

Article

Memory-Accelerating Methods for One-Step Iterative Schemes with Lie Symmetry Method Solving Nonlinear Boundary-Value Problem

Chein-Shan Liu ¹, Chih-Wen Chang ^{2,*} and Chung-Lun Kuo ¹

¹ Center of Excellence for Ocean Engineering, National Taiwan Ocean University, Keelung 202301, Taiwan; cslu@mail.ntou.edu.tw (C.-S.L.); d96510001@mail.ntou.edu.tw (C.-L.K.)

² Department of Mechanical Engineering, National United University, Miaoli 360302, Taiwan

* Correspondence: cwchang@nuu.edu.tw

Abstract: In this paper, some one-step iterative schemes with memory-accelerating methods are proposed to update three critical values $f'(r)$, $f''(r)$, and $f'''(r)$ of a nonlinear equation $f(x) = 0$ with r being its simple root. We can achieve high values of the efficiency index (E.I.) over the bound $2^{2/3} = 1.587$ with three function evaluations and over the bound $2^{1/2} = 1.414$ with two function evaluations. The third-degree Newton interpolatory polynomial is derived to update these critical values per iteration. We introduce relaxation factors into the Džunić method and its variant, which are updated to render fourth-order convergence by the memory-accelerating technique. We developed six types optimal one-step iterative schemes with the memory-accelerating method, rendering a fourth-order convergence or even more, whose original ones are a second-order convergence without memory and without using specific optimal values of the parameters. We evaluated the performance of these one-step iterative schemes by the computed order of convergence (COC) and the E.I. with numerical tests. A Lie symmetry method to solve a second-order nonlinear boundary-value problem with high efficiency and high accuracy was developed.

Keywords: optimal fourth-order one-step iterative schemes; memory-accelerating method; optimal combination function; optimal relaxation factor; Lie symmetry method



Citation: 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. <https://doi.org/10.3390/sym16010120>

Academic Editor: Ioan Raşa

Received: 19 December 2023

Revised: 11 January 2024

Accepted: 15 January 2024

Published: 19 January 2024



Copyright: © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

An elementary, yet very important problem is solving a nonlinear equation $f(x) = 0$. Given an initial guess x_0 , suppose that it is quite close to the real root r with $f(r) = 0$; we can approximate the nonlinear equation by

$$f(x_0) + f'(x_0)(x - x_0) = 0. \quad (1)$$

When $f'(x_0) \neq 0$, solving the equation for x yields

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}. \quad (2)$$

Along this line of thinking,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, \dots \quad (3)$$

is coined as the Newton method (NM), which exhibits quadratic convergence. Since then, there have arisen many studies, and this work continues to now, while different fourth-order methods have been used to modify the Newton method, which aim to more quickly and stably solve the nonlinear equations [1–7]. In general, the fourth-order iterative

methods are two-step with at least three function evaluations. Kung and Traub conjectured that a multi-step iterative scheme without memory based on m evaluations of functions has an optimal convergence order $p = 2^{m-1}$. When the fourth-order iterative method is optimal, the bound of the efficiency index (E.I.) is 1.587.

When the one-step iterative scheme with two function evaluations is considered, like the NM, its optimal order is two, while the E.I. reduces to 1.414. From this aspect, the multi-step iterative scheme is superior to the one-step iterative scheme with multi-function evaluations. In the local convergence analysis of the iterative scheme for solving nonlinear equations near the root r , three critical values $f'(r)$, $f''(r)$, and $f'''(r)$ and their ratios are dominant, which appear as the first three Taylor coefficients. In many iterative schemes, the accelerating parameter and optimal parameter are determined by these critical values. But, the root r is itself an unknown constant, such that the precise values of $f'(r)$, $f''(r)$, and $f'''(r)$ are not available. The primary goal of many memory methods is to develop a powerful updating technique based on the memory of the previous values of the variables to quickly obtain $f'(r)$, $f''(r)$, and $f'''(r)$ step-by-step, which will be used in several one-step iterative schemes developed in the paper for achieving high values of the computed order of convergence (COC) and the E.I. with the memory-accelerating technique.

Traub [8] was the first to develop a memory-dependent accelerating method from Steffensen's iterative scheme by giving x_0 and γ_0 :

$$\begin{cases} w_n = x_n + \gamma_n f(x_n), \\ x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \\ \gamma_{n+1} = -\frac{1}{f[x_n, x_{n+1}]} \end{cases} \quad (4)$$

With this modification, by taking the memory of (x_n, w_n) into account, the computational order of convergence is raised from 2 to at least 2.414. The iterative methods using information from the current and previous iteration are the methods with memory. In Equation (4), when x_n is a step variable, w_n is an adjoint variable, which does not have an iteration for itself. The role of w_n is different from y_n in the two-step iterative scheme. Later, we will introduce a supplementary variable, which just provides the extra datum used in the data interpolation, and its role is different from x_n and w_n .

In 2013, Džunić [9] proposed a modification of Steffensen's and Traub's iterative schemes by introducing two parameters γ and p in

$$\begin{cases} w_n = x_n + \gamma f(x_n), \\ x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n] + pf(w_n)}. \end{cases} \quad (5)$$

The error equation was derived as

$$e_{n+1} = [1 + \gamma f'(r)](a_2 + p)e_n^2 + \mathcal{O}(e_n^3), \quad (6)$$

where $a_2 = f''(r)/[2f'(r)]$ is a ratio of the second and first Taylor coefficients. Taking $\gamma = -1/f'(r)$ is sufficient for the vanishing of the second-order error term; hence, there is freedom to assign the value for the accelerating parameter p .

There are only a few papers that have been concerned with the one-step iterative schemes with memory [9–11]. In this paper, we develop several memory-dependent one-step iterative methods for a high-performance solution of nonlinear equations. The methodologies involve introducing a free parameter and a combination function, and then, they are optimized to raise the order of convergence, which is original and highly novel. The strategy for updating the values of the parameters with the memory-accelerating method can significantly speed up the convergence and raise the value of the E.I. to a limit bound.

The novelties involved in the paper are as follows:

- Introducing a free parameter in the existing or newly created model of the iterative scheme.
- Inserting a combination function into two iterative schemes; then, the parameter or combination function is optimized to raise the order of convergence.
- Several ideas presented here are novel and have not yet appeared in the literature, which can promote the development of fourth-order one-step iterative schemes while saving the computational cost.
- For the application of the derivative-free one-step iterative schemes, we developed a powerful Lie symmetry method to solve a second-order nonlinear boundary-value problem.

The rest of the paper's contents proceed as follows. In Section 2, we introduce two basic one-step iterative schemes, which are the starting point and motivate the development of the accelerating techniques to update the critical parameters $f'(r)$ and $f''(r)$ appearing in the third-order iterative schemes. Section 3 gives a detailed local convergence analysis of these two basic one-step iterative schemes; a new concept of the optimal combination of these two basic one-step iterative schemes is given in Theorem 3. In Section 4, some numerical experiments are carried out with the computed order of convergence (COC) to evaluate the performance upon comparing to some fourth-order optimal two-step iterative schemes. In Section 5, we introduce the updating techniques of the three critical parameters $f'(r)$, $f''(r)$, and $f'''(r)$ by using the memory-updating technique and the third-degree Newton interpolation polynomial. The idea of the supplementary variable is introduced, and the result is the first memory-accelerating technique. In Section 6, we first derive a new third-order iterative scheme as a variant of Džunić's method; then, the updating techniques of two parameters and three parameters by the memory methods are developed; the second memory-accelerating technique is developed. In Section 7, we improve Džunić's method and propose the new optimal combination methods; three memory-accelerating techniques are developed. In Section 8, we introduce a relaxation factor into Džunić's method, and the optimal value of the relaxation factor is derived; the sixth memory-accelerating technique is developed. As a practical application, a Lie symmetry method is developed in Section 9 to solve the second-order boundary-value problem. Finally, we conclude the achievements in Section 10.

2. Preliminaries

Mathematically speaking, $f(x) = 0$ is equivalent to

$$xf'(x) - b_0xf(x) = xf'(x) - (1 + b_0x)f(x), \quad (7)$$

where the constant b_0 is to be determined. If x_n is known, we can find the next x_{n+1} by solving

$$[f'(x_n) - b_0f(x_n)]x_{n+1} = x_nf'(x_n) - b_0x_nf(x_n) - f(x_n). \quad (8)$$

Upon viewing the terms including x_n as the coefficient on both sides and dividing by $f'(x_n) - b_0f(x_n)$, we can obtain

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n) - b_0f(x_n)}, \quad (9)$$

which includes a parameter $b_0 \neq 0 \in \mathbb{R}$ to be assigned. The above iterative scheme was developed in [12] for a one-step continuation Newton-like method; it is referred to as Wu's method [12]. The iterative scheme (9) was used by Lee et al. [13], Zafar et al. [14], and Thangkhempau et al. [15] as the first step in the multi-step iterative schemes for finding multiple zeros. Recently, Singh and Singh [16] and also Singh and Argyros [17] gave a detailed dynamical analysis of the continuous Newton-like method (9).

As a variant of Equation (9), we consider

$$x_{n+1} = x_n - \frac{f(x_n)}{a_0 + b_0 f'(x_n)}, \quad (10)$$

which is obtained from the equivalent form of $f(x) = 0$:

$$a_0 x + b_0 x f(x) = a_0 x - (1 - b_0 x) f(x). \quad (11)$$

The iterative scheme (10) includes two parameters $a_0, b_0 \neq 0 \in \mathbb{R}$ to be assigned. The iterative scheme (10) is referred to as Liu's method [18].

In 2021, Liu et al. [19] verified that the iterative scheme (10) is in general of one-order convergence; however, if we take $a_0 = f'(r)$, where r is a simple root of $f(x) = 0$, the order rises to two; moreover, if we take $a_0 = f'(r)$ and $b_0 = f''(r)/(2f'(r))$, the order further rises to three. This technique is somewhat like the method of using accelerating parameters to speed up the convergence, whose optimal values can be determined by the convergence analysis.

The memory methods reuse the information from the previous iteration; they are not required to evaluate the function and its derivative at any new point, but it is required to store this information. The so-called R-convergence order of the memory method increases, and at the same time, the E.I. may go over the corresponding one without the memory method. Džunić [9] earlier developed an efficient two-parameter method for solving nonlinear equations by using the memory technique and Newton polynomial interpolation to determine the accelerating parameters. For the progress of the memory methods with accelerating parameters in the multi-step iterative schemes, one can refer to [20–29].

3. Convergence Analysis

Comparing to the Newton method in Equation (3), Equation (9) is still applicable when $f'(x_n) = 0$. But, in this situation, the Newton method would fail, which restricts the practical application of the Newton method. The iterative scheme (9) has some remarkable advantages over the Newton method [30]. It is interesting to study the convergence behaviors of the iterative scheme (9) and its variant in the iterative scheme (10).

Wang and Liu [31] proposed a two-step iterative scheme as an extension of Equation (9):

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n) - \alpha f(x_n)}, \\ x_{n+1} = y_n - \frac{f(y_n)}{f'(x_n) - \alpha f(x_n)}, \end{cases} \quad (12)$$

where $\alpha \in [-1, 1]$ is a parameter. The error formula was proven to be

$$e_{n+1} = (\alpha^2 - 3a_2\alpha + 2a_2^2)e_n^3. \quad (13)$$

The iterative scheme (12) is not the optimal one, which, with three function evaluations, leads to the third-order convergence, not the fourth-order convergence. This motivated us to further investigate the local convergence property of the iterative schemes (9) and (10). A new idea of the optimal combination of the iterative schemes (9) and (10) is introduced, such that a one-step optimal fourth-order iterative scheme can be achieved, which is better than the two-step iterative scheme (12).

Theorem 1. *The iterative scheme (9) for solving $f(x) = 0$ has third-order convergence, if*

$$b_0 = a_2 = \frac{f''(r)}{2f'(r)}. \quad (14)$$

Proof. Let

$$e_n = x_n - r \quad (15)$$

which is small when x_n and r are sufficiently close. Repeating Equation (15) for $n + 1$ and subtracting yield

$$e_{n+1} = e_n + x_{n+1} - x_n. \quad (16)$$

As usual, we have

$$f(x_n) = f'(r)[e_n + a_2e_n^2 + a_3e_n^3 + a_4e_n^4 + \dots], \quad a_n := \frac{f^{(n)}(r)}{n!f'(r)}, \quad n = 2, \dots, \quad (17)$$

$$f'(x_n) = f'(r)[1 + 2a_2e_n + 3a_3e_n^2 + 4a_4e_n^3 + \dots]. \quad (18)$$

Then, using Equation (14), we have

$$\begin{aligned} \frac{f(x_n)}{f'(x_n) - b_0f(x_n)} &= \frac{e_n + a_2e_n^2 + a_3e_n^3 + a_4e_n^4 + \dots}{1 + (2a_2 - b_0)e_n + (3a_3 - b_0a_2)e_n^2 + (4a_4 - b_0a_3)e_n^3 + \dots} \\ &= e_n + C_3e_n^3 + C_4e_n^4 + \dots, \end{aligned} \quad (19)$$

where

$$\begin{aligned} C_3 &= (b_0 - 2a_2)^2 + b_0a_2 - 2a_3 + a_2(b_0 - 2a_2) = b_0^2 - 2b_0a_2 + 2a_2^2 - 2a_3 = a_2^2 - 2a_3, \\ C_4 &= a_3(b_0 - 2a_2) + a_2(b_0a_2 - 3a_3) + a_2(b_0 - 2a_2)^2 \\ &\quad + b_0a_3 - 3a_4 + 2(b_0 - 2a_2)(b_0a_2 - 3a_3). \end{aligned} \quad (20)$$

Inserting Equation (19) into Equation (9) and using Equation (16), we can obtain

$$\begin{aligned} e_{n+1} &= e_n - e_n - C_3e_n^3 - C_4e_n^4 + \dots = -C_3e_n^3 - C_4e_n^4 + \dots \\ &= (2a_3 - a_2^2)e_n^3 + \mathcal{O}(e_n^4). \end{aligned} \quad (21)$$

This ends the proof of Theorem 1. \square

Theorem 2. *If*

$$a_0 = f'(r), \quad b_0 = a_2 = \frac{f''(r)}{2f'(r)}, \quad (22)$$

the iterative scheme (10) is cubically convergent.

Proof. It can be proven similarly by inserting Equation (17) into Equation (10):

$$\begin{aligned} \frac{f(x_n)}{a_0 + b_0f(x_n)} &= \frac{e_n + a_2e_n^2 + a_3e_n^3 + a_4e_n^4 + \dots}{1 + b_0e_n + b_0a_2e_n^2 + b_0a_3e_n^3 + \dots} \\ &= e_n + D_3e_n^3 + D_4e_n^4 + \dots, \end{aligned} \quad (23)$$

where we use Equation (22), for which

$$\begin{aligned} D_3 &= a_3 - 2a_2b_0 + b_0^2 = a_3 - a_2^2, \\ D_4 &= 3b_0^2a_2 - b_0a_2^2 - 2b_0a_3 + a_4. \end{aligned} \quad (24)$$

Inserting Equation (23) into Equation (10) and using Equation (16), we can obtain

$$\begin{aligned} e_{n+1} &= e_n - e_n - D_3e_n^3 - D_4e_n^4 + \dots = -D_3e_n^3 - D_4e_n^4 + \dots \\ &= (a_2^2 - a_3)e_n^3 + \mathcal{O}(e_n^4). \end{aligned} \quad (25)$$

This ends the proof of Theorem 2. \square

The combination of any two iterative schemes of the same order cannot yield a new iterative scheme whose convergence order can be raised by one. The conditions for the success of the combination are that, in the two error equations of these two iterative schemes,

the coefficients preceding a_2^2 cannot be the same, and at the same time, the coefficients preceding a_3 cannot be the same.

Theorem 3. As a combination of the iterative schemes (9) and (10) with a function Q , the iterative scheme:

$$x_{n+1} = x_n - \frac{Qf(x_n)}{f'(x_n) - b_0f(x_n)} - \frac{(1-Q)f(x_n)}{a_0 + b_0f(x_n)} \quad (26)$$

has fourth-order convergence, if

$$a_0 = f'(r), \quad b_0 = \frac{f''(r)}{2f'(r)}, \quad (27)$$

$$Q = \frac{D_3}{D_3 - C_3} = \frac{a_3 - a_2^2}{3a_3 - 2a_2^2} = \frac{1}{2} + \frac{f'(r)f'''(r)}{6[f''(r)^2 - f'(r)f'''(r)]}. \quad (28)$$

Proof. Inserting Equations (17) and (18) into Equation (26) leads to

$$\begin{aligned} & \frac{Qf(x_n)}{f'(x_n) - b_0f(x_n)} + \frac{(1-Q)f(x_n)}{a_0 + b_0f(x_n)} \\ &= \frac{Q(e_n + a_2e_n^2 + a_3e_n^3 + a_4e_n^4 + \dots)}{1 + (2a_2 - b_0)e_n + (3a_3 - b_0a_2)e_n^2 + (4a_4 - b_0a_3)e_n^3 + \dots} \\ &+ \frac{(1-Q)(e_n + a_2e_n^2 + a_3e_n^3 + a_4e_n^4 + \dots)}{1 + b_0e_n + b_0a_2e_n^2 + b_0a_3e_n^3 + \dots} \\ &= Qe_n + QC_3e_n^3 + QC_4e_n^4 + (1-Q)e_n + (1-Q)D_3e_n^3 + (1-Q)D_4e_n^4 + \dots \\ &= e_n + QC_4e_n^4 + (1-Q)D_4e_n^4 + \dots, \end{aligned} \quad (29)$$

where we use Equations (27) and (28). Inserting Equation (29) into Equation (26) and using Equation (16), we can obtain

$$e_{n+1} = e_n - e_n - QC_4e_n^4 - (1-Q)D_4e_n^4 + \dots = -[QC_4 + (1-Q)D_4]e_n^4 + \dots \quad (30)$$

This completes the proof of Theorem 3. \square

4. Numerical Experiments

For the purposes of comparison, we list the fourth-order iterative schemes developed by Chun [3]:

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \left[1 + 2\frac{f(y_n)}{f'(x_n)} + \frac{f^2(y_n)}{f^2(x_n)}\right] \frac{f(y_n)}{f'(x_n)}, \end{cases} \quad (31)$$

by King [4]:

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f(x_n) + \beta f(y_n)}{f(x_n) + (\beta - 2)f(y_n)} \frac{f(y_n)}{f'(x_n)}, \end{cases} \quad (32)$$

where $\beta \in \mathbb{R}$, and by Chun and Ham [1]:

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{4f^2(x_n) + 6f(x_n)f(y_n) + 3f^2(y_n)}{4f^2(x_n) - 2f(x_n)f(y_n) - f^2(y_n)} \frac{f(y_n)}{f'(x_n)}. \end{cases} \quad (33)$$

Equations (31)–(33) are fourth-order optimal iterative schemes with the E.I. = $\sqrt[3]{4} = 1.587$. However, the E.I. of the iterative scheme (26) can be larger, as shown in Section 5. Furthermore, the iterative scheme (26) is a single-step one, rather than the two steps of the iterative schemes (31)–(33).

The convergence criteria are given by

$$|x_{n+1} - x_n| < \varepsilon \text{ and } |f(x_{n+1})| < \varepsilon, \quad (34)$$

where $\varepsilon = 10^{-15}$ is fixed for all tests. In [32], the numerically computed order of convergence (COC) is approximated by

$$\text{COC} := \frac{\ln |(x_{n+1} - r)/(x_n - r)|}{\ln |(x_n - r)/(x_{n-1} - r)|}, \quad (35)$$

where r is a solution of $f(x) = 0$.

The iterative schemes in Equations (9), (10), and (26) are named, respectively, Algorithm 1, Algorithm 2, and Algorithm 3. We considered a simple case of $f(x) = x^3 - x = 0$ and specify how to calculate the COC. By starting from $x_0 = 1.5$, the NM achieves the root $r = 1$ with seven iterations, Algorithms 1 and 2 with five iterations, and Algorithm 3 with four iterations. For each triple of the data of x , we can compute the COC by Equation (35). The triple (x_{n+1}, x_n, x_{n-1}) comprises the last three values of x before the convergence, and we set the convergence value of x to be r with $f(r) = 0$. If $\ln |(x_{n+1} - r)/(x_n - r)|$ is not computable due to $x_{n+1} = r$, we can shift the triple forward to (x_n, x_{n-1}, x_{n-2}) , and so on.

By the data from Table 1, we take the COC of the NM to be 1.999, of Algorithm 1 to be 3.046, of Algorithm 2 to be 2.968 and of Algorithm 3 to be 4.899. As expected, both Algorithms 1 and 2 have near third-order convergence; however, Algorithm 3 with COC = 4.899 is greater than the theoretical value of fourth-order.

Table 1. The comparison of different methods for the COCs computed. ×: undefined.

n	2	3	4	5	6	7
NM	1.594	1.841	1.978	1.999	×	×
Algorithm 1	2.779	3.046	×	×		
Algorithm 2	2.634	2.968	×	×		
Algorithm 3	4.899	×	×			

The test examples are given by

$$g_1(x) = x^3 + 4x^2 - 10, \quad (36)$$

$$g_2(x) = x^2 - e^x - 3x + 2, \quad (37)$$

$$g_3(x) = (x - 1)^3 - 2, \quad (38)$$

$$g_4(x) = (x + 2)e^x - 1, \quad (39)$$

$$g_5(x) = \sin^2 x - x^2 + 1. \quad (40)$$

The corresponding solutions are, respectively, $r_1 = 1.3652300134$, $r_2 = 0.2575302854$, $r_3 = 2.2599210499$, $r_4 = -0.442854401002$, and $r_5 = 1.4044916482$.

Algorithm 1 was tested by Wu [12,30], and Algorithm 2 was tested by Liu et al. [19]. Because Algorithm 3 is a new iterative scheme, we tested it for the above examples. In Table 2, for different functions, we list the NI obtained by the presently developed Algorithm 3, which are compared to the NM, the method of Jarratt [33] (JM), the method of Traub–Ostrowski [8] (TM), the method of King [4] (KM) with $\beta = 3$, and the method of Chun and Ham [1] (CM).

Algorithm 3 theoretically has fourth-order convergence with the optimal values of the parameters. For the first example, $g_1(x) = 0$, Algorithm 3 converges faster than other fourth-order iterative schemes. For other examples, Algorithm 3 is much better than the NM and is competitive with the other fourth-order iterative schemes, JM, TM, KM, and CM.

Table 2. The comparison of different methods for the number of iterations.

Functions	x_0	NM	JM	TM	KM	CM	Algorithm 3
g_1	−0.3	55	46	46	49	9	5
g_2	0	5	3	3	3	3	3
g_3	3	7	4	4	4	4	4
g_4	3.5	11	6	6	7	7	5
g_5	1	7	4	4	8	4	4

5. Updating Three Parameters by Memory Method

In order to achieve the fourth-order convergence of the iterative scheme in Equation (26), the values of $f'(r)$, $f''(r)$, and $f'''(r)$ must be known a priori; however, these values are unknown because the root r is an unknown constant.

Therefore, we introduce a supplementary variable predicted by the Newton scheme used in the data interpolation:

$$w_n = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (41)$$

Based on the updated data of x_{n+1} and $w_{n+1} = x_{n+1} - f(x_{n+1})/f'(x_{n+1})$ and the previous data of (x_n, w_n) , we can construct a third-degree Newton interpolatory polynomial by

$$\begin{aligned} \mathcal{N}_3(x) = & f(x_n) + f[x_n, w_n](x - x_n) + f[x_n, w_n, x_{n+1}](x - x_n)(x - w_n) \\ & + f[x_n, w_n, x_{n+1}, w_{n+1}](x - x_n)(x - w_n)(x - x_{n+1}), \end{aligned} \quad (42)$$

where

$$\begin{aligned} f[x_n, w_n] &= \frac{f(x_n) - f(w_n)}{x_n - w_n}, \quad f[x_n, w_n, x_{n+1}] = \frac{f[x_n, w_n] - f[w_n, x_{n+1}]}{x_n - x_{n+1}}, \\ f[x_n, w_n, x_{n+1}, w_{n+1}] &= \frac{f[x_n, w_n, x_{n+1}] - f[w_n, x_{n+1}, w_{n+1}]}{x_n - w_{n+1}}. \end{aligned} \quad (43)$$

It is easy to derive

$$\begin{aligned} \mathcal{N}'_3(x) &= f[x_n, w_n] + f[x_n, w_n, x_{n+1}](2x - x_n - w_n) \\ &+ f[x_n, w_n, x_{n+1}, w_{n+1}][(x - w_n)(x - x_{n+1}) + (x - x_n)(x - x_{n+1}) + (x - x_n)(x - w_n)], \\ \mathcal{N}''_3(x) &= 2f[x_n, w_n, x_{n+1}] + f[x_n, w_n, x_{n+1}, w_{n+1}](6x - 2x_n - 2w_n - 2x_{n+1}), \\ \mathcal{N}'''_3(x) &= 6f[x_n, w_n, x_{n+1}, w_{n+1}]. \end{aligned} \quad (44)$$

We update the values of $A = f'(r)$, $B = f''(r)/[2f'(r)]$ and Q in Equation (28) by

$$A_{n+1} = \mathcal{N}'_3(w_{n+1}), \quad B_{n+1} = \frac{\mathcal{N}''_3(w_{n+1})}{2\mathcal{N}'_3(w_{n+1})}, \quad (45)$$

$$Q_{n+1} = \frac{1}{2} + \frac{\mathcal{N}'_3(w_{n+1})\mathcal{N}'''_3(w_{n+1})}{6[\mathcal{N}''_3(w_{n+1})^2 - \mathcal{N}'_3(w_{n+1})\mathcal{N}'''_3(w_{n+1})]}. \quad (46)$$

Now, we have the first memory-accelerating technique for the iterative scheme (26) in Theorem 3, which reads as (i) giving A_0 , B_0 , and Q_0 and $w_0 = x_0 - f(x_0)/f'(x_0)$ and (ii) performing for $n = 0, 1, \dots$:

$$x_{n+1} = x_n - \frac{Q_n f(x_n)}{f'(x_n) - B_n f(x_n)} - \frac{(1 - Q_n) f(x_n)}{A_n + B_n f(x_n)}, \quad (47)$$

$$w_{n+1} = x_{n+1} - \frac{f(x_{n+1})}{f'(x_{n+1})}, \quad (48)$$

$$\begin{aligned} A_{n+1} &= f[x_n, w_n] + f[x_n, w_n, x_{n+1}](2w_{n+1} - x_n - w_n) + f[x_n, w_n, x_{n+1}, w_{n+1}] \\ &\times [(w_{n+1} - w_n)(w_{n+1} - x_{n+1}) + (w_{n+1} - x_n)(w_{n+1} - x_{n+1}) + (w_{n+1} - x_n)(w_{n+1} - w_n)], \\ D_{n+1} &= 2f[x_n, w_n, x_{n+1}] + f[x_n, w_n, x_{n+1}, w_{n+1}](6w_{n+1} - 2x_n - 2w_n - 2x_{n+1}), \\ E_{n+1} &= 6f[x_n, w_n, x_{n+1}, w_{n+1}], \\ B_{n+1} &= \frac{D_{n+1}}{2A_{n+1}}, \\ Q_{n+1} &= \frac{1}{2} + \frac{A_{n+1}E_{n+1}}{6[D_{n+1}^2 - A_{n+1}E_{n+1}]}. \end{aligned} \quad (49)$$

Some numerical tests of the first memory-accelerating technique for the iterative scheme are listed in Table 3, for which we can observe that the values of the COC are very large. Because the differential term $f'(x_n)$ was included, the E.I. = $(\text{COC})^{1/3}$ was computed. All E.I.s are greater than the E.I. = 1.587 of the optimal fourth-order iterative scheme without memory; they have the same number of function evaluations. The last four E.I.s are also greater than the E.I. = 1.682 of the optimal eighth-order iterative scheme without memory and with four function evaluations.

Table 3. The NI, COC, and E.I. for the method by updating three parameters, A , B , and Q , in the first memory-accelerating technique.

Functions	x_0	$[A_0, B_0]$	Q_0	NI	COC	E.I. = $(\text{COC})^{1/3}$
g_1	1.3	[17.09, 0.48]	−0.1	3	4.207	1.614
g_2	0.2	[−3.94, −0.027]	−0.5	3	5.433	1.758
g_3	2.1	[3.99, 0.87]	0	3	5.090	1.720
g_4	−0.5	[2.503, 0.627]	0.3	3	6.095	1.827
g_5	1.3	[−1.926, 0.927]	0	3	6.795	1.894

6. A New Fourth-Order Iterative Scheme with Memory Updating

Below, we will develop the memory-accelerating technique without using the differential term $f'(x_n)$. The test examples will be changed to

$$f_1(x) = x^3 + 4x^2 - 10, \quad (50)$$

$$f_2(x) = (x - 1)(x^6 + 1/x^6 + 4) \sin(x^2), \quad (51)$$

$$f_3(x) = (x - 1)^3 - 2, \quad (52)$$

$$f_4(x) = (x + 2)e^x - 1, \quad (53)$$

$$f_5(x) = \sin^2 x - x^2 + 1. \quad (54)$$

The corresponding solutions are, respectively, $r_1 = 1.3652300134$, $r_2 = 1$, $r_3 = 2.2599210499$, $r_4 = -0.442854401002$, and $r_5 = 1.4044916482$.

In order to compare the numerical results with that given in [9], a new function f_2 is added in Equation (51). For consistency, the other four functions are rewritten as f_1, f_3, f_4 , and f_5 to replace the g_1, g_3, g_4 , and g_5 appearing in Equations (36) and (38)–(40).

In this section, a new one-step iterative scheme with the aid of a supplementary variable and a relaxation factor is introduced. The detailed convergence analysis is derived. Then, we accelerate the introduced parameters by the memory-updating technique.

6.1. A New Third-Order Result

We address the following iterative scheme:

$$\begin{cases} w_n = x_n + \gamma f(x_n), \\ x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n] + pf(x_n)}, \end{cases} \quad (55)$$

which is a novel one-step iterative scheme with w_n an adjoint variable. Equation (55) is a variant of Džunić's method [9], where $f[x_n, w_n] + pf(w_n)$ appears in the denominator, not $f[x_n, w_n] + pf(x_n)$.

To derive the convergence order of Equation (55), let

$$f(x_n) = F(e_n)f'(r) = f'(r)[e_n + a_2e_n^2 + a_3e_n^3 + a_4e_n^4 + \dots], \quad (56)$$

$$f'(x_n) = G(e_n)f'(r) = f'(r)[1 + 2a_2e_n + 3a_3e_n^2 + 4a_4e_n^3 + \dots], \quad (57)$$

$$f''(x_n) = 2H(e_n)f'(r); H(e_n) = a_2 + 3a_3e_n + 6a_4e_n^2 + \dots \quad (58)$$

Theorem 4. The iterative scheme (55) has third-order convergence:

$$e_{n+1} = -[(1 + 3\eta)a_3 + a_2^2]e_n^3 + \mathcal{O}(e_n^4), \quad (59)$$

where $\eta \in (-1, 1]$ is a relaxation factor, and

$$\gamma = -\frac{1 + \eta}{f'(r)}, \quad p = \eta a_2. \quad (60)$$

If we take

$$\eta = -\frac{1}{3} - \frac{a_2^2}{3a_3} = -\frac{1}{3} - \frac{f''(r)^2}{2f'(r)f'''(r)}, \quad (61)$$

then the order of convergence is further increased to four.

Proof. Using the Taylor series yields

$$f(w_n) - f(x_n) = f'(x_n)(w_n - x_n) + \frac{1}{2}f''(x_n)(w_n - x_n)^2 + \dots, \quad (62)$$

where

$$w_n - x_n = \gamma f'(r)F(e_n). \quad (63)$$

Hence, it follows from Equations (57) and (58) that

$$f[x_n, w_n] = f'(r)G(e_n) + \gamma f'(r)^2 F(e_n)H(e_n). \quad (64)$$

Inserting Equations (56) and (64) into the second one in Equation (55), we have

$$e_{n+1} = e_n - \frac{F(e_n)}{G(e_n) + \gamma f'(r)F(e_n)H(e_n) + pF(e_n)}. \quad (65)$$

Through some operations, we can obtain

$$\begin{aligned} G(e_n) + \gamma f'(r)F(e_n)H(e_n) + pF(e_n) &= 1 + 2a_2e_n + 3a_3e_n^2 + 4a_4e_n^3 + \dots \\ &+ \gamma f'(r)(e_n + a_2e_n^2 + a_3e_n^3 + \dots)(a_2 + 3a_3e_n + 6a_4e_n^2 + \dots) \\ &+ p(e_n + a_2e_n^2 + a_3e_n^3 + \dots), \end{aligned} \quad (66)$$

which can be written as

$$G(e_n) + \gamma f'(r)F(e_n)H(e_n) + pF(e_n) = 1 + P_1e_n + P_2e_n^2 + P_3e_n^3 + \dots, \quad (67)$$

where

$$\begin{aligned} P_1 &= 2a_2 + a_2\gamma f'(r) + p, \\ P_2 &= 3a_3 + (a_2^2 + 3a_3)\gamma f'(r) + pa_2, \\ P_3 &= 4a_4 + \gamma f'(r)(6a_4 + 4a_2a_3) + pa_3. \end{aligned} \quad (68)$$

Then, Equation (65) is further reduced to

$$\begin{aligned} e_{n+1} &= e_n - \frac{F(e_n)}{G(e_n) + \gamma f'(r)F(e_n)H(e_n) + pF(e_n)} \\ &= e_n - (e_n + a_2e_n^2 + a_3e_n^3 + \dots)[1 - P_1e_n - P_2e_n^2 - P_3e_n^3 + (P_1e_n + P_2e_n^2 + P_3e_n^3)^2] \\ &= (P_1 - a_2)e_n^2 + (P_2 + a_2P_1 - a_3 - P_1^2)e_n^3 + (P_3 + a_2P_2 + a_3P_1 - 2P_1P_2 - a_4)e_n^4. \end{aligned} \quad (69)$$

Letting $P_1 - a_2 = 0$ for the vanishing of e_n^2 , we can derive a relation between γ and p :

$$a_2[1 + \gamma f'(r)] + p = 0 \quad (70)$$

Taking $1 + \gamma f'(r) = -\eta$, we prove Equation (60). By using $P_1 = a_2$ and Equation (68), the coefficient preceding e_n^3 in Equation (69) can be simplified to

$$(P_2 + a_2P_1 - a_3 - P_1^2)e_n^3 = (P_2 - a_3)e_n^3 = [3a_3 + (a_2^2 + 3a_3)\gamma f'(r) + pa_2 - a_3]e_n^3, \quad (71)$$

which, further using $\gamma f'(r) = -(1 + \eta)$ and $p = \eta a_2$, reduces to

$$(P_2 + a_2P_1 - a_3 - P_1^2)e_n^3 = [2a_3 - (1 + \eta)(a_2^2 + 3a_3) + \eta a_2^2]e_n^3 = -[(1 + 3\eta)a_3 + a_2^2]e_n^3. \quad (72)$$

If Equation (61) is satisfied, the error equation becomes $e_{n+1} = \mathcal{O}(e_n^4)$. This completes the proof of Theorem 4. \square

Theorem 4 gives us a clue to achieving a fourth-order iterative scheme (55), if γ and p are given by Equations (60) and (61). However, there are three unknown parameters, $f'(r)$, $f''(r)$, and $f'''(r)$. We will apply the memory-accelerating technique to adapt the values of γ and p in Equation (55) per iteration. The accuracy of this memory-dependent adaption technique depends on how many current and previous data are taken into account. Similar to what was performed in [9], we can further estimate the lower bound of the convergence order of such a memory-accelerating method, upon giving the updating technique for γ and p . Instead of deriving the formulas of this kind of estimation, we use the numerical values of the COC to display the numerical performance.

6.2. Updating Two Parameters

To mimic the memory-updating procedure in Section 5, we can obtain a memory method of the iterative scheme in Equation (55) by (i) giving A_0 , B_0 , and η and $w_0 = x_0 - (1 + \eta)f(x_0)/A_0$ and (ii) performing for $n = 0, 1, \dots$:

$$x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n] + \eta B_n f(x_n)}, \quad (73)$$

$$A_{n+1} = f[x_n, w_n] + f[x_n, w_n, x_{n+1}](2x_{n+1} - x_n - w_n), \quad (74)$$

$$w_{n+1} = x_{n+1} - \frac{(1 + \eta)f(x_{n+1})}{A_{n+1}}, \quad (75)$$

$$\begin{aligned}
C_{n+1} &= f[x_n, w_n] + f[x_n, w_n, x_{n+1}](2w_{n+1} - x_n - w_n) + f[x_n, w_n, x_{n+1}, w_{n+1}] \\
&\times [(w_{n+1} - w_n)(w_{n+1} - x_{n+1}) + (w_{n+1} - x_n)(w_{n+1} - x_{n+1}) + (w_{n+1} - x_n)(w_{n+1} - w_n)], \\
D_{n+1} &= 2f[x_n, w_n, x_{n+1}] + f[x_n, w_n, x_{n+1}, w_{n+1}](6w_{n+1} - 2x_n - 2w_n - 2x_{n+1}), \\
B_{n+1} &= \frac{D_{n+1}}{2C_{n+1}}.
\end{aligned} \tag{76}$$

In the above iterative scheme, η is a given constant value of the parameter.

To demonstrate the usefulness of the above iterative scheme, we consider the solution of $f_1(x) = 0$. We fix $x_0 = 1.3$, $A_0 = 17.09$ and $B_0 = 0.4798$ and vary the values of η in Table 4. Because only two function evaluations are required, the E.I. = $(\text{COC})^{1/2}$ was computed.

Table 4. The NI, COC, and E.I. for the method by updating two parameters, A and B .

η	−0.9	−0.6	−0.3	−0.2	0.1	0.3	0.6	0.9
NI	3	3	3	3	3	3	3	3
COC	3.413	3.114	3.233	3.443	3.081	2.974	2.870	2.795
E.I. = $(\text{COC})^{1/2}$	1.847	1.765	1.798	1.855	1.755	1.724	1.694	1.672

6.3. Updating Three Parameters

Table 4 reveals an optimal value of η , such that the COC and E.I. are the best. Indeed, by setting the coefficient preceding e_n^3 as zero in Theorem 4, we can truly obtain a fourth-order iterative scheme, whose η is determined by Equation (61).

Thus, we have the second memory-accelerating technique for the iterative scheme (55) in Theorem 4 using η in Equation (61), which reads as (i) giving A_0 , B_0 , and η_0 and $w_0 = x_0 - (1 + \eta_0)f(x_0)/A_0$, and (ii) performing for $n = 0, 1, \dots$

$$x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n] + \eta_n B_n f(x_n)}, \tag{77}$$

$$A_{n+1} = f[x_n, w_n] + f[x_n, w_n, x_{n+1}](2x_{n+1} - x_n - w_n), \tag{78}$$

$$w_{n+1} = x_{n+1} - \frac{(1 + \eta_n)f(x_{n+1})}{A_{n+1}}, \tag{79}$$

$$\begin{aligned}
C_{n+1} &= f[x_n, w_n] + f[x_n, w_n, x_{n+1}](2w_{n+1} - x_n - w_n) + f[x_n, w_n, x_{n+1}, w_{n+1}] \\
&\times [(w_{n+1} - w_n)(w_{n+1} - x_{n+1}) + (w_{n+1} - x_n)(w_{n+1} - x_{n+1}) + (w_{n+1} - x_n)(w_{n+1} - w_n)], \\
D_{n+1} &= 2f[x_n, w_n, x_{n+1}] + f[x_n, w_n, x_{n+1}, w_{n+1}](6w_{n+1} - 2x_n - 2w_n - 2x_{n+1}), \\
E_{n+1} &= 6f[x_n, w_n, x_{n+1}, w_{n+1}],
\end{aligned} \tag{80}$$

$$\begin{aligned}
B_{n+1} &= \frac{D_{n+1}}{2C_{n+1}}, \\
\eta_{n+1} &= -\frac{1}{3} - \frac{B_{n+1}^2}{2A_{n+1}E_{n+1}}.
\end{aligned} \tag{81}$$

Some numerical tests of the second memory-accelerating technique for the iterative scheme are listed in Table 5. For the equation $f_2(x) = 0$, the COC = 3.535 obtained is slightly larger than the COC = 3.48 obtained in [9].

Table 5. The NI, COC, and E.I. for the method by updating three parameters, A , B , and η , in the second memory-accelerating technique.

Functions	x_0	$[A_0, B_0]$	η_0	NI	COC	E.I. = (COC) ^{1/2}
f_1	1	[5.25, 0.9048]	−0.3	4	4.680	2.163
f_2	0.9	[14.44, −0.33]	4	4	3.535	1.880
f_3	2	[3.25, −0.231]	−0.3	4	3.992	1.998
f_4	−0.5	[6.13, 0.54]	10	4	4.237	2.058
f_5	1.5	[0.548, −1.318]	0.3	5	3.610	1.900

7. Improvement of Džunić's Method and Optimal Combination

In this section, we improve Džunić's method [9] by deriving the optimal parameters and their accelerating techniques. The idea of two optimal combinations of Džunić's and Wu's method, and Džunić's and Liu's method are introduced. Then, three new one-step iterative schemes with the memory-updating techniques are developed.

7.1. Improvement of Džunić's Memory Method

As was performed in [9], γ and p in Equation (5) were taken to be $\gamma = -1/f'(r)$ and $p = -a_2$. To guarantee the third-order convergence of Džunić's method, $1 + \gamma f'(r) = 0$ is sufficient, as shown in Equation (6). Therefore, there exists the freedom to choose the value of p .

Theorem 5. For solving $f(x) = 0$, the iterative scheme (5) has third-order convergence:

$$e_{n+1} = -[a_3 + pa_2 + a_2^2]e_n^3 + \mathcal{O}(e_n^4), \quad (82)$$

if $\gamma = -1/f'(r)$. If we take, furthermore,

$$p = -a_2 - \frac{a_3}{a_2} = -\frac{f''(r)}{2f'(r)} - \frac{f'''(r)}{3f''(r)}, \quad (83)$$

then Equation (5) has fourth-order convergence:

$$e_{n+1} = \mathcal{O}(e_n^4). \quad (84)$$

Proof. By Equations (62) and (63) and Equations (56)–(58), we have

$$f(w_n) = f'(r)F(e_n) + \gamma f'(r)F(e_n)G(e_n) + \gamma^2 f'(r)^3 F^2(e_n)H(e_n). \quad (85)$$

At the same time, Equation (65) is modified to

$$e_{n+1} = e_n - \frac{F(e_n)}{G(e_n) + \gamma f'(r)F(e_n)[H(e_n) + pG(e_n)] + pF(e_n) + p\gamma^2 f'(r)^2 H(e_n)F^2(e_n)}. \quad (86)$$

Since we do not have interest in the details of the fourth-order error equation, we write

$$\begin{aligned} & G(e_n) + \gamma f'(r)F(e_n)H(e_n) + p\gamma f'(r)F(e_n)G(e_n) + pF(e_n) + p\gamma^2 f'(r)^2 H(e_n)F^2(e_n) \\ & = 1 + P_1 e_n + P_2 e_n^2 + \dots, \end{aligned} \quad (87)$$

where

$$\begin{aligned} P_1 &= 2a_2 + a_2\gamma f'(r) + p[1 + \gamma f'(r)], \\ P_2 &= 3a_3 + (a_2^2 + 3a_3)\gamma f'(r) + pa_2 + pa_2[3\gamma f'(r) + \gamma^2 f'(r)^2]. \end{aligned} \quad (88)$$

Then, Equation (86) is reduced to

$$\begin{aligned}
 e_{n+1} &= e_n - (e_n + a_2e_n^2 + a_3e_n^3 + \dots)[1 - P_1e_n - P_2e_n^2 - P_3e_n^3 + (P_1e_n + P_2e_n^2 + P_3e_n^3)^2] \\
 &= (P_1 - a_2)e_n^2 + (P_2 + a_2P_1 - a_3 - P_1^2)e_n^3 + \mathcal{O}(e_n^4),
 \end{aligned}
 \tag{89}$$

where $P_1 - a_2 = (a_2 + p)(1 + \gamma f'(r))$ was derived in [9].

Letting $P_1 = a_2$ for the vanishing of e_n^2 , we can derive

$$P_2 + a_2P_1 - a_3 - P_1^2 = 2a_3 + (a_2^2 + 3a_3)\gamma f'(r) + pa_2[1 + 3\gamma f'(r) + \gamma^2 f'(r)^2] = -a_3 - a_2^2 - pa_2,
 \tag{90}$$

if we take $\gamma f'(r) = -1$. By using Equation (83), the coefficient preceding e_n^3 is reduced to zero. This completes the proof of Theorem 5. \square

The third memory-accelerating technique for the iterative scheme (5) in Theorem 5 using p in Equation (83) reads as (i) giving A_0 and p_0 and $w_0 = x_0 - f(x_0)/A_0$ and (ii) performing for $n = 0, 1, \dots$:

$$x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n] + p_n f(w_n)},
 \tag{91}$$

$$A_{n+1} = f[x_n, w_n] + f[x_n, w_n, x_{n+1}](2x_{n+1} - x_n - w_n),
 \tag{92}$$

$$w_{n+1} = x_{n+1} - \frac{f(x_{n+1})}{A_{n+1}},
 \tag{93}$$

$C_{n+1}, D_{n+1}, E_{n+1}$ computed by Equation (80),

$$p_{n+1} = -\frac{D_{n+1}}{2C_{n+1}} - \frac{E_{n+1}}{3D_{n+1}}.
 \tag{94}$$

Some numerical tests of the third memory-accelerating technique for the iterative scheme are listed in Table 6. We can notice that, for the equation $f_2(x) = 0$, the COC = 4.002 obtained is larger than the COC = 3.48 obtained in [9].

Table 6. The NI, COC, and E.I. for the method by updating two parameters, A and p , in the third memory-accelerating technique.

Functions	x_0	A_0	p_0	NI	COC	E.I. = (COC) ^{1/2}
f_1	1	5.25	-0.3	3	5.169	2.274
f_2	1.3	4.44	-0.1	4	4.022	2.005
f_3	2	5.11	-3	3	3.126	2.031
f_4	-0.5	4.83	-10	3	7.227	2.688
f_5	1.3	-1.926	-10	3	6.945	2.635

7.2. Optimal Combination of Džunić's and Wu's Iterative Methods

Theorem 6. If

$$\gamma = -\frac{1}{f'(r)}, \quad p = a_2, \quad b_0 = a_2,
 \tag{95}$$

then the combination of the iterative schemes (5) and (9):

$$\begin{cases} w_n = x_n + \gamma f(x_n), \\ x_{n+1} = x_n - \frac{Qf(x_n)}{f[x_n, w_n] + pf(w_n)} - \frac{(1-Q)f(x_n)}{f'(x_n) - b_0f(x_n)} \end{cases}
 \tag{96}$$

has fourth-order convergence, where

$$Q = \frac{2a_3 - a_2^2}{3a_3 + a_2^2}.
 \tag{97}$$

Proof. By Equations (20), (21), and (82), we have

$$e_{n+1} = -\left[(1-Q)(2a_2^2 - a_3) + Qa_3 + Qpa_2 + Qa_2^2\right]e_n^3 + \mathcal{O}(e_n^4). \quad (98)$$

If we take $p = a_2$ and Q by Equation (97), then Equation (98) becomes

$$e_{n+1} = \mathcal{O}(e_n^4). \quad (99)$$

This ends the proof of Theorem 6. \square

The fourth memory-accelerating technique for the iterative scheme (96) in Theorem 6 reads as (i) giving A_0 , B_0 , and Q_0 and $w_0 = x_0 - f(x_0)/A_0$ and (ii) performing for $n = 0, 1, \dots$:

$$x_{n+1} = x_n - \frac{Q_0 f(x_n)}{f[x_n, w_n] - B_n f(w_n)} - \frac{(1-Q_0)f(x_n)}{f'(x_n) - B_n f(x_n)}, \quad (100)$$

$$A_{n+1} = f[x_n, w_n] + f[x_n, w_n, x_{n+1}](2x_{n+1} - x_n - w_n), \quad (101)$$

$$w_{n+1} = x_{n+1} - \frac{f(x_{n+1})}{A_{n+1}}, \quad (102)$$

C_{n+1} , D_{n+1} , E_{n+1} computed by Equation (80),

$$B_{n+1} = \frac{D_{n+1}}{2C_{n+1}},$$

$$G_{n+1} = \frac{E_{n+1}}{6C_{n+1}},$$

$$Q_{n+1} = \frac{2G_{n+1} - D_{n+1}^2}{3G_{n+1} + D_{n+1}^2}. \quad (103)$$

Some numerical tests of the fourth memory-accelerating technique for the iterative scheme are listed in Table 7, which shows that the values of COC are high, which, however, need the differential term $f'(x_n)$. We can notice that, for the equation $f_2(x) = 0$, the COC = 6.643 obtained is much larger than the COC = 3.48 obtained in [9].

Table 7. The NI, COC, and E.I. for the method by updating three parameters, A , B , and Q , in the fourth memory-accelerating technique.

Functions	x_0	A_0	B_0	Q_0	NI	COC	E.I. = (COC) ^{1/3}
f_1	1.2	23.75	0.39	5	3	5.156	1.728
f_2	1.4	-1.97	8.34	0.5	5	6.643	1.880
f_3	2.2	3.25	-0.23	2	3	9.127	2.090
f_4	-0.5	4.83	0.61	0.5	3	7.517	1.959
f_5	1.3	-3.733	0.475	0.1	3	7.091	1.921

7.3. Optimal Combination of Džunić's and Liu's Iterative Methods

Theorem 7. If

$$\gamma = -\frac{1}{f'(r)}, \quad p = -a_2, \quad a_0 = f'(r), \quad b_0 = a_2, \quad (104)$$

then the combination of the iterative schemes (5) and (10):

$$\begin{cases} w_n = x_n + \gamma f(x_n), \\ x_{n+1} = x_n - \frac{Qf(x_n)}{f[x_n, w_n] + pf(w_n)} - \frac{(1-Q)f(x_n)}{f'(r) + b_0 f(x_n)} \end{cases} \quad (105)$$

has fourth-order convergence, where

$$Q = 1 - \frac{a_3}{a_2^2}. \tag{106}$$

Proof. By Equations (24) and (82), we have

$$e_{n+1} = - \left[(1 - Q)(a_2^2 - a_3) + Qa_3 + Qpa_2 + Qa_2^2 \right] e_n^3 + \mathcal{O}(e_n^4). \tag{107}$$

If we take $p = -a_2$ and Q by Equation (106), then Equation (107) becomes

$$e_{n+1} = \mathcal{O}(e_n^4). \tag{108}$$

This ends the proof of Theorem 7. \square

The fifth memory-accelerating technique for the iterative scheme (105) in Theorem 7 is (i) giving $A_0 = C_0, B_0,$ and Q_0 and $w_0 = x_0 - f(x_0)/A_0$ and (ii) performing for $n = 0, 1, \dots$

$$x_{n+1} = x_n - \frac{Q_0 f(x_n)}{f[x_n, w_n] - B_n f(w_n)} - \frac{(1 - Q_0) f(x_n)}{C_n + B_n f(x_n)}, \tag{109}$$

$$A_{n+1} = f[x_n, w_n] + f[x_n, w_n, x_{n+1}](2x_{n+1} - x_n - w_n), \tag{110}$$

$$w_{n+1} = x_{n+1} - \frac{f(x_{n+1})}{A_{n+1}}, \tag{111}$$

$C_{n+1}, D_{n+1}, E_{n+1}$ computed by Equation (80),

$$B_{n+1} = \frac{D_{n+1}}{2C_{n+1}},$$

$$Q_{n+1} = 1 - \frac{E_{n+1}}{D_{n+1}^2}. \tag{112}$$

Some numerical tests of the fifth memory-accelerating technique for the iterative scheme are listed in Table 8. We can notice that, for the equation $f_2(x) = 0$, the COC = 5.013 obtained is larger than the COC = 3.48 obtained in [9].

Table 8. The NI, COC, and E.I. for the method by updating three parameters, A, B and Q , in the fifth memory-accelerating technique.

Functions	x_0	A_0	B_0	Q_0	NI	COC	E.I. = (COC) ^{1/2}
f_1	1.2	9.99	0.67	2	3	5.028	2.242
f_2	1.3	5.06	469,683	1	6	5.013	2.239
f_3	2.2	3.25	-0.231	2	3	9.127	3.021
f_4	-0.5	4.83	0.61	5	3	7.343	2.742
f_5	1.3	-3.733	0.475	0.1	3	7.091	2.663

8. Modification of Džunić’s Method

As was performed in [9], γ and p in Equation (5) were taken to be $\gamma = -1/f'(r)$ and $p = -a_2$. To guarantee the third-order convergence of Džunić’s method, $p = -a_2$ is sufficient, as shown in Equation (6). Therefore, there exists the freedom to chose the value of γ . We propose a modification of Džunić’s method by

$$\begin{cases} w_n = x_n - \beta \frac{f(x_n)}{f'(r)}, \\ x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n] + pf(w_n)}, \end{cases} \tag{113}$$

that is, we take $p = -a_2$ and $\gamma f'(r) = -\beta$, where β is a relaxation factor to be determined for increasing the order of convergence. If we take $\beta = 1$, Džunić's method is recovered. The present modification is different from that analyzed in Sections 7.1 and 7.2, where p is a free parameter.

Theorem 8. For solving $f(x) = 0$, the iterative scheme (113) with $p = -a_2$ has third-order convergence:

$$e_{n+1} = [(2 - 3\beta)a_3 - a_2^2(\beta^2 - 2\beta + 1)]e_n^3 + \mathcal{O}(e_n^4), \quad (114)$$

If we take, furthermore,

$$\beta = \frac{2 - 3q + \sqrt{9q^2 - 4q}}{2}, \quad q = \frac{a_3}{a_2^2}, \quad (115)$$

then Equation (113) has fourth-order convergence:

$$e_{n+1} = \mathcal{O}(e_n^4). \quad (116)$$

Proof. Let $\beta = 1 + \eta$. It follows from Equations (89) and (90) that

$$e_{n+1} = (P_2 + a_2P_1 - a_3 - P_1^2)e_n^3 + \mathcal{O}(e_n^4) = (P_2 - a_3)e_n^3 + \mathcal{O}(e_n^4), \quad (117)$$

$$\begin{aligned} P_2 - a_3 &= 2a_3 + (a_2^2 + 3a_3)\gamma f'(r) - a_2^2[1 + 3\gamma f'(r) + \gamma^2 f'(r)^2] \\ &= (2 - 3\beta)a_3 - a_2^2[\beta^2 - 2\beta + 1], \end{aligned} \quad (118)$$

where $P_1 = a_2$, $p = -a_2$, and $\gamma f'(r) = -\beta$ were inserted. For the vanishing of e_n^3 , we need to solve

$$\beta^2 + \left(\frac{3a_3}{a_2^2} - 2\right)\beta + 1 - \frac{2a_3}{a_2^2} = 0, \quad (119)$$

whose solution is given by Equation (115). \square

The sixth memory-accelerating technique for the iterative scheme (113) in Theorem 8 is (i) giving A_0 , B_0 , and β_0 and $w_0 = x_0 - \beta_0 f(x_0)/A_0$ and (ii) for $n = 0, 1, \dots$, performing

$$x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n] - B_n f(w_n)}, \quad (120)$$

$$A_{n+1} = f[x_n, w_n] + f[x_n, w_n, x_{n+1}](2x_{n+1} - x_n - w_n), \quad (121)$$

$$w_{n+1} = x_{n+1} - \beta_{n+1} \frac{f(x_{n+1})}{A_{n+1}}, \quad (122)$$

C_{n+1} , D_{n+1} , E_{n+1} computed by Equation (80),

$$B_{n+1} = \frac{D_{n+1}}{2C_{n+1}},$$

$$q_{n+1} = \frac{E_{n+1}}{D_{n+1}^2},$$

$$\beta_{n+1} = \frac{2 - 3q_{n+1} + \sqrt{9q_{n+1}^2 - 4q_{n+1}}}{2}. \quad (123)$$

Some numerical tests of the sixth memory-accelerating technique for the iterative scheme are listed in Table 9. For the equation $f_2(x) = 0$, the COC = 8.526 is much larger than the COC = 3.48 obtained in [9].

In Equation (113), if we take $\beta = 1$, then Džunić's method is recovered. In Table 10, the values of the COC are compared, which shows that the COC obtained by the sixth memory-accelerating technique is larger than that obtained by Džunić's memory method.

Table 9. The NI, COC, and E.I. for the method by updating three parameters, A , B , and β , in the sixth memory-accelerating technique.

Functions	x_0	A_0	B_0	β_0	NI	COC	E.I. = (COC) ^{1/2}
f_1	1.2	17.09	0.48	2	3	5.028	2.423
f_2	1.3	−13.79	0.721	2	5	8.526	2.920
f_3	2.2	3.25	−0.23	2	3	7.879	2.807
f_4	−0.5	22.29	0.394	1	3	5.502	2.346
f_5	1.3	−3.733	0.475	1	3	5.081	2.254

Table 10. The values of the COC for Džunić's memory method and the sixth memory accelerating technique.

Functions	f_1	f_2	f_3	f_4	f_5
Džunić's method	4.819	4.990	5.852	4.496	4.469
Equation (113)	5.028	8.526	7.879	5.502	5.081

Notice that, by using the suggested initial values of $x_0 = 1.3$, $A_0 = 10$, and $B_0 = 0.1$ given in [9] and using

$$\text{COC} := \frac{\ln |(f(x_{n+1})/f(x_n))|}{\ln |f(x_n)/f(x_{n-1})|}, \quad (124)$$

instead of that in Equation (35), we can obtain the COC = 3.538 for $f_2(x) = 0$, which is close to the lower bound of 3.56 derived in [9]. However, using $A_0 = 15.63$ and $B_0 = -0.077$, the COC = 4.990 is obtained by Equation (35), and the COC = 3.895 is obtained by Equation (124). Most papers have used Equation (35) to compute the COC. In any case, the lower bound derived in [9] may underestimate the true value of the COC. As shown in Table 10, all COCs obtained by Džunić's memory method are greater than 3.56.

9. A Lie Symmetry Method

As a practical application of the proposed iterative schemes, we developed a Lie symmetry method based on the Lie group $SL(2, \mathbb{R})$ to solve the second-order nonlinear boundary-value problem. This Lie symmetry method was first developed in [34] for computing the eigenvalues of the generalized Sturm–Liouville problem.

Let

$$u''(y) = \frac{3}{2}u^2(y), \quad y \in (0, 1), \quad (125)$$

$$u(0) = 4, \quad u(1) = 1, \quad (126)$$

whose exact solution is

$$u(y) = \frac{4}{(y+1)^2}. \quad (127)$$

The conventional shooting method is assumed to have an unknown initial slope $u'(0) = x$ and integrated with Equation (125) with the initial conditions $u(0) = 4$ and $u'(0) = x$, which results in an implicit equation $f(x) = u(1, x) - 1 = 0$ to be solved.

From Equation (125), a nonlinear system consists of two first-order ordinary differential equations:

$$\frac{d}{dy} \begin{bmatrix} u(y) \\ u'(y) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{3u(y)}{2} & 0 \end{bmatrix} \begin{bmatrix} u(y) \\ u'(y) \end{bmatrix}. \quad (128)$$

Let

$$\mathbf{A} := \begin{bmatrix} 0 & 1 \\ \frac{3u(y)}{2} & 0 \end{bmatrix} \quad (129)$$

being the coefficient matrix. The Lie symmetry system in Equation (128) permits a Lie group symmetry $SL(2, \mathbb{R})$, a two-dimensional real-valued special linear group, because of $\text{tr}\mathbf{A} = 0$.

By using the closure property of the Lie group, there exists a $\mathbf{G} \in SL(2, \mathbb{R})$, such that the following mapping holds:

$$\begin{bmatrix} u(1) \\ u'(1) \end{bmatrix} = \mathbf{G} \begin{bmatrix} u(0) \\ u'(0) \end{bmatrix}, \quad (130)$$

where

$$\mathbf{G}(x) = \exp[\mathbf{A}(\hat{u})], \quad (131)$$

$$\hat{u}(x) = xu(0) + (1-x)u(1) = 3x + 1, \quad (132)$$

and $x \in (0, 1)$ is an unknown weighting factor to be determined.

We can derive

$$\mathbf{G} = \begin{bmatrix} \cosh \sqrt{\frac{3\hat{u}}{2}} & \sqrt{\frac{2}{3\hat{u}}} \sinh \sqrt{\frac{3\hat{u}}{2}} \\ \sqrt{\frac{3\hat{u}}{2}} \sinh \sqrt{\frac{3\hat{u}}{2}} & \cosh \sqrt{\frac{3\hat{u}}{2}} \end{bmatrix}. \quad (133)$$

Then, it follows from Equations (130) and (133) that

$$u(1) = \cosh \sqrt{\frac{3\hat{u}}{2}} u(0) + \sqrt{\frac{2}{3\hat{u}}} \sinh \sqrt{\frac{3\hat{u}}{2}} u'(0), \quad (134)$$

$$u'(1) = \sqrt{\frac{3\hat{u}}{2}} \sinh \sqrt{\frac{3\hat{u}}{2}} u(0) + \cosh \sqrt{\frac{3\hat{u}}{2}} u'(0). \quad (135)$$

Since $u(0) = 4$ and $u(1) = 1$ are given, we can obtain $u'(0)$ from Equations (132) and (134):

$$u'(0) = \frac{1 - 4 \cosh \sqrt{\frac{3\hat{u}}{2}}}{\sqrt{\frac{2}{3\hat{u}}} \sinh \sqrt{\frac{3\hat{u}}{2}}} = \frac{1 - 4 \cosh \sqrt{\frac{3(3x+1)}{2}}}{\sqrt{\frac{2}{3(3x+1)}} \sinh \sqrt{\frac{3(3x+1)}{2}}}. \quad (136)$$

It is interesting that the unknown slope $u'(0)$ can be derived in Equation (136) by using the Lie symmetry method, which is more powerful than the traditional shooting method, of which no such explicit formula for $u'(0)$ can be obtained.

Now, we apply the fourth-order Runge–Kutta method to integrate Equation (125) with the initial conditions $u(0) = 4$ and $u'(0)$ given in Equation (136) in terms of x . The right-end value must satisfy $u(1) - 1 = 0$, which is an implicit function of x . We take $N = 2000$ steps in the Runge–Kutta method and fix the initial guess $x_0 = 0.5$. In Table 11, we compare the NI, the error of $u'(0)$, and the maximum error of u obtained by comparing to Equation (127). The weighting factor $x = 0.614814522263784$ is obtained, such that $u'(0)$ computed from Equation (136) is very close to the exact one $u'(0) = -8$.

Table 11. For Equations (125) and (126), comparing the performances of the second, third, fifth, and sixth memory-accelerating techniques in the solution of a second-order nonlinear boundary-value problem by the Lie symmetry method.

Methods	Second	Third	Fifth	Sixth
NI	4	3	3	4
Error of $u'(0)$	2.736×10^{-14}	2.736×10^{-14}	2.736×10^{-14}	2.736×10^{-14}
Maximum error of u	3.997×10^{-14}	3.997×10^{-14}	3.997×10^{-14}	3.997×10^{-14}

Instead of the Dirichlet boundary conditions in Equation (126), we consider mixed-type boundary conditions:

$$u(0) = 4, \quad u'(1) = -1. \quad (137)$$

Now, from Equation (135), it follows that

$$u'(0) = -\frac{1 + 4\sqrt{\frac{3(3x+1)}{2}} \sinh \sqrt{\frac{3(3x+1)}{2}}}{\cosh \sqrt{\frac{3(3x+1)}{2}}}. \quad (138)$$

In Table 12, we compare the NI, the error of $u'(0)$, and the maximum error of u . The weighting factor $x = 0.5602528690788923$ is obtained, such that the $u'(0)$ computed from Equation (138) is very close to the exact one $u'(0) = -8$.

Table 12. For Equations (125) and (137), comparing the performances of the second, third, fifth, and sixth memory-accelerating techniques in the solution of a second-order nonlinear boundary-value problem by the Lie symmetry method.

Methods	Second	Third	Fifth	Sixth
NI	5	3	3	3
Error of $u'(0)$	2.665×10^{-13}	2.647×10^{-13}	2.647×10^{-13}	2.647×10^{-13}
Maximum error of u	4.352×10^{-14}	4.397×10^{-14}	4.397×10^{-14}	4.397×10^{-14}

10. Conclusions

In this paper, we addressed five one-step iterative methods: A (Wu's method), B (Liu's method), C (a novel method), D (Džunić's method), and E (a modification of Džunić's method). Without using specific values of the parameters and without memory, they all have second-order convergence; when specific optimal values of the parameters are used, they have third-order convergence. Three critical values of $f'(r)$, $f''(r)$, and $f'''(r)$ are parameters in a_2 and a_3 , which are crucial for achieving good performance in the designed iterative scheme, such that the coefficients (involving a_2 and a_3) preceding e_n^2 and e_n^3 are zeros. We introduced a combination function, which is determined by raising the order of convergence. The optimal combination of A and B can generate a fourth-order one-step iterative scheme. When the values of the parameters and combination function were obtained by a memory-accelerating method with the third-degree Newton polynomial to interpolate the previous and current data, we obtained the first memory-accelerating technique to realize a fourth-order one-step iterative scheme.

In the novel method C, a relaxation factor appeared. If we used the memory-accelerating method for updating the values of the relaxation factor and other parameters, we obtained the second memory-accelerating technique to realize the fourth-order convergence of the derived novel one-step iterative scheme.

We mathematically improved Džunić's method to an iterative scheme with fourth-order convergence, and the third memory-accelerating technique was developed to realize a fourth-order one-step iterative scheme based on Džunić's memory method.

The optimal combination of A and D generated the fourth memory-accelerating technique to realize a fourth-order one-step iterative scheme based on an optimal combination function between Džunić's and Wu's methods. The optimal combination of B and D generated the fifth memory-accelerating technique to realize a fourth-order one-step iterative scheme based on an optimal combination function between Džunić's and Liu's methods.

In E, we finally introduced a relaxation factor into Džunić's method, which is optimized to the fourth-order convergence by the sixth memory-accelerating technique.

In the first and fourth memory-accelerating techniques, three evaluations of the function and its derivative were required. In contrast, the second, third, fifth, and sixth memory-accelerating techniques needed two evaluations of the function. Numerical tests confirmed that these fourth-order one-step iterative schemes performed very well with high values of the COC and E.I. Among them, the fifth memory-accelerating technique was the best one with the $\text{COC} > 5.028$ and $\text{E.I.} > 2.413$ for all testing examples. Recall that the efficiency index of the optimal fourth-order two-step iterative scheme with three evaluations of the function and without memory is the $\text{E.I.} = \sqrt[3]{4} = 1.587$.

As an application of the derivative-free one-step iterative schemes with the second, third, fifth, and sixth memory-accelerating technique, a second-order nonlinear boundary-value problem was solved by the Lie symmetry method. It is remarkable that the Lie symmetry method can derive the unknown initial slope to be an explicit formula of the weighting factor x , whose implicit nonlinear equation $f(x) = 0$ can be solved with high efficiency and high accuracy.

The basic iterative schemes in Equations (9) and (10) are applicable to find the multiple roots of a nonlinear equation, for instance $f_3(x) = (x - 1)^3 - 2 = 0$ with a triple root $r = 2^{1/3} + 1$. It can be treated well by the proposed accelerated one-step iterative schemes. As for the system of nonlinear equations, more studies are needed by extending the presented accelerating techniques.

Author Contributions: Conceptualization, C.-S.L.; Methodology, C.-S.L. and C.-W.C.; Software, C.-S.L. and C.-W.C.; Validation, C.-S.L. and C.-W.C.; Formal analysis, C.-S.L. and C.-W.C.; Investigation, C.-S.L., C.-W.C. and C.-L.K.; Resources, C.-S.L., C.-W.C. and C.-L.K.; Data curation, C.-S.L., C.-W.C. and C.-L.K.; Writing—original draft, C.-S.L.; Writing—review & editing, C.-W.C.; Visualization, C.-S.L., C.-W.C. and C.-L.K.; Supervision, C.-S.L. and C.-W.C.; Project administration, C.-W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding authors. The data are not publicly available due to restrictions privacy.

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

References

1. Chun, C.; Ham, Y. Some fourth-order modifications of Newton's method. *Appl. Math. Comput.* **2008**, *197*, 654–658. [[CrossRef](#)]
2. Noor, M.A.; Noor, K.I.; Waseem, M. Fourth-order iterative methods for solving nonlinear equations. *Int. J. Appl. Math. Eng. Sci.* **2010**, *4*, 43–52.
3. Chun, C. Some fourth-order iterative methods for solving nonlinear equations. *Appl. Math. Comput.* **2008**, *195*, 454–459. [[CrossRef](#)]
4. King, R. A family of fourth-order iterative methods for nonlinear equations. *SIAM J. Numer. Anal.* **1973**, *10*, 876–879. [[CrossRef](#)]
5. Li, S. Fourth-order iterative method without calculating the higher derivatives for nonlinear equation. *J. Algorithms Comput. Technol.* **2019**, *13*. [[CrossRef](#)]
6. Chun, C. Certain improvements of Chebyshev-Halley methods with accelerated fourth-order convergence. *Appl. Math. Comput.* **2007**, *189*, 597–601. [[CrossRef](#)]
7. Kuo, J.; Li, Y.; Wang, X. Fourth-order iterative methods free from second derivative. *Appl. Math. Comput.* **2007**, *184*, 880–885.
8. Traub, J.F. *Iterative Methods for the Solution of Equations*; Prentice-Hall: New York, NY, USA, 1964.
9. Džunić, J. On efficient two-parameter methods for solving nonlinear equations. *Numer. Algorithms* **2013**, *63*, 549–569.
10. Haghani, F.K. A modified Steffensen's method with memory for nonlinear equations. *Int. J. Math. Model. Comput.* **2015**, *5*, 41–48.
11. Khdhr, F.W.; Saeed, R.K.; Soleymani, F. Improving the computational efficiency of a variant of Steffensen's method for nonlinear equations. *Mathematics* **2019**, *7*, 306. [[CrossRef](#)]

12. Wu, X.Y. A new continuation Newton-like method and its deformation. *Appl. Math. Comput.* **2000**, *112*, 75–78. [[CrossRef](#)]
13. Lee, M.Y.; Kim, Y.I.; Magrenán, A.A. On the dynamics of tri-parametric family of optimal fourthorder multiple-zero finders with a weight function of the principal m^{th} root of a function-function ratio. *Appl. Math. Comput.* **2017**, *315*, 564–590.
14. Zafar, F.; Cordero, A.; Torregrosa, J.R. Stability analysis of a family of optimal fourth-order methods for multiple roots. *Numer. Algorithms* **2019**, *81*, 947–981. [[CrossRef](#)]
15. 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**, *14*, 2036. [[CrossRef](#)]
16. Singh, M.K.; Singh, A.K. A derivative free globally convergent method and its deformations. *Arab. J. Math.* **2021**, *10*, 481–496. [[CrossRef](#)]
17. Singh, M.K.; Argyros, I.K. The dynamics of a continuous Newton-like method. *Mathematics* **2022**, *10*, 3602. [[CrossRef](#)]
18. Liu, C.S. A new splitting technique for solving nonlinear equations by an iterative scheme. *J. Math. Res.* **2020**, *12*, 40–48. [[CrossRef](#)]
19. Liu, C.S.; Hong, H.K.; Lee, T.L. A splitting method to solve a single nonlinear equation with derivative-free iterative schemes. *Math. Comput. Simul.* **2021**, *190*, 837–847. [[CrossRef](#)]
20. Džunić, J.; Petković, M.S. On generalized biparametric multipoint root finding methods with memory. *J. Comput. Appl. Math.* **2014**, *255*, 362–375.
21. Wang, X.; Zhang, T. A new family of Newton-type iterative methods with and without memory for solving nonlinear equations. *Calcolo* **2014**, *51*, 1–15. [[CrossRef](#)]
22. Cordero, A.; Lotfi, T.; Bakhtiari, P.; Torregrosa, J.R. An efficient two-parameter family with memory for nonlinear equations. *Numer. Algorithms* **2015**, *68*, 323–335. [[CrossRef](#)]
23. Chanu, W.H.; Panday, S.; Thangkhenpau, G. Development of optimal iterative methods with their applications and basins of attraction. *Symmetry* **2022**, *14*, 2020. [[CrossRef](#)]
24. Lotfi, T.; Soleymani, F.; Noori, Z.; Kiliçman, A.; Haghani, F.K. Efficient iterative methods with and without memory possessing high efficiency indices. *Discret. Dyn. Nat. Soc.* **2014**, *2014*, 912796. [[CrossRef](#)]
25. Wang, X. An Ostrowski-type method with memory using a novel self-accelerating parameter. *J. Comput. Appl. Math.* **2018**, *330*, 710–720. [[CrossRef](#)]
26. Chicharro, F.I.; Cordero, A.; Torregrosa, J.R. Dynamics of iterative families with memory based on weight functions procedure. *Appl. Math. Comput.* **2019**, *354*, 286–298. [[CrossRef](#)]
27. Torkashvand, V.; Kazemi, M.; Moccari, M. Structure a family of three-step with-memory methods for solving nonlinear equations and their dynamics. *Math. Anal. Convex Optim.* **2021**, *2*, 119–137.
28. Sharma, E.; Panday, S.; Mittal, S.K.; Joit, D.M.; Pruteanu, L.L.; Jäntschi, L. Derivative-free families of with- and without-memory iterative methods for solving nonlinear equations and their engineering applications. *Mathematics* **2023**, *14*, 4512. [[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. Wu, X.Y. Newton-like method with some remarks. *Appl. Math. Comput.* **2007**, *118*, 433–439. [[CrossRef](#)]
31. Wang, H.; Liu, H. Note on a cubically convergent Newton-type method under weak conditions. *Acta Appl. Math.* **2010**, *110*, 725–735 [[CrossRef](#)]
32. 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](#)]
33. Argyros, I.K.; Chen, D.; Qian, Q. The Jarratt method in Banach space setting. *J. Comput. Appl. Math.* **1994**, *51*, 103–106. [[CrossRef](#)]
34. Liu, C.S. Computing the eigenvalues of the generalized Sturm-Liouville problems based on the Lie-group $SL(2, \mathbb{R})$. *J. Comput. Appl. Math.* **2012**, *236*, 4547–4560. [[CrossRef](#)]

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.