvert(x_1 - x_0)\rvert$	$\lvert(x_2 - x_1)\rvert$	$\lvert(x_3 - x_2)\rvert$	$\lvert\Omega(x_3)\rvert$	COC
BK8	0.10449	1.7327×10^{-8}	5.0831×10^{-63}	6.9206×10^{-449}	8.0000
KP10	0.10449	1.7514×10^{-8}	6.4198×10^{-77}	6.9790×10^{-761}	10.0000
OSO10	0.10449	1.8070×10^{-8}	3.4911×10^{-78}	6.2812×10^{-775}	10.0000
NJ10	0.10449	4.6902×10^{-8}	2.5380×10^{-76}	9.1343×10^{-759}	10.0035
NAJJ10	0.10449	6.7004×10^{-9}	2.6528×10^{-85}	7.3337×10^{-849}	9.9991
XT10	0.10449	3.1329×10^{-8}	6.5196×10^{-78}	1.9895×10^{-774}	10.0011
NWM11	0.10449	1.5128×10^{-8}	5.4741×10^{-86}	2.6817×10^{-897}	10.4823

Table 6. Comparison of without-memory and with-memory algorithms after the first three (n = 3) iterations for $\Omega_6(x)$.

Method	$\lvert(x_1 - x_0)\rvert$	$\lvert(x_2 - x_1)\rvert$	$\lvert(x_3 - x_2)\rvert$	$\lvert\Omega(x_3)\rvert$	COC
BK8	0.05443	1.1772×10^{-12}	4.7565×10^{-98}	4.7039×10^{-780}	8.0000
KP10	0.05443	7.9291×10^{-14}	2.3095×10^{-132}	1.4133×10^{-1316}	10.0000
OSO10	0.05443	6.7134×10^{-7}	3.8971×10^{-62}	2.3485×10^{-613}	10.0000
NJ10	0.05443	3.1474×10^{-12}	4.7101×10^{-120}	3.6954×10^{-1197}	10.0000
NAJJ10	0.05443	3.9472×10^{-12}	4.5328×10^{-119}	2.5181×10^{-1187}	10.0000
XT10	0.05443	1.7397×10^{-12}	3.1353×10^{-122}	1.5779×10^{-1218}	10.0000
NWM11	0.05443	9.0914×10^{-13}	2.8262×10^{-135}	1.0309×10^{-1481}	11.0000

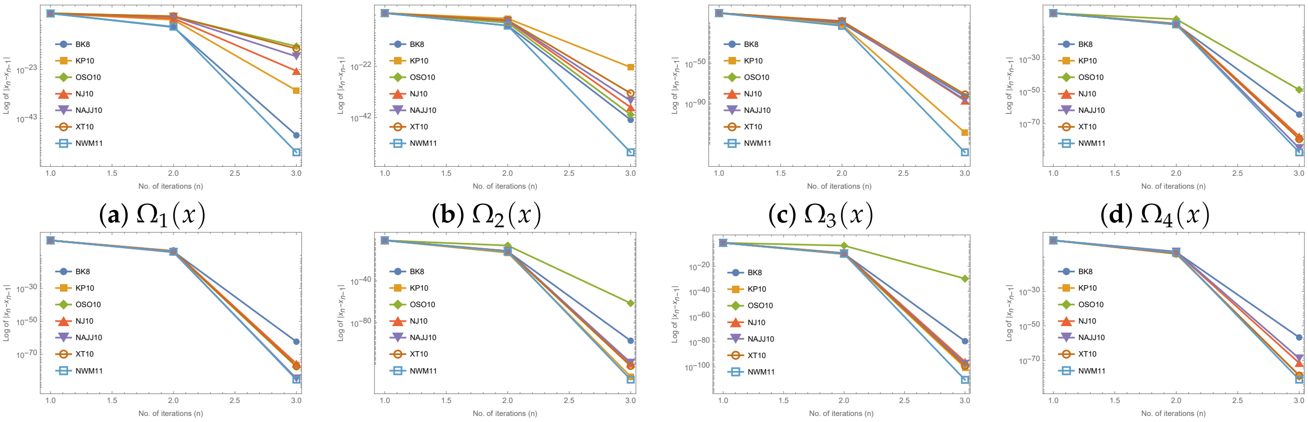
Table 7. Comparison of without-memory and with-memory algorithms after the first three (n = 3) iterations for $\Omega_7(x)$.

Method	$\lvert(x_1 - x_0)\rvert$	$\lvert(x_2 - x_1)\rvert$	$\lvert(x_3 - x_2)\rvert$	$\lvert\Omega(x_3)\rvert$	COC
BK8	0.04742	1.0014×10^{-10}	3.1891×10^{-80}	1.2499×10^{-634}	8.0000
KP10	0.04742	4.8045×10^{-11}	2.9840×10^{-101}	9.4424×10^{-1002}	10.0000
OSO10	0.04773	3.0677×10^{-4}	1.5964×10^{-30}	8.7071×10^{-293}	10.0380
NJ10	0.04742	3.3965×10^{-10}	1.4725×10^{-96}	1.2795×10^{-958}	10.0000
NAJJ10	0.04742	2.1575×10^{-10}	2.9855×10^{-98}	2.8465×10^{-975}	10.0000
XT10	0.04742	1.3920×10^{-10}	6.2415×10^{-100}	7.5966×10^{-992}	10.0000
NWM11	0.04742	8.9889×10^{-11}	6.1255×10^{-111}	3.3390×10^{-1211}	11.0000

Based on the numerical results in Tables 1–8 and Figure 1, it can be concluded that the newly proposed with-memory algorithm (NWM11) is competitive and demonstrates fast convergence toward the roots with minimal absolute residual error and a minimum error value in consecutive iterations compared to the aforementioned existing methods. Additionally, the numerical results indicate that the computational order of convergence supports the theoretical convergence order of the newly presented family of algorithms in the test functions.

Table 8. Comparison of without-memory and with-memory algorithms after the first three (n = 3) iterations for $\Omega_8(x)$.

Method	$\|(x_1 - x_0)\|$	$\|(x_2 - x_1)\|$	$\|(x_3 - x_2)\|$	$\|\Omega(x_3)\|$	COC
BK8	0.53000	7.6607×10^{-8}	1.9425×10^{-57}	3.8364×10^{-454}	8.0000
KP10	0.53000	7.5754×10^{-9}	1.7738×10^{-79}	1.0153×10^{-785}	10.0000
OSO10	*divergent*	*divergent*	*divergent*	*divergent*	*divergent*
NJ10	0.53000	7.8233×10^{-8}	2.5696×10^{-72}	3.8701×10^{-717}	10.0013
NAJJ10	0.53000	1.6015×10^{-7}	1.2265×10^{-69}	1.7842×10^{-689}	9.9797
XT10	0.53000	1.0726×10^{-8}	1.2785×10^{-79}	4.1929×10^{-790}	10.0180
NWM11	0.53000	3.5063×10^{-8}	5.5401×10^{-82}	3.9536×10^{-850}	10.4091



(a) $\Omega_1(x)$ **(b)** $\Omega_2(x)$ **(c)** $\Omega_3(x)$ **(d)** $\Omega_4(x)$

(e) $\Omega_5(x)$ **(f)** $\Omega_6(x)$ **(g)** $\Omega_7(x)$ **(h)** $\Omega_8(x)$

Figure 1. Comparison of the algorithms based on the error in consecutive iterations, $|x_n - x_{n-1}|$, after the first three iterations.

4. Conclusions

In this paper, a three-point with-memory iterative algorithm featuring two self-accelerating parameters is presented. By incorporating these parameters, computed using the Hermite interpolating polynomial, into an existing eighth-order method, its R-order of convergence is enhanced from 8 to 10.5208, and its efficiency index is enhanced from EI = 1.3161 to EI = 1.6011, without additional function evaluations. This algorithm not only accelerates convergence but also requires fewer function evaluations compared to other established algorithms, despite its higher convergence order. The findings in this paper demonstrate that the newly developed NWM11 algorithm offers superior performance with faster convergence and lower asymptotic constants, positioning it as a highly efficient alternative for solving nonlinear equations.

Author Contributions: Conceptualization, S.K.M. and S.P.; methodology, S.K.M. and S.P.; software, S.K.M., S.P. and C.E.S.; validation, S.K.M. and S.P.; formal analysis, S.P., S.K.M. and L.J.; resources, S.K.M.; writing—original draft preparation, S.P. and S.K.M.; writing—review and editing, S.K.M., S.P. and C.E.S.; visualization, S.P. and S.K.M.; supervision, S.P. and L.J. All authors have read and agreed to the published version of the manuscript.

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

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

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

References

1. Pho, K.H. Improvements of the Newton–Raphson method. *J. Comput. Appl. Math.* **2022**, *408*, 114106. [CrossRef]
2. Traub, J.F. *Iterative Methods for the Solution of Equations*; American Mathematical Soc.: Providence, RI, USA, 1982; Volume 312.
3. Gutierrez, C.; Gutierrez, F.; Rivara, M.C. Complexity of the bisection method. *Theor. Comput. Sci.* **2007**, *382*, 131–138. [CrossRef]
4. Sharma, H.; Kansal, M. A modified Chebyshev–Halley-type iterative family with memory for solving nonlinear equations and its stability analysis. *Math. Methods Appl. Sci.* **2023**, *46*, 12549–12569. [CrossRef]
5. Petković, I.; Herceg, D. Computers in mathematical research: The study of three-point root-finding methods. *Numer. Algorithms* **2020**, *84*, 1179–1198. [CrossRef]
6. Lu, Y.; Tang, Y. Solving Fractional Differential Equations Using Collocation Method Based on Hybrid of Block-pulse Functions and Taylor Polynomials. *Turk. J. Math.* **2021**, *45*, 1065–1078. [CrossRef]
7. Assari, P.; Dehghan, M. A meshless local Galerkin method for solving Volterra integral equations deduced from nonlinear fractional differential equations using the moving least squares technique. *Appl. Numer. Math.* **2019**, *143*, 276–299. [CrossRef]
8. Argyros, I.K.; Sharma, D.; Argyros, C.I.; Parhi, S.K.; Sunanda, S.K.; Argyros, M.I. Extended three step sixth order Jarratt- like methods under generalized conditions for nonlinear equations. *Arab. J. Math.* **2022**, *11*, 443–457. [CrossRef]
9. Temple, B.; Young, R. Inversion of a non-uniform difference operator and a strategy for Nash–Moser. *Methods Appl. Anal.* **2022**, *29*, 265–294. [CrossRef]
10. Putri, R.Y.; Wartono, W. Modifikasi metode Schroder tanpa turunan kedua dengan orde konvergensi empat. *Aksioma J. Mat. Dan Pendidik. Mat.* **2020**, *11*, 240–251. [CrossRef]
11. Argyros, I.K.; George, S. Local convergence of osada's method for finding zeros with multiplicity. In *Understanding Banach Spaces*; Sáanchez, D.G., Ed.; Nova Science Publishers: Hauppauge, NY, USA, 2019; pp. 147–151.
12. Postigo Beleña, C. Ostrowski's Method for Solving Nonlinear Equations and Systems. *J. Mech. Eng. Autom.* **2023**, *13*, 1–6. [CrossRef]
13. Ivanov, S.I. General Local Convergence Theorems about the Picard Iteration in Arbitrary Normed Fields with Applications to Super–Halley Method for Multiple Polynomial Zeros. *Mathematics* **2020**, *8*, 1599. [CrossRef]
14. Coclite, G.M.; Fanizzi, A.; Lopez, L.; Maddalena, F.; Pellegrino, S.F. Numerical methods for the nonlocal wave equation of the peridynamics. *Appl. Numer. Math.* **2020**, *155*, 119–139. [CrossRef]
15. Darvishi, M.T.; Barati, A. A fourth-order method from quadrature formulae to solve systems of nonlinear equations. *Appl. Math. Comput.* **2007**, 188, 257–261. [CrossRef]
16. Nisha, S.; Parida, P.K. Super-Halley method under majorant conditions in Banach spaces. *Cubo (Temuco)* **2020**, *22*, 55–70. [CrossRef]
17. Sharma, J.R.; Kumar, D.; Argyros, I.K. An efficient class of Traub-Steffensen-like seventh order multiple-root solvers with applications. *Symmetry* **2019**, *11*, 518. [CrossRef]
18. Ostrowski, A.M. *Solution of Equations in Euclidean and Banach Spaces*; Academic Press: Cambridge, MA, USA, 1973.
19. Choubey, N.; Jaiswal, J.P.; Choubey, A. Family of multipoint with memory iterative schemes for solving nonlinear equations. *Int. J. Appl. Comput. Math.* **2022**, *8*, 83. [CrossRef]
20. Sharma, E.; Mittal, S.K.; Jaiswal, J.P.; Panday, S. An Efficient Bi-Parametric With-Memory Iterative Method for Solving Nonlinear Equations. *Appl. Math.* **2023**, *3*, 1019–1033. [CrossRef]
21. Abdullah, S.; Choubey, N.; Dara, S. An efficient two-point iterative method with memory for solving non-linear equations and its dynamics. *J. Appl. Math. Comput.* **2024**, *70*, 285–315. [CrossRef]
22. Thangkhenpau, G.; Panday, S.; Mittal, S.K. New Derivative-Free Families of Four-Parametric with and Without Memory Iterative Methods for Nonlinear Equations. In *International Conference on Science, Technology and Engineering*; Springer Nature: Singapore, 2023; pp. 313–324.
23. Thangkhenpau, G.; Panday, S.; Mittal, S.K.; Jäntschi, L. Novel Parametric Families of with and without Memory Iterative Methods for Multiple Roots of Nonlinear Equations. *Mathematics* **2023**, *11*, 2036. [CrossRef]
24. Liu, C.S.; Chang, C.W.; Kuo, C.L. Memory-Accelerating Methods for One-Step Iterative Schemes with Lie Symmetry Method Solving Nonlinear Boundary-Value Problem. *Symmetry* **2024**, *16*, 120. [CrossRef]
25. Liu, C.S.; Chang, C.W. New Memory-Updating Methods in Two-Step Newton's Variants for Solving Nonlinear Equations with High Efficiency Index. *Mathematics* **2024**, *12*, 581. [CrossRef]
26. Erfanifar, R. A class of efficient derivative free iterative method with and without memory for solving nonlinear equations. *Comput. Math. Comput. Model. Appl.* **2022**, *1*, 20–26.
27. Howk, C.L.; Hueso, J.L.; Martínez, E.; Teruel, C. A class of efficient high-order iterative methods with memory for nonlinear equations and their dynamics. *Math. Meth. Appl. Sci.* **2018**, *41*, 7263–7282. [CrossRef]
28. Sharma, H.; Kansal, M.; Behl, R. An Efficient Two-Step Iterative Family Adaptive with Memory for Solving Nonlinear Equations and Their Applications. *Math. Comput. Appl.* **2022**, *27*, 97. [CrossRef]
29. Thangkhenpau, G.; Panday, S.; Bolundut, L.C.; Jäntschi, L. Efficient families of multi-point iterative methods and their self-acceleration with memory for solving nonlinear equations. *Symmetry* **2023**, *15*, 1546. [CrossRef]
30. Thangkhenpau, G.; Panday, S.; Chanu, W.H. New efficient bi-parametric families of iterative methods with engineering applications and their basins of attraction. *Result. Control Opt.* **2023**, *12*, 100243. [CrossRef]
31. Chanu, W.H.; Panday, S.; Thangkhenpau, G. Development of optimal iterative methods with their applications and basins of attraction. *Symmetry* **2022**, *14*, 2020. [CrossRef]

32. Choubey, N.; Jaiswal, J.P. Two-and three-point with memory methods for solving nonlinear equations. *Num. Anal. Appl.* **2017**, *10*, 74–89. [CrossRef]
33. Choubey, N.; Cordero, A.; Jaiswal, J.P.; Torregrosa, J.R. Dynamical techniques for analyzing iterative schemes with memory. *Complexity* **2018**, *2018*, 1232341. [CrossRef]
34. Wang, X.; Zhang, T. Some Newton-type iterative methods with and without memory for solving nonlinear equations. *Int. J. Comput. Meth.* **2014**, *11*, 1350078. [CrossRef]
35. Kong-ied, B. Two new eighth and twelfth order iterative methods for solving nonlinear equations. *Int. J. Math. Comput. Sci.* **2021**, *16*, 333–344.
36. Ortega, J.M.; Rheinboldt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2000.
37. Alefeld, G.; Herzberger, J. *Introduction to Interval Computation*; Academic Press: Berlin, Germany, 2012.
38. Devi, K.; Maroju, P. Local convergence study of tenth-order iterative method in Banach spaces with basin of attraction. *AIMS Math.* **2024**, *9*, 6648–6667. [CrossRef]
39. Ogbereyivwe, O.; Izevbizua, O.; Umar, S.S. Some high-order convergence modifications of the Householder method for nonlinear equations. *Commun. Nonlinear Anal.* **2023**, *11*, 1–11.
40. Weerakoon, S.; Fernando, T. A variant of Newton's method with accelerated third-order convergence. *Appl. Math. Lett.* **2000**, *13*, 87–93. [CrossRef]