fractal and fractional

MDPI

Article

On Traub–Steffensen-Type Iteration Schemes With and Without Memory: Fractal Analysis Using Basins of Attraction

Moin-ud-Din Junjua [1,*], Shahid Abdullah [2], Munish Kansal [3] and Shabbir Ahmad [4,5,*]

1 School of Mathematical Sciences, Zhejiang Normal University, Jinhua 321004, China
2 School of Advanced Science and Languages, VIT Bhopal University, Kothri-Kalan, Sehore 466114, India; shahid.abdullah2021@vitbhopal.ac.in
3 Department of Mathematics, Thapar Institute of Engineering and Technology, Patiala 147004, India; munish.kansal@thapar.edu
4 School of Engineering, Universidade Federal do Rio Grande, Rio Grande 96201-900, Brazil
5 Institute of Geophysics and Geomatics, China University of Geosciences, Wuhan 321004, China
* Correspondence: moinuddin@zjnu.edu.cn (M.-u.-D.J.); shabbiraleem@cug.edu.cn (S.A.)

Abstract: This paper investigates the design and stability of Traub–Steffensen-type iteration schemes with and without memory for solving nonlinear equations. Steffensen's method overcomes the drawback of the derivative evaluation of Newton's scheme, but it has, in general, smaller sets of initial guesses that converge to the desired root. Despite this drawback of Steffensen's method, several researchers have developed higher-order iterative methods based on Steffensen's scheme. Traub introduced a free parameter in Steffensen's scheme to obtain the first parametric iteration method, which provides larger basins of attraction for specific values of the parameter. In this paper, we introduce a two-step derivative free fourth-order optimal iteration scheme based on Traub's method by employing three free parameters and a weight function. We further extend it into a two-step eighth-order iteration scheme by means of memory with the help of suitable approximations of the involved parameters using Newton's interpolation. The convergence analysis demonstrates that the proposed iteration scheme without memory has an order of convergence of 4, while its memory-based extension achieves an order of convergence of at least 7.993, attaining the efficiency index $7.993^{1/3} \approx 2$. Two special cases of the proposed iteration scheme are also presented. Notably, the proposed methods compete with any optimal j-point method without memory. We affirm the superiority of the proposed iteration schemes in terms of efficiency index, absolute error, computational order of convergence, basins of attraction, and CPU time using comparisons with several existing iterative methods of similar kinds across diverse nonlinear equations. In general, for the comparison of iterative schemes, the basins of iteration are investigated on simple polynomials of the form $z^n - 1$ in the complex plane. However, we investigate the stability and regions of convergence of the proposed iteration methods in comparison with some existing methods on a variety of nonlinear equations in terms of fractals of basins of attraction. The proposed iteration schemes generate the basins of attraction in less time with simple fractals and wider regions of convergence, confirming their stability and superiority in comparison with the existing methods.

Keywords: nonlinear equations; iteration methods with-memory; order of convergence; efficiency index; fractal analysis; basins of attraction

MSC: 65H05; 65D05; 65B99

check for updates

Citation: Junjua, M.-u.-D.; Abdullah, S.; Kansal, M.; Ahmad, S. On Traub–Steffensen-Type Iteration Schemes With and Without Memory: Fractal Analysis Using Basins of Attraction. *Fractal Fract.* **2024**, *8*, 698. https://doi.org/10.3390/fractalfract8120698

Academic Editors: Neus Garrido, Francisco I. Chicharro and Paula Triguero-Navarro

Received: 6 October 2024
Revised: 22 November 2024
Accepted: 23 November 2024
Published: 26 November 2024

1. Introduction

Several real-life problems in engineering and applied sciences involve nonlinear equations of the form $\phi(\omega) = 0$, where $\phi : I \in \mathbb{R} \to \mathbb{R}$ and I is an open interval. The solution of these nonlinear equations is the basic aim of this research, which has a simple zero, say α. Since the roots of a nonlinear equation cannot always be determined accurately, we have to find a numerical solution by using numerical methods. For this purpose, iteration methods, like Newton's method, are frequently used [1–3]. Traub [3] classified these iterative methods into two categories; one-point (one-step) iterative methods and multi-point (multi-step) iterative methods. Newton's method [1] and Steffensen's method [4] are famous examples of one-step, one-point iterative methods, given by (1) and (2), respectively.

$$\omega_{j+1} = \omega_j - \frac{\phi(\omega_j)}{\phi'(\omega_j)}, j \geq 0, \tag{1}$$

$$\begin{aligned} \chi_j &= \omega_j + \phi(\omega_j), j \geq 0, \\ \omega_{j+1} &= \omega_j - \frac{\phi(\omega_j)}{\phi[\omega_j, \chi_j]}, \end{aligned} \tag{2}$$

where $\phi[\omega_j, \chi_j] = \frac{\phi(\omega_j) - \phi(\chi_j)}{\omega_j - \chi_j}$.

The investigation of the dynamical behavior of iterative methods using basins of attraction provides information about the regions of convergence and the selection of initial guesses for which a method converges or fails to converge. To investigate the regions of convergence of an iteration scheme for solving a nonlinear equation $\phi(z) = 0$, we plot its basins of attraction, i.e., the set of initial guesses for which the iteration scheme converges to the roots [5,6], as follows. We chose an initial guess z_0 from a grid of 500×500 points within the rectangle $D \subset \mathbb{C}$, which contains all of the roots of $\phi(z) = 0$, each allocated by a unique color. Starting with an initial point in D, an iteration method may either converge to one of the roots or diverge after a specified number of iterations '20', usually marked with the color black. For more details regarding basins of attraction, one should refer to [6,7]. For instance, we plot the basins of attraction of $\phi(z) = z^3 - 1$ for Steffensen's method (2), which has three roots—1, $-0.5 - 0.866025\imath$, and $-0.5 + 0.866025\imath$—contained in $D = [-3, 3] \times [-3, 3]$ and represented by cyan, magenta, and yellow, respectively. Figure 1 represents the basins of attraction of $\phi(z) = z^3 - 1$ using Newton's method (1) and Steffensen's method (2) with 1335 and 226,616 black points, respectively, from a total of 251,001 points in the region.

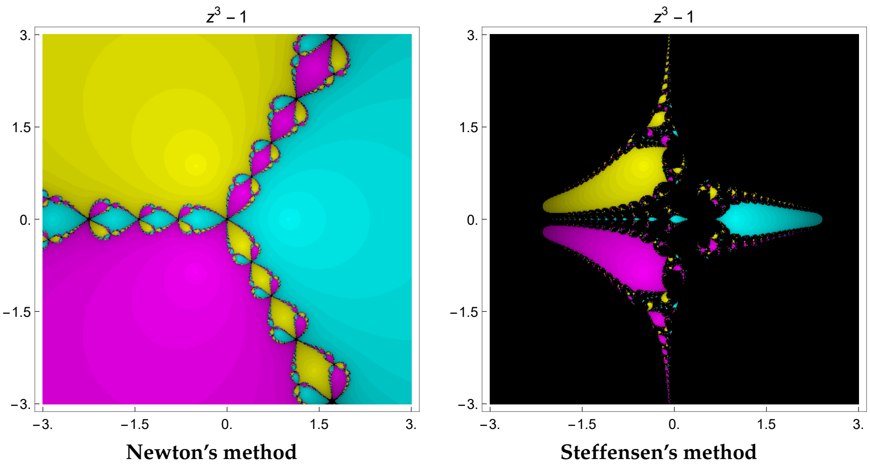

Figure 1. Basins of attraction of $\phi(z) = z^3 - 1$ using Newton's method (1) and Steffensen' method (2).

Steffensen's method overcomes the difficulty of derivative calculation in Newton's scheme, but, in general, it has smaller sets of initial guesses that converge to the desired roots (basins of attraction) as shown in Figure 2. In recent years, several researchers have developed higher-order variants of Steffensen's scheme despite this drawback of Steffensen's method. Traub [3] introduced a free parameter in Steffensen's scheme (2) to obtain the first parametric derivative free iteration method, which provides larger basins of attraction for specific values of the parameter. Traub [3] further presented an iteration method with memory by using a suitable approximation of the free parameter β_j as follows:

$$\begin{aligned} \chi_j &= \omega_j + \beta_j \phi(\omega_j),\ \beta_j \neq 0, \\ \omega_{j+1} &= \omega_j - \frac{\phi(\omega_j)}{\phi[\omega_j, \chi_j]},\ j \geq 0, \end{aligned} \tag{3}$$

where $\omega_0,\ \beta_0$ are given, $\phi[\omega,\chi] = \frac{\phi(\omega)-\phi(\chi)}{(\omega-\chi)}$ denotes the divided difference of first order and $\beta_j = \frac{-1}{N_1'(\omega_j)}, j \geq 1$, where $N_1 = \phi(\omega_j) + (\omega - \omega_j)\phi[\omega_j, \chi_j]$ is the first degree of Newton's interpolation polynomial. The iterative scheme with memory given by (3) has a convergence order of 2.41. Figure 2 represents the basins of attraction of $\phi(z) = z^3 - 1$ using Traub's method (3) for $\beta_0 = 0.01$ and $\beta_0 = 0.001$ with 5825 and 2177 black points, respectively.

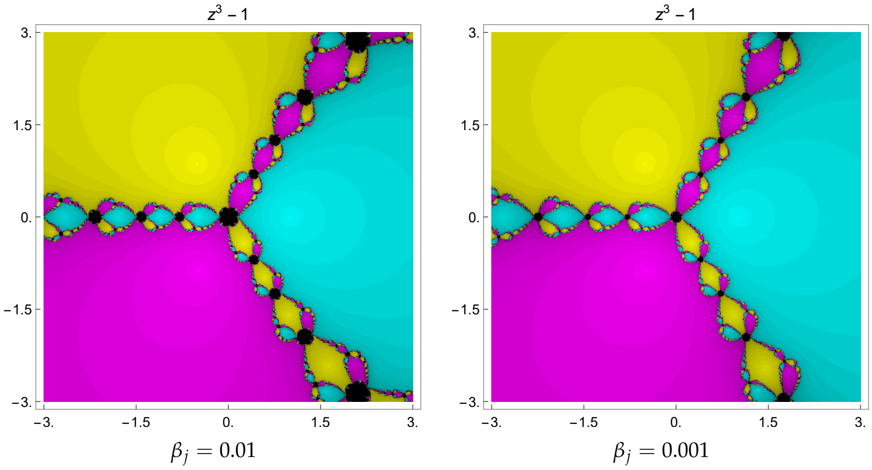

Figure 2. Basins of attraction of $\phi(z) = z^3 - 1$ using Traub's method (3).

The concept of an optimal root finding method was stated by Kung and Traub [8], that a multi-step iterative method without memory using $j+1$ function evaluations per iteration has an order of convergence of 2^j (optimal method). Ostrowski [2] defined the efficiency index, i.e., $EI = \rho^{1/j}$, to compute the efficiency of a root-finding iteration scheme, where ρ and j denote the order of convergence and the total number of function evaluations used by an iterative scheme per iterative step, respectively. For an optimal j-step iterative method (based on $j+1$ function evaluations) without memory, the efficiency index is $EI = \lim\limits_{j \to \infty} 2^{\frac{j}{\rho+1}} = 2$. For instance, the efficiency index of the two-step optimal fourth order King's method [9] is $4^{\frac{1}{3}} \simeq 1.587$ (and requires three functional evaluations).

Since multi-step (multi-point) methods have advantages over the one-step (one-point) iteration methods in terms of the efficiency index and convergence order, several optimal multi-step (multi-point) iteration methods without memory for solving nonlinear equations have been derived in recent years (see, for example, [10–18]).

Traub [3] pointed out that in some cases, the convergence order and efficiency index EI of an iteration scheme can be improved without using additional functional evaluations based on the approximation of an accelerating parameter, which appears in its error term by using an interpolating polynomial, which passes through the available points at current and previous iterations. Such iteration methods are defined as methods with memory [3]. Inspired by this idea, in recent years, several two- and three-step iterative methods with memory have been developed by employing free parameters [19–30]. Recently, Abdullah et al. [30] have developed a two-point iterative method with-memory by using Hermite interpolation polynomials in an existing sixth-order method without memory. They improved the $R-$order of convergence of a sixth-order method to 7.2749 and the efficiency index from 1.37 to 1.64. For more details regarding the improvement of convergence order by means of memory, one should see, e.g., [14,24].

In this paper, we present a family of two-step iterative root-finding methods with memory with a convergence order of $7.993 \simeq 8$ and an efficiency index of $7.993^{1/3} \simeq 2$, which is equal to an efficiency index of an $j-$point optimal method without memory of order 2^j. In addition, the proposed methods possess wider regions of convergence, illustrated in terms of basins of attraction. The remaining contents of the paper proceed as follows. In Section 2, based on Traub's scheme (3) and the second step of King's method [9] and by using a parametric approximation of a derivative along with a weight function, we obtain a new optimal fourth-order derivative-free iteration scheme. In Section 3, we employ three self-accelerating parameters in the new optimal fourth-order scheme such that the convergence order is improved from 4 to 8 without using additional functional evaluations (i.e., using only three functional evaluations). It is necessary to remark that the efficiency index of the fourth-order method is improved from 1.587 to 2. Section 4 is devoted to presenting some particular cases of the proposed family and weight functions. In Section 5, some numerical examples and real-life applications are reported to test the efficiency and performance of proposed methods and to justify the theoretical results. Section 6 presents an extensive analysis and comparison of proposed methods with existing ones in terms of fractals of basins of attractions in the complex plane on a variety of nonlinear functions. Finally, some concluding remarks are given in Section 7.

2. Two-Step Traub-Steffensen Type Iterative Scheme

In this section, we design a derivative-free two-step fourth-order optimal iteration scheme without memory. We introduce a free parameter q in Traub's method without memory and combine it with the second step of King's scheme [9] as follows:

$$\begin{aligned} \chi_j &= \omega_j + p\phi(\omega_j), p \neq 0,\ j \geq 0, \\ \omega_{j+1} &= \omega_j - \frac{\phi(\omega_j)}{\phi[\omega_j, \chi_j] + q\phi(\omega_j)}, \\ \omega_{j+1} &= z_j - \frac{\phi(z_j)}{\phi'(\omega_j)} \frac{\phi(\omega_j) + \lambda\phi(z_j)}{\phi(\omega_j) + (\lambda - 2)\phi(z_j)}, \lambda \in \mathbb{R}. \end{aligned} \tag{4}$$

By approximating $\phi'(\omega_j)$ with $\phi[z_j, \chi_j] + q\phi(\chi_j) + s(z_j - \chi_j)(z_j - \omega_j)$ in the second step of the scheme (4), the following derivative-free two-step iteration scheme is obtained:

$$\begin{aligned} z_j &= \omega_j - \frac{\phi(\omega_j)}{\phi[\omega_j, \chi_j] + q\phi(\chi_j)}, \chi_j = \omega_j + p\phi(\omega_j), j \geq 0, \\ \omega_{j+1} &= z_j - \frac{\phi(z_j)}{\phi[z_j, \chi_j] + q\phi(\chi_j) + s(z_j - \chi_j)(z_j - \omega_j)} \frac{\phi(\omega_j) + \lambda\phi(z_j)}{\phi(\omega_j) + (\lambda - 2)\phi(z_j)}, \end{aligned} \tag{5}$$

where the scalars $p \neq 0, q$ and s are free parameters. With the help of Taylor series expansions, one can obtain the following error equation for the iteration scheme (5):

$$e_{j+1} = -(c_2 + q)^2(1 + pc_1)^2 e_j^3,$$

where $e_j = \omega_j - \alpha$ (ω_j and α being approximate and exact roots, respectively) is the error at jth iteration and $c_k = \frac{\phi^k(\alpha)}{k!\phi'(\alpha)}$. Note that the scheme (5) is not optimal as it provides convergence order 3 by using three functional evaluations. To make it optimal, we use a real valued weight function $G(t_j)$ (where $t_j = \frac{\phi(z_j)}{\phi(\omega_j)}$) in (5) and achieve the following family of optimal fourth-order methods:

$$z_j = \omega_j - \frac{\phi(\omega_j)}{\phi[\omega_j, \chi_j] + q\phi(\chi_j)}, \chi_j = \omega_j + p\phi(\omega_j), j \geq 0,$$
$$\omega_{j+1} = z_j - G(t_j)\frac{\phi(z_j)}{\phi[z_j, \chi_j] + q\phi(\chi_j) + s(z_j - \chi_j)(z_j - \omega_j)} \frac{\phi(\omega_j) + \lambda\phi(z_j)}{\phi(\omega_j) + (\lambda - 2)\phi(z_j)}. \quad (6)$$

The subsequent theorem demonstrates the conditions on the weight function $G(t_j)$ to obtain optimal fourth-order convergence of the scheme (6).

Theorem 1. *Let $\alpha \in I$ be a simple root of a sufficiently differentiable nonlinear function ϕ such that $\phi : I \subseteq \mathbb{R} \to \mathbb{R}$, where $I \subseteq \mathbb{R}$ is an open set. Let an initial approximation ω_0 be close enough to α and if $G(0) = 1$, $G'(0) = -1$ and $G''(0) < \infty$, then the iteration scheme (6) has convergence order 4 with the error equation as follows:*

$$\begin{aligned} e_{j+1} = & -\frac{1}{2c_1}(1 + pc_1)^2(c_2 + q)(-4\lambda q^2 pc_1^2 + G''(0)q^2 pc_1^2 + 2q^2 pc_1^2 \\ & -8\lambda qc_2 pc_1^2 + 4qc_1^2 c_2 p + 2G''(0)qc_2 pc_1^2 - 4\lambda c_2^2 pc_1^2 + G''(0)c_2^2 pc_1^2 \\ & +2c_2^2 pc_1^2 - 4c_1\lambda q^2 + c_1 G''(0)q^2 + 2c_1 q^2 - 8c_1\lambda qc_2 + 2c_1 G''(0)qc_2 \\ & -4c_1\lambda c_2^2 + 2c_1 c_3 - 2c_1 c_2^2 + c_1 G''(0)c_2^2 - 2s)e_j^4, \end{aligned} \quad (7)$$

where $\lambda \in \mathbb{R}, p \neq 0, q$ and s are free parameters, $c_1 = \phi'(\alpha)$ and $c_k = \frac{\phi^{(k)}(\alpha)}{k!\phi'(\alpha)}, k \geq 2$.

Proof. Let the error at jth iteration be $e_j = \omega_j - \alpha$. By using Taylor's series expansions of the function ϕ in the jth iteration, the proof is similar to those given in [14,19,21]. Hence, it is omitted. □

Remark 1. *Theorem 1 demonstrated that convergence order of the iteration scheme (6) is 4 and it's efficiency index is $4^{\frac{1}{3}} \approx 1.587$.*

Remark 2. *If we chose $p = -\frac{1}{c_1}$ and $q = -c_2$, then error Equation (7) becomes*

$$e_{j+1} = \frac{-c_1^2 c_2^2 c_3^2 + sc_1 c_2^2 c_3}{c_1^2} e_j^7 + O(e_j^8). \quad (8)$$

Further, by choosing $s = c_1 c_3$, the obtained method has a convergence order of 8. Therefore, it is concluded from the error analysis that the free parameters p, q and s in (7) perform a significant role in the with-memorization of the method without memory (6). These parameters are called self-accelerating parameters. Hence, the scheme (6) is extendable to a novel method with memory with an accelerated order of convergence 8 and a very high-efficiency index 2.

3. Two-Step Tri-Parametric Iterative Scheme With-Memory

In this section, without requiring any additional functional evaluations, we extend the Traub–Steffensen type fourth-order tri-parametric iteration scheme (6) to an eighth-order iteration scheme with memory. To achieve this goal, we employ Newton's interpolation polynomials of an appropriate degree to recursively determine the self-accelerating parameters p, q, and s utilizing the already saved points at the current and previous iterations. We select the associated parameters p, q, and s in a manner that increases the fourth order of convergence of the scheme (6), as previously discussed.

If we choose $p = -\frac{1}{c_1} =, q = -c_2$ and $s = c_1 c_3$, the order of the scheme (6) increase up to eight. Since the root α and consequently the values of $\phi'(\alpha), \phi''(\alpha)$ and $\phi'''(\alpha)$ are not known, we approximate the self-accelerators p, q, and s in (6) recursively by using Newton's interpolation polynomials of an appropriate degree at each iterative step as:

$$\begin{aligned} p &= p_j = -\frac{1}{N_9'(\omega_j)} \approx -\frac{1}{\phi'(\alpha)}, \\ q &= q_j = -\frac{N_{10}''(\chi_j)}{2N_{10}'(\chi_j)} \approx -c_2, \\ s &= s_j = \frac{N_{11}'''(z_j)}{6} \approx c_1 c_3. \end{aligned} \tag{9}$$

where $N_9(\xi) = N_9\big(\xi, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}\big)$ is a ninth degree Newton's interpolation polynomial that passes through the points $\omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}$, for any $j \geq 3$, given by:

$$\begin{aligned} & N_9\big(\xi, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}\big) \\ = \; & \phi(\omega_j) + (\xi - \omega_j)\phi[\omega_j, z_{j-1}] + (\xi - \omega_j)(\xi - z_{j-1})\phi[\omega_j, z_{j-1}, \chi_{j-1}] \\ & + (\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})\phi[\omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}] \\ & + (\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})\phi[\omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}] \\ & + (\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2}) \\ & \phi[\omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}] \\ & + (\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2}) \\ & (\xi - \chi_{j-2})\phi[\omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}] \\ & + (\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2}) \\ & (\xi - \chi_{j-2})(\xi - \omega_{j-2})\phi[\omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}] \\ & + (\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2})(\xi - \chi_{j-2}) \\ & (\xi - \omega_{j-2})(\xi - z_{j-3})\phi[\omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}] \\ & + (\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2})(\xi - \chi_{j-2})(\xi - \omega_{j-2}) \\ & (\xi - z_{j-3})(\xi - \chi_{j-3})\phi[\omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}]. \end{aligned} \tag{10}$$

$N_{10}(\xi) = N_{10}(\xi, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3})$ is a tenth degree Newton's interpolation polynomial that passes through the points, $\chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}$, for any $j \geq 3$, given by:

$$\begin{aligned} & N_{10}(\xi, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}) \\ = \; & \phi(\chi_j) + (\xi - \chi_j)\phi[\chi_j, \omega_j] + (\xi - \chi_j)(\xi - \omega_j)\phi[\chi_j, \omega_j, z_{j-1}] \\ & + (\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})\phi[\chi_j, \omega_j, z_{j-1}, \chi_{j-1}] \\ & + (\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})\phi[\chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}] \\ & + (\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})\phi[\chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}] \\ & + (\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2}) \\ & \phi[\chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}] \\ & + (\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2})(\xi - \chi_{j-2}) \\ & \phi[\chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}] \\ & + (\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2})(\xi - \chi_{j-2})(\xi - \omega_{j-2}) \\ & \phi[\chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}] \\ & + (\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2})(\xi - \chi_{j-2})(\xi - \omega_{j-2}) \\ & (\xi - z_{j-3})\phi[\chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}] \\ & + (\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2})(\xi - \chi_{j-2})(\xi - \omega_{j-2}) \\ & (\xi - z_{j-3})(\xi - \chi_{j-3})\phi[\chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}]. \end{aligned} \tag{11}$$

$N_{11}(\xi) = N_{11}(\xi, z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3})$ is an eleventh degree Newton's interpolating polynomial that passes through the points, $z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}$, for any $j \geq 3$, given by:

$$
\begin{aligned}
& N_{11}(\xi, z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}) \\
= \; & \phi(z_j) + (\xi - z_j)\phi[z_j, \chi_j] + (\xi - z_j)(\xi - \chi_j)\phi[z_j, \chi_j, \omega_j] \\
& +(\xi - z_j)(\xi - \chi_j)(\xi - \omega_j)\phi[z_j, \chi_j, \omega_j, z_{j-1}] \\
& +(\xi - z_j)(\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})\phi[z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}] \\
& +(\xi - z_j)(\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})\phi[z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}] \\
& +(\xi - z_j)(\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1}) \\
& \phi[z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}] \\
& +(\xi - z_j)(\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2}) \\
& \phi[z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}] \\
& +(\xi - z_j)(\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2})(\xi - \chi_{j-2}) \\
& \phi[z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}] \\
& +(\xi - z_j)(\xi - \chi_j)(\xi - \omega_j)(\xi - v_{n-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2})(\xi - \chi_{j-2}) \\
& (\xi - \omega_{j-2})\phi[z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}] \\
& +(\xi - z_j)(\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2})(\xi - \chi_{j-2}) \\
& (\xi - \omega_{j-2})(\xi - z_{j-3})\phi[z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}] \\
& +(\xi - z_j)(\xi - \chi_j)(\xi - \omega_j)(\xi - z_{j-1})(\xi - \chi_{j-1})(\xi - \omega_{j-1})(\xi - z_{j-2})(\xi - \chi_{j-2})(\xi - \omega_{j-2}) \\
& (\xi - z_{j-3})(\xi - \chi_{j-3})\phi[z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}].
\end{aligned} \tag{12}
$$

Finally, we present the following two-step tri-parametric family of iterative methods with-memory, i.e., by replacing the parameters p, q, and s in the scheme (6) with self-accelerators p_j, q_j, and s_j, given in (9):

$$
\begin{aligned}
z_j &= \omega_j - \frac{\phi(\omega_j)}{\phi[\omega_j, \chi_j] + q_j\phi(\chi_j)}, \chi_j = \omega_j + p_j\phi(\omega_j), j \geq 0, \\
p_j &= -\frac{1}{N_9'(t_j)}, q_j = -\frac{N_{10}''(\chi_j)}{2N_{10}'(\chi_j)}, s_j = \frac{N_{11}'''(\chi_j)}{6}, \\
\omega_{j+1} &= z_j - G(t_j)\frac{\phi(z_j)}{\phi[z_j, \chi_j] + q_j\phi(\chi_j) + s_j(z_j - \chi_j)(z_j - \omega_j)}\frac{\phi(\omega_j) + \lambda\phi(z_j)}{\phi(\omega_j) + (\lambda - 2)\phi(z_j)}.
\end{aligned} \tag{13}
$$

It is worth mentioning that the initial values p_0, q_0, and s_0 could be taken as very small positive values. Additionally, the self-accelerator p_j is to be computed exactly before the start of each iteration, q_j is computed after χ_j, and s_j is computed after the computation of z_j.

The following theorem demonstrates that the newly presented iterative scheme with-memory (13) has a convergence order of 7.993 with a computational efficiency index of $1.999 \simeq 2$.

Theorem 2. *Let ω_0 be an initial guess near enough to the simple zero α of a sufficiently differentiable function ϕ. If self-accelerators p_j, q_j, and s_j are iteratively computed by using the formulae given in (9), then the $R-$order of convergence of the proposed iterative scheme with memory (13) is at least* 7.993 *with an efficiency index of* $1.999 \simeq 2$.

Proof. The R-order of convergence of the iteration method (13) is ascertained using the Herzberger's matrix method [31]. The lower bound of convergence order of one-step $m-$point method with memory

$$
\omega_j = \psi(\omega_{j-1}, \omega_{j-2}, \cdots, \omega_{l-m}),
$$

is the spectral radius of its associated matrix $Q^{(m)} = \left(l_{i,j}\right), 1 \le i, j, \le m$, with following elements:

$$\begin{aligned} l_{1,j} &= \text{number of functional evaluations evaluated at point } \omega_{l-j} \text{ where } j = 1, 2, \cdots, m, \\ l_{i,i-1} &= 1, \text{ for } i = 2, 3, \cdots, m, \\ l_{i,j} &= 0, \text{ elseways.} \end{aligned}$$

Whereas, the spectral radius of $Q_1 \cdot Q_2 \cdot \cdots \cdot Q_m$ is defined as the lower bound of the order of an $m-$step iterative method $\psi = \psi_1 \circ \psi_2 \circ \cdots \circ \psi_m$, where the matrices Q_t correspond to the iteration steps $\psi_t, 1 \le t \le m$.

According to the scheme (13), we obtain the corresponding matrices as follows:

$$\begin{aligned} \omega_{j+1} &= \psi_1\left(z_j, \chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, \omega_{j-2}, z_{j-2}, \chi_{j-2}, z_{j-3}, \chi_{j-1}, \omega_{j-3}\right) \\ &\to \quad Q_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} z_j &= \psi_2\left(\chi_j, \omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}, z_{j-4}\right) \\ &\to \quad Q_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} \chi_j &= \psi_1\left(\omega_j, z_{j-1}, \chi_{j-1}, \omega_{j-1}, z_{j-2}, \chi_{j-2}, \omega_{j-2}, z_{j-3}, \chi_{j-3}, \omega_{j-3}, z_{j-4}, \chi_{j-4}\right) \\ &\to \quad Q_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \end{aligned}$$

Hence, we obtain:

$$Q^{(3)} = Q_1 \cdot Q_2 \cdot Q_3 = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The above matrix $Q^{(3)}$ has the eigenvalues: $0,0,0,0,0,0,0,0,7.993145621,-0.687271071,$ $-0.152937275+0.8394933233i$, $-0.152937275-0.8394933233i$. As a result, the matrix's spectral radius $Q^{(3)}$ is $\rho\left(Q^{(3)}\right) = 7.993145621$. Hence, we conclude that the order of convergence of the proposed two-step iterative scheme with memory (13) is at least 7.993 with an efficiency index of $1.999 \simeq 2$. $\square$

4. Special Cases

One can obtain several special cases of iteration scheme with memory (13) by choosing the weight functions $G(t_j)$, such that the conditions of Theorem 1, i.e., $G(0) = 1$, $G'(0) = -1$, $G'(0) < \infty$ are satisfied. Here, we present two simple special cases of our iteration scheme (13) as follows.

Case 1: By choosing $G(t_j) = 1 - t_j$ (where $t_j = \dfrac{\phi(z_j)}{\phi(\omega_j)}$) in the scheme (13), we achieve the following specific method using the memory represented by $SM1$:

$$\begin{aligned} z_j &= \omega_j - \frac{\phi(\omega_j)}{\phi[\omega_j,\chi_j] + q_j\phi(\chi_j)}, \chi_j = \omega_j + p_j\phi(\omega_j), j \geq 0, \\ p_j &= -\frac{1}{N_9'(t_j)}, q_j = -\frac{N_{10}''(\chi_j)}{2N_{10}'(\chi_j)}, s_j = \frac{N_{11}'''(\chi_j)}{6}, \\ \omega_{j+1} &= z_j - \left(1 - \frac{\phi(z_j)}{\phi(\omega_j)}\right) \frac{\phi(z_j)}{\phi[z_j,\chi_j] + q_j\phi(\chi_j) + s_j(z_j-\chi_j)(z_j-\omega_j)} \\ &\times \frac{\phi(\omega_j) + \lambda\phi(z_j)}{\phi(\omega_j) + (\lambda-2)\phi(z_j)}. \end{aligned} \tag{14}$$

Case 2: By taking $G(t_j) = \dfrac{1}{1+t_j}$ (being $t_j = \dfrac{\phi(z_j)}{\phi(\omega_j)}$) in the scheme (13), we obtain another method with-memory denoted by $SM2$, given as follows:

$$\begin{aligned} z_j &= \omega_j - \frac{\phi(\omega_j)}{\phi[\omega_j,\chi_j] + q_j\phi(\chi_j)}, \chi_j = \omega_j + p_j\phi(\omega_j), j \geq 0, \\ p_j &= -\frac{1}{N_9'(t_j)}, q_j = -\frac{N_{10}''(\chi_j)}{2N_{10}'(\chi_j)}, s_j = \frac{N_{11}'''(\chi_j)}{6}, \\ \omega_{j+1} &= z_j - \frac{\phi(\omega_j)}{\phi(\omega_j) + \phi(z_j)} \frac{\phi(z_j)}{\phi[z_j,\chi_j] + q_j\phi(\chi_j) + s_j(z_j-\chi_j)(z_j-\omega_j)} \\ &\times \frac{\phi(\omega_j) + \lambda\phi(z_j)}{\phi(\omega_j) + (\lambda-2)\phi(z_j)}. \end{aligned} \tag{15}$$

5. Numerical Experiments and Applications

In this section, we test our two-step tri-parametric methods with-memory (14) and (15) denoted by $(SM1)$ and $(SM2)$, respectively, with the help of different nonlinear functions given by Examples 1–7. To avoid the loss of significant digits and to achieve high accuracy, we have used the arbitrary precision arithmetics with 1000 significant digits in the programming package Maple 18. The formula to compute the computational order of convergence (COC) is given as follows [32]:

$$COC \approx \frac{\log|\phi(\omega_{j+1})/\phi(\omega_j)|}{\log|\phi(\omega_j)/\phi(\omega_{j-1})|}. \tag{16}$$

For all the comparisons, we have chosen $\sigma_{1,j} = \sigma_{2,j} = \sigma_{3,j} = \delta_{1,j} = \delta_{2,j} = \delta_{3,j} = p_j = q_j = s_j = 0.01$ to start the iterations. We compare the accuracy and efficiency of our proposed iteration schemes $(SM1)$ for $\lambda = 2$ and $(SM2)$ for $\lambda = 1$ with the existing two-step methods with-memory of Abdullah et al. [30] denoted by (SH), Zafar et al. [27] denoted by (FZ), Zaka Ullaha et al. [28] denoted by (ZK), Wang et al. [33] denoted by (XW), Choubey et al. [19] denoted by (NC), and Choubey et al. [20] denoted by (JN), described as follows:

Method SH:

$$\begin{aligned} w_j &= \omega_j - \frac{\phi(\omega_j)}{\phi'(\omega_j) - L_j\phi(\omega_j)}, j \geq 0, \\ \omega_{j+1} &= w_j - \frac{2\phi(w_j)\phi'(w_j)}{2\phi'^2(w_j) - \phi(w_j)T_\phi(w_j)}, \end{aligned} \tag{17}$$

where

$$T_\phi(w_j) = \frac{1}{w_j - \omega_j}\left(\phi'(\omega_j) + 2\phi'(w_j) + 3\frac{\phi(\omega_j) - \phi(w_j)}{w_j - \omega_j}\right) \tag{18}$$

and

$$L_j = \frac{2\phi[\omega_j, \omega_j, w_{j-1}] - (2\phi[\omega_j, w_{j-1}, \omega_{j-1}, \omega_{j-1}](\omega_j - w_{j-1}) - 4\phi[\omega_j, \omega_j, w_{j-1}, \omega_{j-1}])}{2\phi'(\omega_j)}. \tag{19}$$

Method FZ:

$$\begin{aligned} \chi_j &= \omega_j + \sigma_{1,l}\phi(\omega_j), \sigma_{1,l} = -\frac{1}{N_3'(\omega_j)}, j \geq 0, \\ r_j &= \omega_j - \frac{\phi(\omega_j)}{\phi[\omega_j, \chi_j] + \sigma_{2,l}\phi(\chi_j)}, \sigma_{2,l} = -\frac{N_4''(\chi_j)}{2N_4'(\chi_j)}, \\ \omega_{j+1} &= r_j - \left(\frac{1}{1+k_j}\right)\frac{1}{\left(1 - \frac{\phi(r_j)}{\phi(\omega_j)}\right)^2}\frac{\phi(r_j)}{\phi[\chi_j, r_j] + \sigma_{2,l}\phi(\chi_j) + \sigma_{3,l}(r_j - \chi_j)(r_j - \omega_j)}, \\ k_j &= \frac{\phi(r_j)}{\phi(\omega_j)}, \sigma_{3,l} = \frac{1}{6}N_5'''(r_j), \end{aligned} \tag{20}$$

where, for $j \geq 1$,

$$\begin{aligned} N_3(\xi, \omega_j, r_{j-1}, \chi_{j-1}, \omega_{j-1}) = \phi(\omega_j) &+ (\xi - \omega_j)\phi[\omega_j, r_{j-1}] + (\xi - \omega_j)(\xi - r_{j-1})\phi[\omega_j, r_{j-1}, \chi_{j-1}] \\ &+ (\xi - \omega_j)(\xi - r_{j-1})(\xi - \chi_{j-1})\phi[\omega_j, r_{j-1}, \chi_{j-1}, \omega_{j-1}], \end{aligned}$$

$$
\begin{aligned}
N_4(\xi,\chi_j,\omega_j,r_{j-1},\chi_{j-1},\omega_{j-1}) = {} & \phi(\chi_j) + (\xi-\chi_j)\phi[\chi_j,\omega_j] + (\xi-\omega_j)(\xi-\chi_j)\phi[\chi_j,\omega_j,r_{j-1}] \\
& + (\xi-\chi_j)(\xi-\omega_j)(\xi-r_{j-1})\phi[\chi_j,\omega_j,r_{j-1},\chi_{j-1}] \\
& + (\xi-\chi_j)(\xi-r_{j-1})(\xi-\omega_j)(\xi-\chi_{j-1})\phi[\chi_j,\omega_j,r_{j-1},\chi_{j-1},\omega_{j-1}],
\end{aligned}
$$

and

$$
\begin{aligned}
N_5(\xi,r_j,\chi_j,\omega_j,r_{j-1},\chi_{j-1},\omega_{j-1}) = {} & \phi(r_j) + (\xi-r_j)\phi[r_j,\chi_j] + (\xi-\chi_j)(\xi-r_j)\phi[r_j,\chi_j,\omega_j] \\
& + (\xi-\omega_j)(\xi-\chi_j)(\xi-r_j)\phi[r_j,\chi_j,\omega_j,r_{j-1}] \\
& + (\xi-r_{j-1})(\xi-\omega_j)(\xi-\chi_j)(\xi-r_j)\phi[r_j,\chi_j,\omega_j,r_{j-1},\chi_{j-1}] \\
& + (\xi-\chi_{j-1})(\xi-r_{j-1})(\xi-\omega_j)(\xi-\chi_j)(\xi-r_j)\phi[r_j,\chi_j,\omega_j,r_{j-1},\chi_{j-1},\omega_{j-1}].
\end{aligned}
$$

Method ZK:

$$
\begin{aligned}
s_j &= \omega_j + \delta_{1,l}\phi(\omega_j), \delta_{1,l} = -\frac{1}{N_6'(\omega_j)}, j \geq 0, \\
d_j &= \omega_j - \frac{\phi(\omega_j)}{\phi[\omega_j,s_j] + \delta_{2,l}\phi(s_j)}, \delta_{2,l} = -\frac{N_7''(s_j)}{2N_7'(s_j)}, \\
\omega_{j+1} &= d_j - \frac{\phi(d_j)}{\phi[\omega_j,d_j] + \phi[s_j,\omega_j,d_j](d_j-\omega_j) + \delta_{3,l}(d_j-\omega_j)(d_j-s_j)}, \\
\delta_{3,l} &= \frac{N_8'''(d_j)}{6},
\end{aligned} \tag{21}
$$

where, for $j \geq 2$, $N_6(\xi)$ is a sixth degree interpolation polynomial passing through $\omega_j, d_{j-1}, \omega_{j-1}, s_{j-1}, d_{j-2}, s_{j-2}, \omega_{j-2}$, $N_7(\xi)$ is a seventh degree interpolation polynomial passing through $s_j, \omega_j, d_{j-1}, \omega_{j-1}, s_{j-1}, d_{j-2}, s_{j-2}, \omega_{j-2}$, and $N_8(\xi)$ is an eighth degree interpolation polynomial passing through $d_j, s_j, \omega_j, d_{j-1}, \omega_{j-1}, s_{j-1}, d_{j-2}, s_{j-2}, \omega_{j-2}$.

Method NC:

$$
\begin{aligned}
\chi_j &= \omega_j - \frac{f(\omega_j)}{f'(\omega_j) - L_j f(\omega_j)}, \\
\omega_{j+1} &= \chi_j - \frac{f(\chi_j)p_1(\omega_j,\chi_j)}{2p_1^2(\omega_j,\chi_j) - f(\chi_j)p_2(\omega_j,\chi_j)},
\end{aligned} \tag{22}
$$

where, $p_1(\omega_j,\chi_j) = 2\left(\frac{\phi(\chi_j)-\phi(\omega_j)}{\chi_j-\omega_j}\right) - \phi'(\omega_j)$, $p_2(\omega_j,\chi_j) = \frac{2}{\chi_j-\omega_j}\left(\frac{\phi(\chi_j)-\phi(\omega_j)}{\chi_j-\omega_j} - \phi'(\omega_j)\right)$.

Method JN:

$$
\begin{aligned}
\chi_j &= \omega_j - \frac{f(\omega_j)}{f'(\omega_j) - L_j f(\omega_j)} \\
\omega_{j+1} &= \chi_j - \frac{2f(\omega_j)f'(\chi_j)f(\chi_j)}{2f(\omega_j)f'(\chi_j)^2 - f'(\omega_j)^2 f(\chi_j) + f'(\omega_j)f'(\chi_j)f(\chi_j)},
\end{aligned} \tag{23}
$$

For methods with-memory SH, NC and JN, the values associated with the parameter L_j have been recorded as:

Formula 1:

$$
L_j = \frac{H_2''(\omega_j)}{2\phi'(\omega_j)}, \tag{24}
$$

where $H_2(\omega) = \phi(\omega_j) + \phi[\omega_j,\omega_j](\omega-\omega_j) + \phi[\omega_j,\omega_j,\chi_{j-1}](\omega-\omega_j)^2$ and $H_2''(\omega_j) = 2\phi[\omega_j,\omega_j,\chi_{j-1}]$.

Formula 2:

$$L_j = \frac{H_3''(\omega_j)}{2\phi'(\omega_j)}, \tag{25}$$

where $H_3(\omega) = H_2(\omega) + \phi[\omega_j, \omega_j, \chi_{j-1}, \omega_{j-1}](\omega - \omega_j)^2(\omega - \chi_{j-1})$ and $H_3''(\omega_j) = 2\phi[\omega_j, \omega_j, \chi_{j-1}, \omega_{j-1}](\omega_j - \chi_{j-1}) + 2\phi[\omega_j, \omega_j, \chi_{j-1}]$.

Formula 3:

$$L_j = \frac{H_4''(\omega_j)}{2\phi'(\omega_j)} \tag{26}$$

where $H_4(\omega) = H_3(\omega) + \phi[\omega_j, \omega_j, \chi_{j-1}, \omega_{j-1}, \omega_{j-1}](\omega - \omega_j)^2(\omega - \chi_{j-1})(\omega - \omega_{j-1})$ and $H_4''(\omega_j) = 2\phi[\omega_j, \omega_j, \chi_{j-1}] - (2\phi[\omega_j, \chi_{j-1}, \omega_{j-1}, \omega_{j-1}](\omega_n - \chi_{j-1}) - 4\phi[\omega_j, \omega_j, \chi_{j-1}, \omega_{j-1}])$.

Method XW:

$$\begin{aligned} \chi_j &= \omega_j - \frac{\phi(\omega_j)}{\phi'(\omega_j) + L_j\phi(\omega_j)} \\ r_j &= \chi_j - \frac{\phi(\omega_j)}{2\phi[\omega_j, \chi_j] - \phi'(\omega_j) + L_j\phi(\chi_j)} \\ \omega_{j+1} &= r_j - [1 + \frac{3}{2}(a_j - b_j^3)]\frac{(\beta + \xi)\phi(r_j)}{2\xi\phi[\chi_j, r_j] + (\beta - \xi)(\phi'(\omega_j) + T\phi(r_j))} \end{aligned} \tag{27}$$

where $a_j = \frac{\phi(r_j)}{\phi(\omega_j)}, b_j = \frac{\phi(\chi_j)}{\phi(\omega_j)}, \beta = \chi_j - \omega_j, \xi = r_j - \omega_j$ and $T \in R$.

For the method with-memory XW, the parameter value L_j is recorded as:

Formula 1:

$$L_j = \frac{H_2''(\omega_j)}{2\phi'(\omega_j)}, \tag{28}$$

where $H_2(\omega) = \phi(\omega_j) + \phi[\omega_j, \omega_j](\omega - \omega_j) + \phi[\omega_j, \omega_j, r_{j-1}](\omega - \omega_j)^2$ and $H_2''(\omega_j) = 2\phi[\omega_j, \omega_j, r_{j-1}]$.

Formula 2:

$$L_j = \frac{H_3''(\omega_j)}{2\phi'(\omega_j)}, \tag{29}$$

where $H_3(\omega) = H_2(\omega) + \phi[\omega_j, \omega_j, r_{j-1}, \chi_{j-1}](\omega - \omega_n)^2(\omega - r_{j-1})$ and $H_3''(\omega_j) = 2\phi[\omega_n, \omega_j, r_{j-1}] + 2\phi[\omega_j, \omega_j, r_{j-1}, \chi_{j-1}](\omega_j - r_{j-1})$.

Formula 3:

$$L_j = \frac{H_4''(\omega_j)}{2\phi'(\omega_j)}, \tag{30}$$

where $H_4(\omega) = H_3(\omega) + \phi[\omega_j, \omega_j, r_{j-1}, \omega_{j-1}](\omega - \omega_j)^2(\omega - r_{j-1}) + \phi[\omega_j, \omega_j, r_{j-1}, \omega_{j-1}, \omega_{j-1}](\omega - \omega_j)^2(\omega - r_{j-1})(\omega - \chi_{j-1})$ and $H_4''(\omega_j) = H_2''(\omega_j) + 2\phi[\omega_j, \omega_j, r_{j-1}, \chi_{j-1}](\omega_n - r_{j-1}) + 2\phi[\omega_n, \omega_n, r_{j-1}, \chi_{j-1}, \omega_{j-1}](\omega_j - r_{j-1})(\omega_j - \chi_{j-1})]$.

Formula 4:

$$L_j = \frac{H_5''(\omega_j)}{2\phi'(\omega_j)}, \tag{31}$$

where $H_5(\omega) = H_4(\omega) + \phi[\omega_j, \omega_j, r_{j-1}, \chi_{j-1}, \omega_{j-1}, \omega_{j-1}](\omega - \omega_j)^2(\omega - r_{j-1})(\omega - \omega_{j-1})(\omega - \omega_{j-1})$ and $H_5''(\omega_j) = H_4''(\omega_j) + 2\phi[\omega_j, \omega_j, r_{j-1}, \chi_{j-1}, \omega_{j-1}, \omega_{j-1}](\omega_j - r_{j-1})(\omega_n - \chi_{j-1})(\omega_n - \omega_{j-1})$.

Example 1. *Location of maximum energy distribution:*

Planck's radiation law is given by

$$\alpha = \frac{8\pi ku\delta^{-5}}{e^{ku/\delta p\phi} - 1}, \tag{32}$$

where α is energy density, δ is wavelength radiation, ϕ is absolute temperature, u is Plank's constant, p is Boltzmann's constant, and k is the speed of light. To maximize the energy density and determine the wavelength, we first evaluate

$$\frac{d\alpha}{d\delta} = \frac{8\pi ku\delta^{-5}}{e^{ku/\delta p\phi} - 1}\left(-5 + \frac{ku/\delta p\phi e^{ku/\delta p\phi}}{e^{ku/\delta p\phi} - 1}\right). \tag{33}$$

The terms on the left side of the parentheses are zero in the limits as $\delta \to 0$, and $\delta \to \infty$, although energy density gives minima in both cases. The maximum we are seeking arises when the terms inside the parentheses are zero. This happens when

$$1 - \frac{ku}{5\delta_{\max} p\phi} = e^{-ku/\delta_{\max} p\phi}, \tag{34}$$

where $\delta_{\max}$ being the wavelength to maximize the energy density. For $\omega = ku/\delta_{\max} p\phi$, the above equation reduces to

$$1 - \frac{\omega}{5} = e^{-\omega}. \tag{35}$$

Now we can define the following non-linear expression,

$$\phi_1(\omega) = e^{-\omega} - 1 + \frac{\omega}{5}. \tag{36}$$

The problem is to solve the nonlinear equation (36), which has two roots, 4.965114232 and 0. We take the exact root $\alpha = 0$ and initial approximation $\omega_0 = -2.5$. The computational results are depicted in Table 1, where, $4.35\ (-1)$ denotes 4.35×10^{-1}. It is observed that the accuracy, computational order of convergence (COC), and efficiency index (EI) of our proposed schemes $SM1$ and $SM2$ are better than the others for the test problem $\phi_1(\omega)$.

Example 2. *Vertical stress:*

Boussinesq's formula computes the vertical stress (s) within an elastic material induced at a specific point beneath the edge of a rectangular strip footing subjected to a uniform pressure q given as follows:

$$\sigma_s = \frac{q}{\pi}\omega + Cos(\omega)Sin(\omega). \tag{37}$$

To determine the value of ω, where the vertical stress (s) is equal to 25 percent of the applied footing stress q, we have to find the value of z at first. To find the point at which the footing stress q is equal to a quarter, we have to solve the following equation:

$$\phi_2(\omega) = \frac{\omega + Cos(\omega)Sin(\omega)}{\pi} - \frac{1}{4}. \tag{38}$$

The exact root of Equation (38) is 0.415856.... We take an initial guess for this root as $\omega = 1.1$ to obtain the numerical results shown in Table 2.

Example 3. *We take the standard nonlinear test equation as follows:*

$$\phi_3(\omega) = e^{\omega^2 - 3\omega} sin(\omega) + ln(\omega^2 + 1). \tag{39}$$

For the above non-linear function, we take $\alpha = 0$ as the exact root and $\omega_0 = -0.5$ as an initial guess. The computational results are shown in Table 3, which illustrate that our proposed schemes $SM1$ and $SM2$ perform better in terms of convergence speed and efficiency.

Example 4. *We consider another standard nonlinear test equation as follows:*

$$\phi_4(\omega) = \frac{1}{\omega^4} - \omega^2 - \frac{1}{\omega} + 1. \tag{40}$$

Here, we take the exact root $\alpha = 1$ *and initial approximation* $\omega_0 = 0.2$*. The numerical results for comparison are illustrated in Table 4, which show that computational order of convergence (COC) and efficiency index (EI) of proposed schemes* $SM1$ *and* $SM2$ *are better than the earlier known schemes* SH*,* NC*,* JN*,* XW*,* FZ *and* ZK*.*

Example 5. *In addition, we pick another standard non-linear test problem, including trigonometric function:*

$$\phi_5(\omega) = (\omega - 2\tan(\omega))\left(\omega^3 - 8\right). \tag{41}$$

The above equation has 3 *real roots* $0, 2$ *and* -4.274782271*. We take* $\alpha = 0$ *as the exact root and* $\omega_0 = -1.5$ *as an initial guess for this problem. The computational results of the function* $\phi_5(\omega)$ *are shown in Table 5, from which it is seen that the proposed iterative schemes* $SM1$ *and* $SM2$ *have a faster convergence speed and better efficiency than the iterative schemes* SH*,* NC*,* JN*,* XW*,* FZ *and* ZK*.*

Example 6. *We take one more standard nonlinear equation, as follows:*

$$\phi_6(\omega) = \omega^3 + \omega^2 - 3\omega - 3. \tag{42}$$

Here, we take $\alpha = 1.732050807...$ *as an exact root. The comparison results by taking the initial guess* $\omega_0 = 3.5$ *are shown in Table 6. It is observed from Table 6 that the schemes* $SM1$ *and* $SM2$ *perform better than the existing schemes* FZ *and* ZK *in terms of convergence and efficiency.*

Example 7. *Blood rheology model:*

Blood rheology refers to the study of how blood flows and behaves in the circulatory system. Modeling blood rheology is important for understanding various physiological and pathological conditions related to blood flow. Numerical iterative methods are commonly used to solve the mathematical equations governing blood rheology. Blood rheology is a branch of medicine that focuses on the physical and flow properties of blood. Since blood is a non-Newtonian fluid, it is categorized as a Caisson fluid. According to this concept, flow in a tube behaves like a plug with little deformation, and a velocity gradient develops close to the wall. We take into account the following nonlinear equation.

$$\phi_7(\omega) = \frac{\omega^8}{441} - \frac{8}{63}\omega^5 - \frac{5714285714}{100000000000}\omega^4 + \frac{16}{9}\omega^2 - \frac{3624489796}{1000000000}\omega + \frac{36}{100}. \tag{43}$$

In order to examine the plug flow of Caisson fluid flow. Here, ω *shows the plug flow of Caisson fluid flow. The one of the solutions of* $\phi_7(\omega)$ *is* $0.1046986515....$ *We choose* $\omega_0 = -2.5$ *as an initial approximation to solve* $\phi_7(\omega) = 0$*. Table 7 display the calculated results.*

It is obvious from Tables 1–7 that the special cases $SM1$ and $SM2$ of our proposed iterative scheme are reliable and efficient than the earlier iterative methods SH, NC, JN, XW, FZ and ZK in terms of accuracy, computational order of convergence (COC) and efficiency index (EI) for different test problems.

Furthermore, Figure 3 demonstrates the graphical comparison of proposed iterative techniques $SM1$ and $SM2$ with other methods in terms of absolute error $|\omega_k - \alpha|$ while Figure 4 shows the comparison in terms of computational order of convergence (COC), efficiency index (EI), and CPU time, in the first three iterations for solving $\phi_1(\omega)$–$\phi_7(\omega)$. From Figures 3 and 4, it is observed that the proposed schemes $SM1$ and $SM2$ are more robust than the others.

Table 1. Numerical comparison of several iteration schemes with memory for $\phi_1(\omega)$.

Methods	$\lvert\omega_1 - \alpha\rvert$	$\lvert\omega_2 - \alpha\rvert$	$\lvert\omega_3 - \alpha\rvert$	$f(\omega_3)$	*COC*	*EI*	*CPU*
SH (17)–(24), $L = 0.1$	4.35 (−1)	1.64 (−5)	1.54 (−34)	1.23 (−34)	6.38	1.58	0.828
SH (17)–(25), $L = 0.1$	4.35 (−1)	5.11 (−5)	1.69 (−33)	1.35 (−33)	7.03	1.62	0.938
SH (17)–(26), $L = 0.1$	4.35 (−1)	1.85 (−6)	2.05 (−45)	1.64 (−45)	7.09	1.64	0.828
NC (22)–(24), $L = 0.1$	*D*	*D*	*D*	*D*	-	-	-
NC (22)–(25), $L = 0.1$	*D*	*D*	*D*	*D*	-	-	-
NC (22)–(26), $L = 0.1$	*D*	*D*	*D*	*D*	-	-	-
JN (23)–(24), $L = 0.1$	1.59 (−1)	3.99 (−5)	8.11 (−28)	8.11 (−28)	6.37	1.58	0.781
JN (23)–(25), $L = 0.1$	1.59 (−1)	1.91 (−5)	4.49 (−32)	3.59 (−32)	6.86	1.61	0.718
JN (23)–(26), $L = 0.1$	1.59 (−3)	2.34 (−5)	2.78 (−34)	2.23 (−34)	7.63	1.66	0.750
XW (27)–(28), $L = 0.1$, $T = 2$	8.68 (−1)	1.02 (−3)	1.71 (−23)	1.37 (−23)	6.23	1.57	0.812
XW (27)–(29), $L = 0.1$, $T = 2$	8.68 (−1)	1.24 (−3)	6.84 (−23)	5.47 (−23)	6.24	1.58	0.859
XW (27)–(30), $L = 0.1$, $T = 2$	8.68 (−1)	6.65 (−4)	4.70 (−25)	3.76 (−25)	6.30	1.59	0.844
XW (27)–(31), $L = 0.1$, $T = 2$	8.68 (−1)	8.95 (−4)	5.04 (−24)	4.03 (−24)	6.27	1.58	0.86
FZ (20)	3.27 (−4)	3.28 (−29)	4.61 (−220)	3.69 (−220)	7.63	1.96	0.446
ZK (21)	2.79 (−4)	2.38 (−30)	4.64 (−226)	3.45 (−226)	7.64	1.97	0.45
*SM*1 (14)	2.68 (−5)	3.34 (−26)	2.68 (−206)	2.15 (−206)	8.61	2.03	0.467
*SM*2 (15)	2.78 (−6)	5.75 (−33)	2.03 (−260)	1.62 (−260)	8.52	2.03	0.456

D stands for fails to converge.

Table 2. Numerical comparison of several iteration schemes with-memory for $\phi_2(\omega)$.

Methods	$\lvert\omega_1 - \alpha\rvert$	$\lvert\omega_2 - \alpha\rvert$	$\lvert\omega_3 - \alpha\rvert$	$f(\omega_3)$	*COC*	*EI*	*CPU*
SH (17)–(24), $L = -0.5$	1.90 (−3)	2.96 (−20)	5.22 (−133)	2.78 (−133)	6.70	1.60	0.969
SH (17)–(25), $L = -0.5$	1.90 (−3)	4.64 (−21)	1.10 (−144)	5.89 (−145)	7.01	1.62	0.875
SH (17)–(26), $L = -0.5$	1.90 (−3)	1.17 (−22)	3.18 (−162)	1.64 (−162)	7.26	1.64	0.89
NC (22)–(24), $L = -0.5$	1.05 (−2)	2.61 (−10)	2.87 (−45)	1.53 (−45)	4.59	1.66	0.766
NC (22)–(25), $L = -0.5$	1.05 (−2)	1.06 (−10)	1.83 (−49)	9.78 (−50)	4.84	1.69	0.813
NC (22)–(25), $L = -0.5$	1.05 (−2)	5.55 (−12)	3.73 (−58)	1.99 (−58)	4.97	1.70	0.844
JN (23)–(24), $L = -0.5$	2.16 (−3)	6.44 (−17)	2.04 (−96)	1.09 (−96)	5.87	1.55	0.890
JN (23)–(25), $L = -0.5$	2.16 (−3)	8.38 (−18)	1.52 (−107)	8.10 (−108)	6.22	1.57	0.891
JN (23)–(26), $L = -0.5$	2.16 (−3)	4.75 (−20)	3.36 (−127)	1.79 (−127)	6.43	1.60	0.829
XW (27)–(28), $L = 0.5$, $T = 2$	2.69 (−3)	4.44 (−21)	2.37 (−163)	1.26 (−163)	8.00	1.68	0.876
XW (27)–(29), $L = 0.5$, $T = 2$	2.69 (−3)	4.53 (−21)	2.77 (−163)	1.47 (−163)	8.00	1.68	0.859
XW (27)–(30), $L = 0.5$, $T = 2$	2.69 (−3)	4.52 (−21)	2.71 (−163)	1.44 (−163)	8.00	1.68	0.907
XW (27)–(31), $L = 0.5$, $T = 2$	2.69 (−3)	4.52 (−21)	2.71 (−163)	4.03 (−24)	8.00	1.68	0.875
FZ (20)	7.23 (−3)	2.65 (−17)	3.05 (−129)	1.62 (−129)	7.75	1.97	0.41
ZK (21)	8.43 (−3)	8.80 (−20)	6.64 (−158)	5.59 (−158)	7.90	1.99	0.610
*SM*1 (14)	3.30 (−5)	8.82 (−26)	1.57 (−144)	8.38 (−145)	7.77	2.00	0.608
*SM*2 (15)	7.80 (−10)	1.93 (−51)	3.65 (−247)	1.94 (−247)	8.00	2.00	0.588

Table 3. Numerical comparison of several iteration schemes with memory for $\phi_3(\omega)$.

Methods	$\|\omega_1 - \alpha\|$	$\|\omega_2 - \alpha\|$	$\|\omega_3 - \alpha\|$	$f(\omega_3)$	COC	EI	CPU
SH (17)–(24), $L = 0.5$	9.40 (-2)	8.94 (-5)	5.36 (-26)	5.36 (-26)	6.81	1.61	0.859
SH (17)–(25), $L = 0.5$	9.40 (-2)	7.74 (-6)	1.77 (-34)	1.77 (-34)	6.85	1.61	0.859
SH (17)–(26), $L = 0.5$	9.40 (-2)	9.65 (-6)	3.04 (-36)	3.04 (-36)	7.47	1.65	0.844
NC (22)–(24), $L = 0.5$	D		D	D	–	–	–
NC (22)–(25), $L = 0.5$	D		D	D	–	–	–
NC (22)–(26), $L = 0.5$	D		D	D	–	–	–
JN (23)–(24), $L = 0.5$	7.27 (-2)	2.00 (-4)	2.43 (-20)	2.43 (-20)	6.35	1.55	0.750
JN (23)–(25), $L = 0.5$	7.27 (-2)	2.38 (-4)	5.49 (-21)	5.49 (-21)	6.84	1.61	0.906
JN (23)–(26), $L = 0.5$	7.27 (-2)	2.57 (-4)	8.75 (-25)	8.75 (-25)	8.54	1.70	0.953
XW (27)–(28), $L = 0.5$, $T = 2$	1.38 (-1)	7.04 (-4)	2.28 (-21)	2.28 (-21)	7.15	1.63	0.922
XW (27)–(28), $L = 0.5$, $T = 2$	1.38 (-1)	8.47 (-4)	8.88 (-21)	8.88 (-21)	7.18	1.63	0.923
XW (27)–(30), $L = 0.5$, $T = 2$	1.38 (-1)	7.96 (-4)	5.48 (-21)	5.48 (-21)	7.18	1.63	0.875
XW (27)–(31), $L = 0.5$, $T = 2$	1.38 (-1)	8.22 (-4)	7.06 (-21)	7.06 (-21)	7.18	1.63	0.975
FZ (20)	1.90 (-3)	6.44 (-20)	1.17 (-142)	1.17 (-142)	7.45	1.95	0.496
ZK (21)	2.02 (-4)	4.93 (-22)	3.15 (-166)	1.32 (-166)	7.74	1.97	0.50
$SM1$ (14)	9.56 (-5)	1.48 (-24)	6.98 (-189)	6.98 (-189)	8.73	2.01	0.696
$SM2$ (15)	3.74 (-7)	4.32 (-34)	3.61 (-265)	3.61 (-265)	8.57	2.04	0.738

D stands for fails to converge.

Table 4. Numerical comparison of several iteration schemes with memory for $\phi_4(\omega)$.

Methods	$\|\omega_1 - \alpha\|$	$\|\omega_2 - \alpha\|$	$\|\omega_3 - \alpha\|$	$f(\omega_3)$	COC	EI	CPU
SH (17)–(24), $L = 0.1$	D	D	D	D	–	–	–
SH (17)–(25), $L = 0.1$	D	D	D	D	–	–	–
SH (17)–(26), $L = 0.1$	D	D	D	D	–	–	–
NC (22)–(24), $L = 0.1$	D	D	D	D	–	–	–
NC (22)–(25), $L = 0.1$	D	D	D	D	–	–	–
NC (22)–(26), $L = 0.1$	D	D	D	D	–	–	–
JN (23)–(24), $L = 0.1$	D	D	D	D	–	–	–
JN (23)–(25), $L = 0.1$	D	D	D	D	–	–	–
JN (23)–(26), $L = 0.1$	D	D	D	D	–	–	–
XW (27)–(28), $L = 0.1$, $T = 2$	D	D	D	D	–	–	–
XW (27)–(29), $L = 0.1$, $T = 2$	D	D	D	D	–	–	–
XW (27)–(30), $L = 0.1$, $T = 2$	D	D	D	D	–	–	–
XW (27)–(31), $L = 0.1$, $T = 2$	D	D	D	D	–	–	–
FZ (20)	7.50(-1)	1.92 (-2)	1.31 (-5)	6.57 (-5)	0.93	0.97	0.360
ZK (21)	2.30 (-4)	2.80 (-18)	4.25 (-144)	3.53 (-144)	7.94	1.99	0.341
$SM1$ (14)	3.72 (-4)	7.09 (-20)	3.27 (-152)	1.63 (-151)	8.25	2.01	0.452
$SM2$ (15)	3.65 (-5)	1.33 (-27)	5.13 (-214)	2.56 (-213)	8.30	2.09	0.488

D stands for fails to converge.

Table 5. Numerical comparison of several iteration schemes with memory for $\phi_5(\omega)$.

Methods	$\lvert\omega_1-\alpha\rvert$	$\lvert\omega_2-\alpha\rvert$	$\lvert\omega_3-\alpha\rvert$	$f(\omega_3)$	*COC*	*EI*	*CPU*
SH (17)–(17), $L=-0.2$	*D*	*D*	*D*	*D*	–	–	–
SH (17)–(17), $L=-0.2$	*D*	*D*	*D*	*D*	–	–	–
SH (17)–(17), $L=-0.2$	*D*	*D*	*D*	*D*	–	–	–
NC (22)–(24), $L=-0.2$	*D*	*D*	*D*	*D*	–	–	–
NC (22)–(25), $L=-0.2$	*D*	*D*	*D*	*D*	–	–	–
NC (22)–(26), $L=-0.2$	*D*	*D*	*D*	*D*	–	–	–
JN (23)–(24), $L=-0.2$	*D*	*D*	*D*	*D*	–	–	–
JN (23)–(25), $L=-0.2$	*D*	*D*	*D*	*D*	–	–	–
JN (23)–(26),$L=-0.2$	*D*	*D*	*D*	*D*	–	–	–
XW (27)–(28), $L=-0.2$, $T=5$	8.79 (−1)	7.95 (−2)	2.53 (−11)	2.02 (−10)	7.19	1.63	0.735
XW (27)–(29), $L=-0.2$, $T=5$	8.79 (−1)	8.62 (−2)	5.15 (−12)	4.12 (−11)	7.96	1.68	0.829
XW (27)–(30), $L=-0.2$, $T=5$	8.79 (−1)	2.66 (−2)	1.34 (−15)	1.07 (−14)	7.40	1.64	0.733
XW (27)–(31), $L=-0.2$, $T=5$	8.79 (−1)	3.06 (−1)	4.66 (−6)	3.73 (−5)	6.86	1.61	0.72
FZ (20)	1.23(−1)	4.60 (−9)	9.47 (−67)	7.57 (−66)	7.76	1.98	0.52
ZK (21)	3.17 (−4)	3.32 (−16)	2.54 (−168)	2.15 (−167)	7.92	1.99	0.601
*SM*1 (14)	2.54 (−4)	7.28 (−19)	1.56 (−183)	1.25 (−182)	10.12	2.16	0.52
*SM*2 (15)	4.08 (−5)	1.73 (−26)	9.31 (−260)	7.44 (−259)	10.91	2.21	0.606

D stands for fails to converge.

Table 6. Numerical comparison of several iteration schemes with memory for $\phi_6(\omega)$.

Methods	$\lvert\omega_1-\alpha\rvert$	$\lvert\omega_2-\alpha\rvert$	$\lvert\omega_3-\alpha\rvert$	$f(\omega_3)$	*COC*	*EI*	*CPU*
SH (17)–(24), $L=0.1$	3.86 (−2)	1.17 (−13)	3.56 (−94)	3.37 (−93)	6.98	1.62	0.593
SH (17)–(25), $L=0.1$	3.86 (−2)	1.80 (−15)	2.24 (−135)	2.12 (−134)	8.98	1.62	0.562
SH (17)–(26), $L=0.1$	3.86 (−2)	1.80 (−15)	2.24 (−135)	2.12 (−134)	8.98	1.73	0.609
NC (22)–(24), $L=0.1$	9.66 (−2)	1.12 (−6)	2.03 (−29)	1.92 (−28)	4.63	1.66	0.672
NC (22)–(25), $L=0.1$	9.66 (−2)	3.28 (−7)	1.29 (−34)	1.22 (−33)	5.03	1.71	0.672
NC (22)–(26), $L=0.1$	9.66 (−2)	3.28 (−7)	1.29 (−34)	1.22 (−33)	5.03	1.71	0.578
JN (23)–(24), $L=0.1$	2.00 (−2)	1.24 (−17)	1.87 (−66)	1.77 (−65)	5.95	1.56	0.626
JN (23)–(25), $L=0.1$	2.00 (−2)	8.62 (−14)	2.17 (−93)	2.05 (−92)	7.00	1.62	0.625
JN (23)–(26), $L=0.1$	2.00 (−2)	8.62 (−14)	2.17 (−93)	2.05 (−92)	7.00	1.62	0.673
XW (27)–(28), $L=0.1$, $T=2$	3.85 (−2)	1.73 (−10)	1.50 (−77)	1.42 (−76)	8.04	1.68	0.625
XW (27)–(29), $L=0.1$, $T=2$	3.85 (−2)	1.46 (−10)	3.86 (−78)	3.65 (−77)	8.03	1.68	0.594
XW (27)–(30), $L=0.1$, $T=2$	3.85 (−2)	1.46 (−10)	3.86 (−78)	3.65 (−77)	8.03	1.68	0.578
XW (27)–(31), $L=0.1$, $T=2$	3.85 (−2)	1.46 (−10)	3.86 (−78)	3.65 (−77)	8.03	1.68	0.687
FZ (20)	9.96 (−8)	3.46 (−40)	1.31 (−219)	2.16 (−218)	7.55	1.96	0.401
ZK (21)	6.94 (−8)	4.22 (−45)	1.56 (−238)	2.58 (−237)	7.90	1.99	0.410
*SM*1 (14)	2.46 (−9)	1.12 (−51)	5.48 (−247)	5.19 (−246)	8.10	2.00	0.404
*SM*2 (15)	4.56 (−11)	4.19 (−63)	1.04 (−287)	9.91 (−287)	8.10	2.00	0.401

D stands for fails to converge.

Table 7. Numerical comparison of several iteration schemes with memory for $\phi_7(\omega)$

Methods	$\lvert\omega_1-\alpha\rvert$	$\lvert\omega_2-\alpha\rvert$	$\lvert\omega_3-\alpha\rvert$	$f(\omega_3)$	COC	EI	CPU
SH (17)–(24), $L=-0.7$	4.30 (-1)	1.47 (-7)	3.56 (-51)	1.25 (-50)	6.64	1.60	0.610
SH (17)–(25), $L=-0.7$	4.30 (-1)	1.80 (-5)	1.47 (-37)	4.80 (-37)	7.17	1.63	0.719
SH (17)–(26), $L=-0.7$	4.30 (-1)	6.26 (-6)	8.05 (-41)	2.61 (-40)	7.07	1.63	0.720
NC (22)–(24), $L=-0.7$	D	D	D	D	–	–	–
NC (22)–(25), $L=-0.7$	D	D	D	D	–	–	–
NC (22)–(26), $L=-0.7$	D	D	D	D	–	–	–
JN (23)–(24), $L=-0.7$	1.43	1.01 (-1)	6.08 (-16)	1.97 (-15)	5.46	1.52	0.609
JN (23)–(25), $L=-0.7$	1.43	5.50 (-2)	5.29 (-11)	1.72 (-10)	5.30	1.51	0.688
JN (23)–(26), $L=-0.7$	1.43	2.94 (-3)	4.46 (-18)	1.45 (-17)	5.01	1.49	0.656
XW (27)–(28), $L=-0.7$, $T=2$	5.08 (-1)	5.88 (-5)	7.55 (-34)	2.45 (-33)	7.14	1.63	0.641
XW (27)–(29), $L=-0.7$, $T=2$	5.08 (-1)	1.82 (-5)	6.50 (-38)	2.11 (-37)	7.12	1.63	0.718
XW (27)–(30),$L=-0.7$, $T=2$	5.08 (-1)	6.45 (-5)	1.60 (-33)	5.20 (-33)	7.14	1.63	0.781
XW (27)–(31), $L=-0.7$, $T=2$	5.08 (-1)	2.40 (-5)	5.84 (-37)	1.89 (-36)	7.13	1.68	0.797
FZ (20)	3.35 (-2)	5.16 (-16)	6.68 (-116)	2.17 (-115)	7.58	1.96	0.47
ZK (21)	2.51 (-6)	5.22 (-38)	3.55 (-190)	2.17 (-189)	7.92	1.99	0.442
$SM1$ (14)	2.51 (-7)	2.69 (-40)	5.15 (-199)	1.67 (-198)	8.00	2.00	0.434
$SM2$ (15)	2.41 (-8)	6.17 (-47)	1.41 (-225)	4.61 (-225)	8.00	2.00	0.457

D stands for fails to converge.

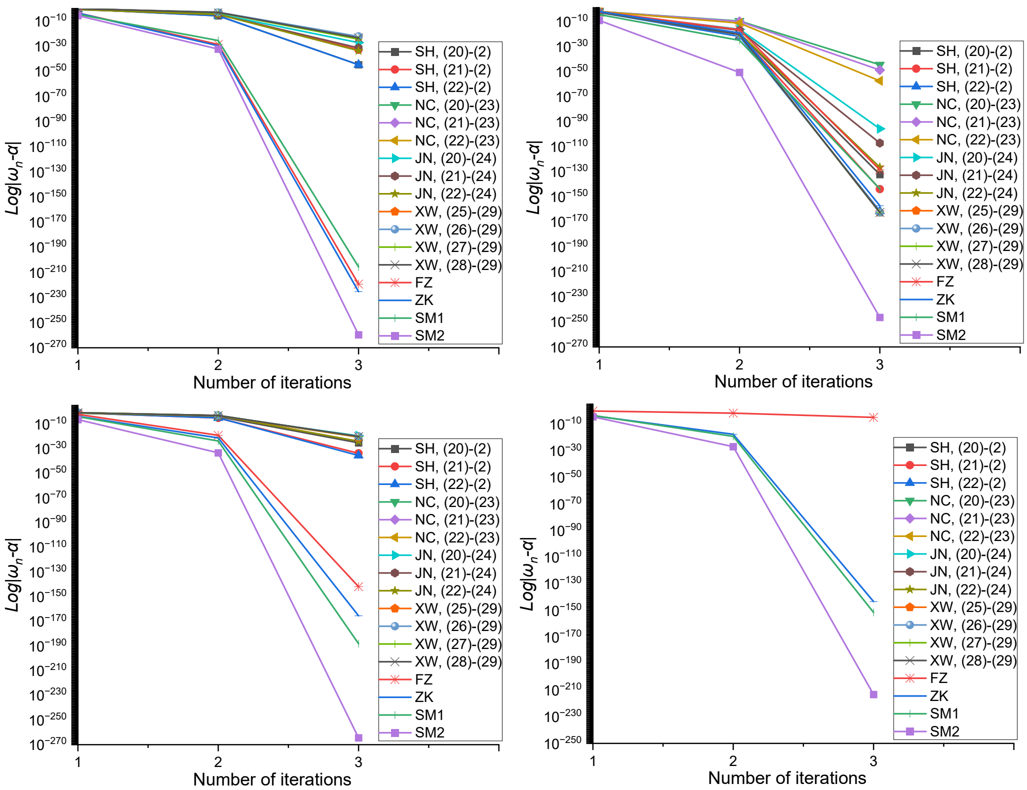

Figure 3. *Cont.*

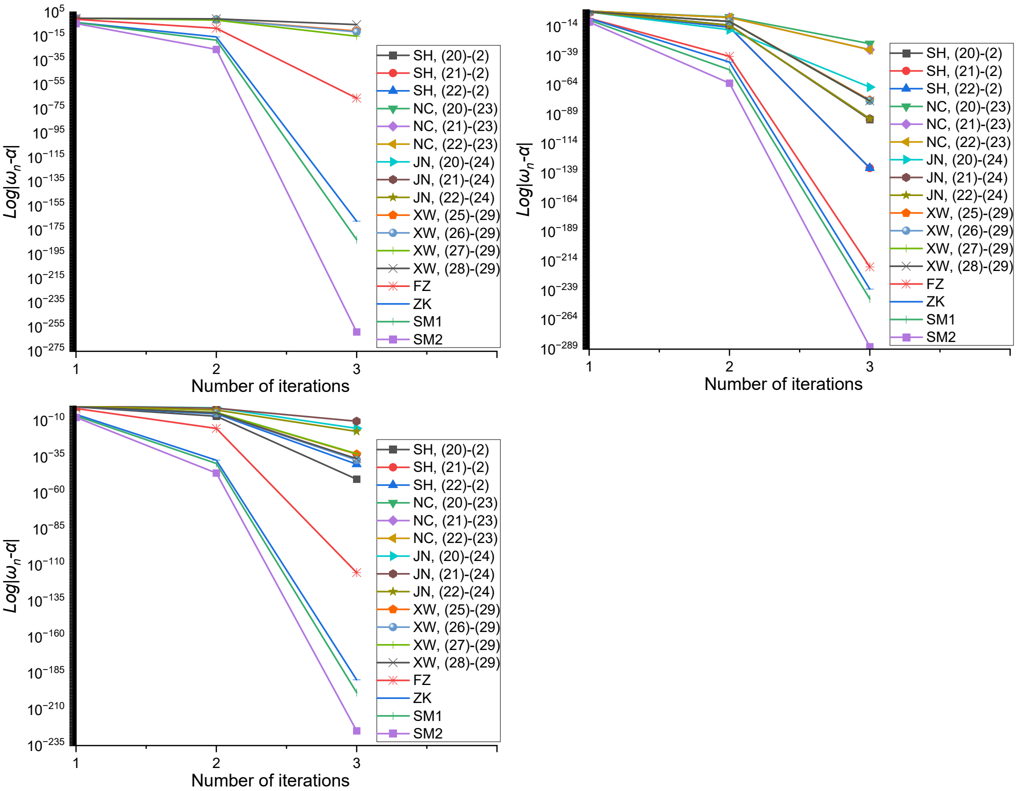

Figure 3. Comparisons of various iterative methods with-memory in terms of absolute error $|\omega_j - \alpha|$ for $\phi_1(\omega) - \phi_7(\omega)$ in first three iterations.

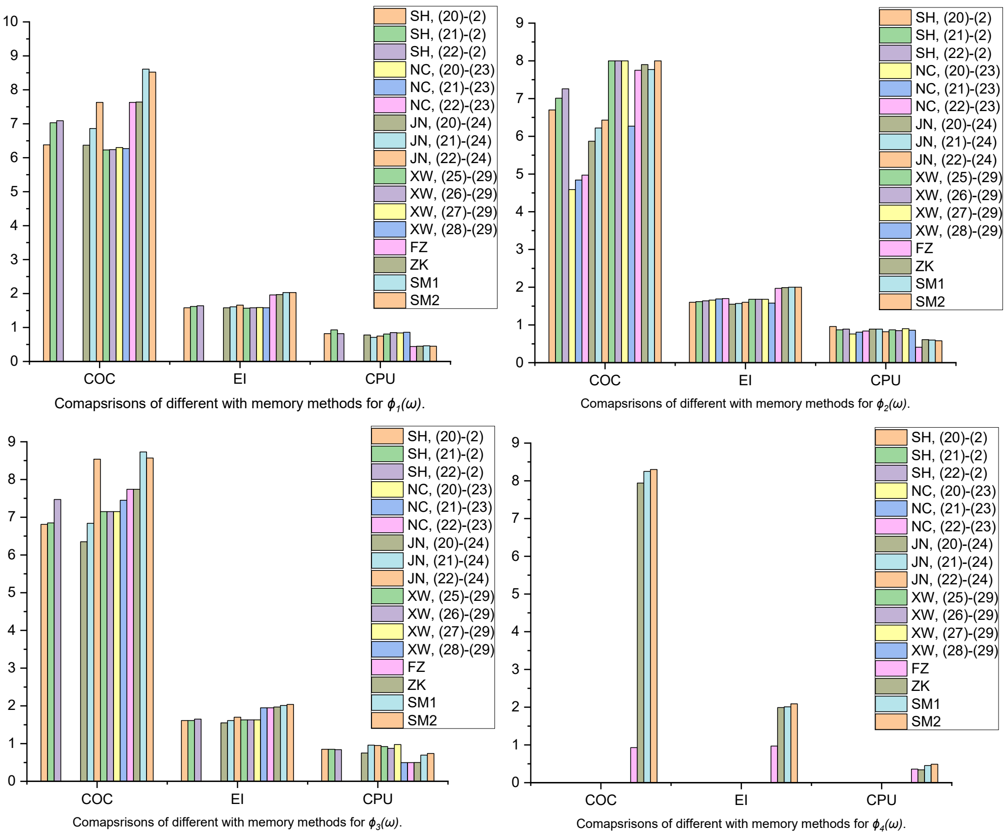

Figure 4. *Cont*.

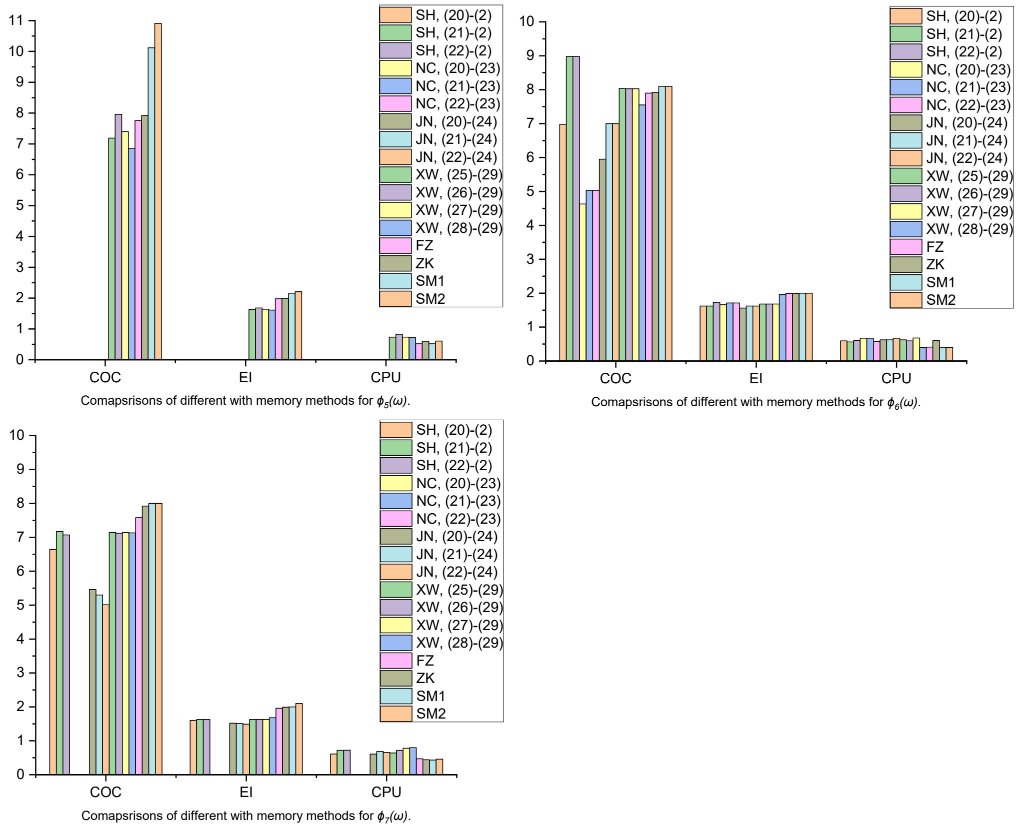

Figure 4. Comparisons of various iterative methods with-memory in terms of COC, EI, and CPU time for $\phi_1(\omega) - \phi_7(\omega)$ respectively.

6. Fractals of Basins of Attraction

In this section, we investigate the comparison of fractal behavior of the proposed iteration method (14) for different values of λ with the iterative schemes SH (17), FZ (20), ZK (21), and NC (22) discussed in Section 5. We compare their fractal behavior in terms of basins of attraction in the complex plane, which helps us to better understand their stability and convergence. Let ϕ be a nonlinear function to be solved by an iterative algorithm; we know that, in general, the the boundary between the basins of attraction for distinct zeros of ϕ represents a complex fractal form. By assigning a specific color to each basin, we generally obtain very beautiful fractals, which illustrate the performance of iterative methods. Initially, Stewart [5] and Varona [6] presented a graphical comparison between some classical iterative methods in 2001 and 2002, respectively. After that, it is a common practice to compare iteration methods graphically with the help of fractal images of basins of attraction. The book of Kalantari [34] provided several artistic fractal pictures of different polynomials. More recently, this kind of comparison has been studied in the papers [7,25,35–37]. All of these papers present a comparison of the methods by plotting basins of attraction for simple polynomials of the form $z^n - 1$ in the complex plane. We investigate convergence regions of different methods by representing their basins of attractions on the variety of nonlinear equations, including real-life problems discussed in Section 5.

To plot fractals of basins of attraction, we chose an initial guess z_0 from a grid of 500×500 points within a square D contained in $\mathbb{C}$ such that it contains all of the roots of $\phi(z) = 0$, each allocated by a unique color. For a given initial point in D, an iteration scheme within 25 iterations either converges to one of the roots, painted with a color assigned to that root, or diverges, usually marked with the color black. The brighter color of the basins indicates that a few iterations required for a method to converge to the root.

Basins of attraction of of $\phi_1(z) = 0$ are shown in Figure 5, which has two roots $0, 4.96511$ contained in $D = [-5,5] \times [-5,5]$, represented by colors—cyan and magenta, respectively. Due to the limited space, we have written reduced significant digits of the roots. Figure 5 illustrates that the methods ZK and $SM(\lambda = 0.1)$ show wide basins of attraction as compared to those of $SH(L_0 = 0.1), FZ, NC(L_0 = 0.1)$ and $SM(\lambda = 2, 1)$ while fast convergence is obtained by SH and $SM(\lambda = 0.1)$.

For the nonlinear function $\phi_2(z)$, which has the root 0.415856, we take $D = [-2,2] \times [-2,2]$ and assign color cyan to each initial point in D for which the method converges to 0.415856. Fractals of basins for this problem are represented in Figure 6, which illustrate that all the methods possess similar regions of convergence except the method $NC(L_0 = 0.1)$ with several black regions. The method $SM(\lambda = 0.1)$ has fast convergence for initial points near the root since its basins are brighter than those of ZK and $NC(L_0 = 0.1)$.

Similarly, we take $D = [-2,2] \times [-2,2]$, for $\phi_3(z) = 0$, which has the root 0. We assign color cyan to each initial point in D for which an iteration method converges to 0. Figure 7 represents the fractals of basins for this problem which illustrate that the proposed methods $SM(\lambda = 1)$, $SM(\lambda = 0.5)$, $SM(\lambda = 0.1)$ provide wide basins of attractions and have fast convergence for initial points near the root than those of FZ, ZK and $NC(L_0 = 0.1)$.

We take $D = [-2,2] \times [-2,2]$, for the nonlinear function $\phi_4(z)$, which has six roots; $1, -1.40360, -0.454979 - 0.649504i, -0.454979 + 0.649504i, 0.656780 - 0.837592i, 0.656780 + 0.837592i$, represented by green, cyan, yellow, orange, red and magenta, respectively. Fractals of basins for $\phi_4(z) = 0$ are represented in Figure 8, which illustrates that the methods $SM(\lambda = 0.1)$ and $SH(L_0 = 0.1)$ are the best since they produce simple and wide regions of convergence as compared to other methods.

Fractal images of basins of attraction of $\phi_5(z) = 0$ are shown in Figure 9, which has three real roots; $0, 2, 4.27478$ contained in $D = [-5,5] \times [-5,5]$, represented by cyan, magenta and yellow, respectively. Figure 9 illustrates that all the methods produce wide regions of divergence (black regions); however, the methods $SM(\lambda = 0.1, 0.5)$ and $SH(L_0 = 0.1)$ have comparatively better performances in terms of speed and regions of convergence.

For the nonlinear function $\phi_6(z)$, which has roots $-1, -1.73205, 1.73205$, we take $D = [-5,5] \times [-5,5]$ and assign the colors magenta, cyan and yellow to each initial point in D, for which the method converges to -1, -1.73205 and 1.73205, respectively. Fractals of basins for $\phi_6(z) = 0$ are represented in Figure 10, which illustrates that the proposed method $SM(\lambda = 0.1)$ provide fast convergence with simple fractals and wide regions of convergence as compared to others except the methods $SH(L_0 = 0.1)$ and $NC(L_0 = 0.1)$.

Basins of attraction of $\phi_7(z) = 0$ are shown in Figure 11, which has eight roots contained in $D = [-5,5] \times [-5,5]$; $0.104698, 3.82238, -2.27869 - 1.98747i, -2.27869 + 1.98747i, -1.23876 - 3.40852i, -1.23876 + 3.40852i, 1.55391 - 0.940414i, 1.55391 + 0.940414i$, represented by colors, cyan, green, orange, yellow, red, magenta, pink and brown respectively. Figure 11 illustrates that the proposed method $SM(\lambda = 0.1, 0.5)$ is the best one among all others, yielding fast convergence and simple fractals and wide regions of convergence. However, none of the methods converge to the roots $1.553919 - 0.940414i$ and $1.55391 + 0.940414i$.

It is observed that for all of the problems, the proposed iteration scheme SM for $\lambda = 0.1$ provides wider and brighter basins of attraction with simple fractals which yields its stability and robustness. Furthermore, the smaller values of the parameter λ result in wider basins of attraction for the proposed iteration schemes.

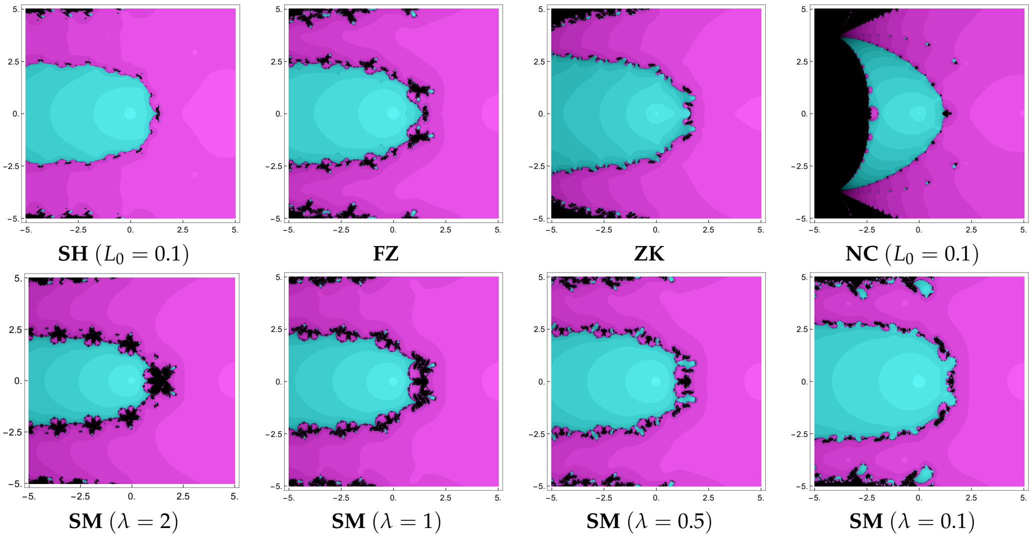

Figure 5. Basins of attraction of $\phi_1(z)$ using several iteration methods without memory.

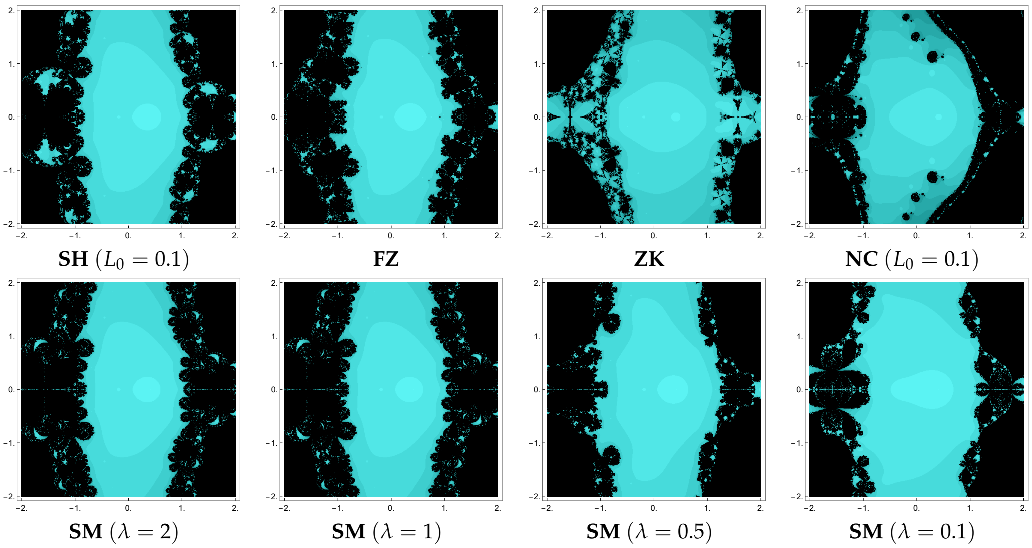

Figure 6. Basins of attraction of $\phi_2(z)$ using several iteration methods without memory.

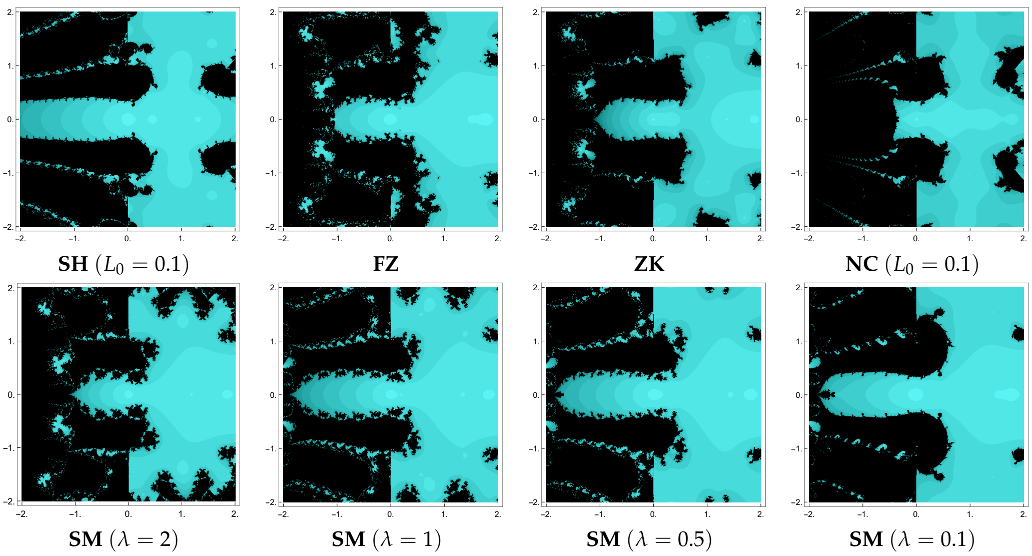

Figure 7. Basins of attraction of $\phi_3(z)$ using several iteration methods without memory.

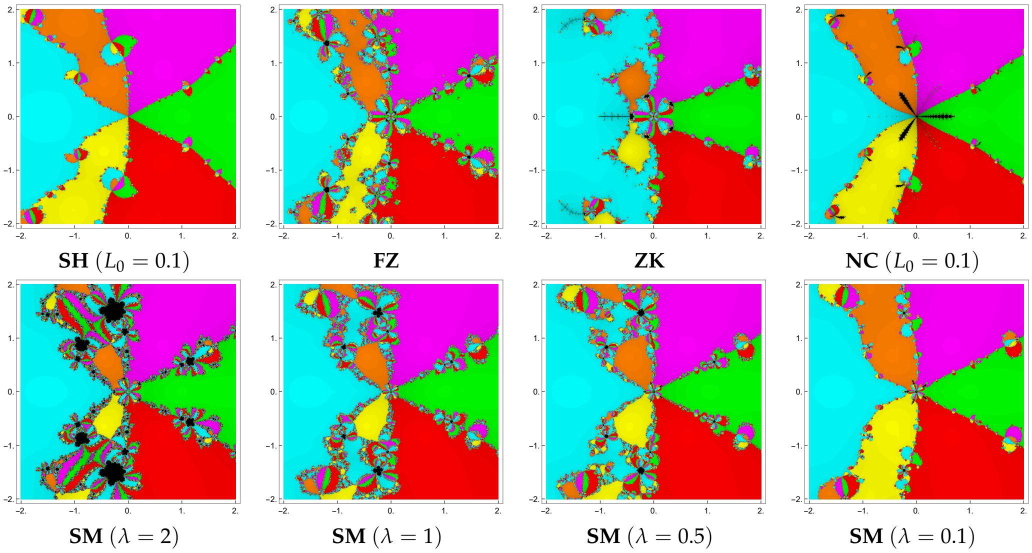

Figure 8. Basins of attraction of $\phi_4(z)$ using several iteration methods without memory.

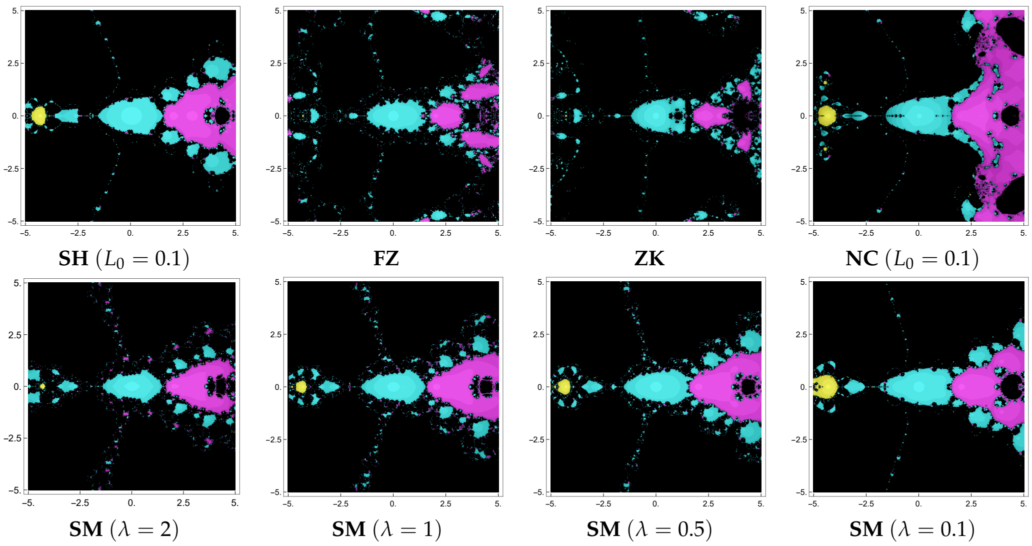

Figure 9. Basins of attraction of $\phi_5(z)$ using several iteration methods without memory.

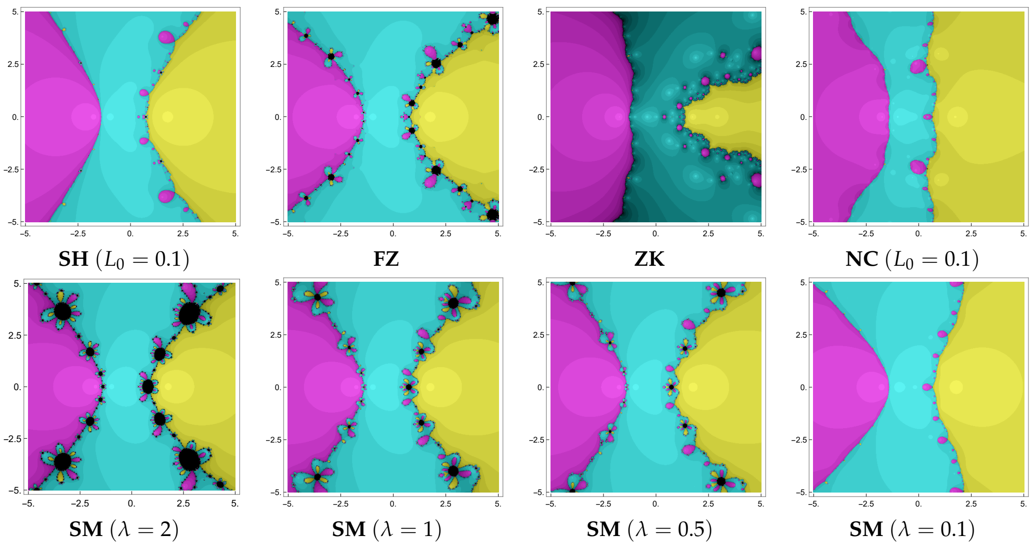

Figure 10. Basins of attraction of $\phi_6(z)$ using several iteration methods without memory.

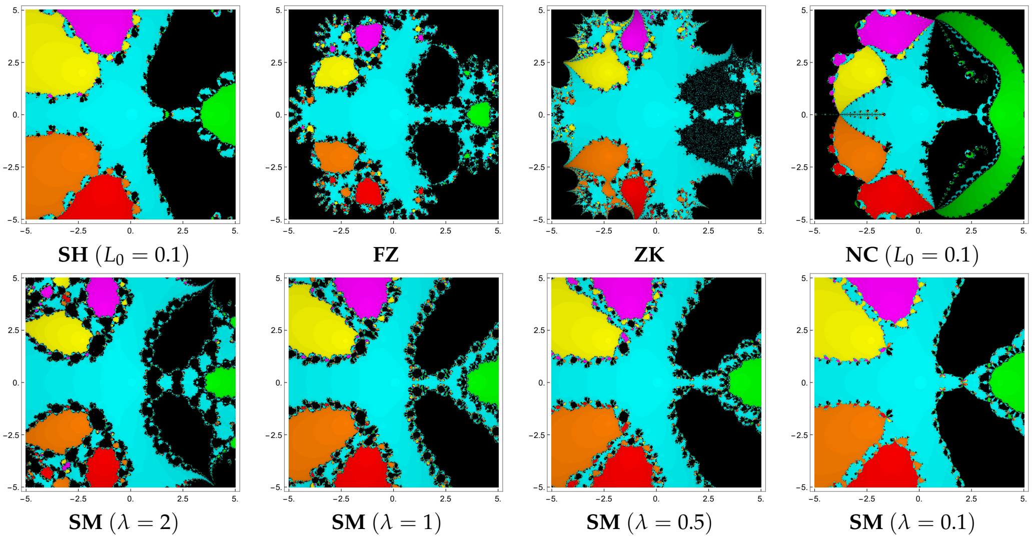

Figure 11. Basins of attraction of $\phi_7(z)$ using several iteration methods without memory.

7. Conclusions

In this manuscript, we have introduced derivative-free two-step iteration methods of optimal orders four and eight without memory and with-memory, respectively, for solving nonlinear equations. The suggested techniques are higher-order two-step variants of the one-step Traub's method of optimal order two. It is to be remarked that the eighth-order convergence of the proposed iteration technique with memory is achieved by using only three functional evaluations. The proposed two-step technique's efficiency index is $7.993^{1/3} \approx 2$, making it the highest in the literature and better than the efficiency of several multi-step iteration schemes with-memory. The proposed two-step iteration methods with-memory compete with any j-point optimal method without memory since its efficiency index equals 2. To evaluate the effectiveness of the suggested iterative techniques and to support the theoretical findings, several numerical examples and real-world applications are given. The numerical outcomes of the proposed methods are presented in terms of absolute error, computational order of convergence (COC), and CPU time (sec). Further, we have investigated the fractal behavior and comparison of different iteration methods using fractals of basins of attraction on several nonlinear equations, including real-life problems. The fractals of basins of attractions illustrate the robustness and superiority of the proposed iteration methods without memory. The stability of the proposed iteration methods without memory is affirmed by the simple fractals defined by their wider basins of attraction in comparison with existing iteration methods. Additionally, the numerical tests illustrate that the proposed two-step Traub–Steffensen type iteration schemes with memory outperform existing multi-step iteration schemes with and without memory in many situations. Further research can be conducted to explore general criteria for the selection of free parameters. The current study focuses on the solution of univariate nonlinear equations, while its extension to multivariate equations is left for future research.

Author Contributions: Conceptualization, M.-u.-D.J.; Methodology, M.-u.-D.J.; Validation, M.-u.-D.J., S.A. (Shahid Abdullah), M.K. and S.A. (Shabbir Ahmad); Investigation, M.-u.-D.J., S.A. (Shahid Abdullah), M.K. and S.A. (Shabbir Ahmad); Writing—original draft, M.-u.-D.J.; Writing—review & editing, S.A. (Shahid Abdullah), M.K. and S.A. (Shabbir Ahmad). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Acknowledgments: The authors are thankful to the editor and anonymous reviewers for their valuable suggestions and comments.

Conflicts of Interest: The authors declare that there are no conflicts of interest.

References

1. Ortega, J.M.; Rheinboldt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*; Academic Press: New York, NY, USA, 1970.
2. Ostrowski, A.M. *Solution of Equations and Systems of Equations*; Academic Press: New York, NY, USA, 1960.
3. Traub, J.F. *Iterative Methods for the Solution of Equations*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1964.
4. Steffensen, I.F. Remarks on iteration. *Scand. Actuar. J.* **1933**, *16*, 64–72. [CrossRef]
5. Stewart, B.D. Attractor Basins of Various Root-Finding Methods. Ph.D. Thesis, Naval Postgraduate School, Monterey, CA, USA, 2001.
6. Varona, J.L. Graphic and numerical comparison between iterative methods. *Math. Intell.* **2002**, *24*, 37–47. [CrossRef]
7. Varona, J.L. An Optimal Thirty-Second-Order Iterative Method for Solving Nonlinear Equations and a Conjecture. *Qual. Theory Dyn. Syst.* **2022**, *21*, 39. [CrossRef]
8. Kung, H.T.; Traub, J.F. Optimal order of one-point and multipoint iteration. *Assoc. Comput. Math.* **1974**, *21*, 643–651. [CrossRef]
9. King, R.F. A family of fourth-order methods for nonlinear equations. *SIAM J. Numer. Anal.* **1973**, *10*, 876–879. [CrossRef]
10. Behl, R.; Alshomrani, A.S.; Chun, C. A general class of optimal eighth-order derivative free methods for nonlinear equations. *J. Math. Chem.* **2020**, *58*, 854–867. [CrossRef]
11. Cordero, A.; Reyes, J.A.; Torregrosa, J.R.; Vassileva, M.P. Stability analysis of a new fourth-order optimal iterative scheme for nonlinear equations. *Axioms* **2024**, *13*, 34. [CrossRef]
12. Moscoso-Martinez, M.; Chicharro, F.I.; Cordero, A.; Torregrosa, J.R.; Urena-Callay, G. Achieving Optimal Order in a Novel Family of Numerical Methods: Insights from Convergence and Dynamical Analysis Results. *Axioms* **2024**, *13*, 458. [CrossRef]
13. Petković, L.D.; Petković, M.S.; Džunić, J. A class of three-point root-solvers of optimal order of convergence. *Appl. Math. Comput.* **2010**, *216*, 671–676. [CrossRef]
14. Petković, M.S.; Neta, B.; Petković, L.D.; Džunić, J. *Multipoint Methods for Solving Nonlinear Equations*; Elsevier: Amsterdam, The Netherlands, 2013.
15. Sharma, J.R.; Sharma, R. A new family of modified Ostrowski's methods with accelerated eighth order convergence. *Numer. Algor.* **2010**, *54*, 445–458. [CrossRef]
16. Sharma, J.R.; Kumar, S.; Singh, H. A new class of derivative-free root solvers with increasing optimal convergence order and their complex dynamics. *SEMA J.* **2023**, *8*, 333–352. [CrossRef]
17. Wang, X.; Liu, L. New eighth-order iterative methods for solving nonlinear equations. *Comput. Appl. Math.* **2010**, *234*, 1611–1620. [CrossRef]
18. Zheng, Q.; Li, J.; Huang, F. An optimal Steffensen-type family for solving nonlinear equations. *Appl. Math. Comput.* **2011**, *217*, 9592–9597. [CrossRef]
19. Choubey, N.; Panday, B.; Jaiswal, J.P. Several two-point with memory iterative methods for solving non-linear equations. *Afr. Mat.* **2018**, *29*, 435–449. [CrossRef]
20. Choubey, N.; Jaiswal, J.; Choubey, A. Family of multipoint with memory iterative schemes for solving nonlinear equations. *Int. J. Appl. Comput. Math.* **2022**, *8*, 83. [CrossRef]
21. Cordero, A.; Garrido, N.; Torregrosa, J.R.; Triguero-Navarro, P. Three-step iterative weight function scheme with memory for solving nonlinear problems. *Math. Method. Appl. Sci.* **2022**, *early view*. [CrossRef]
22. Cordero, A.; Garrido, N.; Torregrosa, J.R.; Triguero-Navarro, P. Memory in the iterative processes for nonlinear problems. *Math. Method. Appl. Sci.* **2023**, *46*, 4145–4158. [CrossRef]
23. Cordero, A.; Garrido, N.; Torregrosa, J.R.; Triguero-Navarro, P. Design of iterative methods with memory for solving nonlinear systems. *Math. Method. Appl. Sci.* **2023**, *46*, 12361–12377. [CrossRef]
24. Petković, M.S.; Neta, B.; Petković, L.D.; Džunić, J. Multipoint methods for solving nonlinear equations: A survey. *Appl. Math. Comput.* **2014**, *226*, 635–660. [CrossRef]
25. Sharma, S.; Kansal, M. A modified Chebyshev–Halley-type iterative family with memory for solving nonlinear equations and its stability analysis. *Math. Method. Appl. Sci.* **2023**, *46*, 12549–12569. [CrossRef]
26. Soleymani, F.; Lotfi, T.; Tavakoli, E.; Khaksar, F. Haghani, Several iterative methods with memory using self accelerators. *Appl. Math. Comput.* **2015**, *254*, 452–458.
27. Zafar, F.; Yasmin, N.; Kutbi, M.A.; Zeshan, M. Construction of Tri-parametric derivative free fourth order with and without memory iterative method. *J. Nonli. Sci. Appl.* **2016**, *9*, 1410–1423. [CrossRef]
28. Ullaha, M.Z.; Kosari, S.; Soleymani, F.; Haghani, F.K.; Al-Fhaid, A.S. A super-fast tri-parametric iterative method with memory. *Appl. Math. Comput.* **2016**, *289*, 486–491.
29. Akram, S.; Khalid, M.; Junjua, M.-U.; Altaf, S.; Kumar, S. Extension of King's Iterative Scheme by Means of Memory for Nonlinear Equations. *Symmetry* **2023**, *15*, 1116. [CrossRef]
30. 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.* **2023**, *70*, 285–315. [CrossRef]
31. Herzberger, J. Über matrixdarstellungen fur iterationverfahren bei nichtlinearen gleichungen. *Computing* **1974**, *12*, 215–222. [CrossRef]

32. Jay, L. A note on Q-order of convergence. *BIT Numer. Math.* **2001**, *41*, 422–429. [CrossRef]
33. Wang, X.; Zhang, T. Some Newton-type iterative methods with and without memory for solving nonlinear equations. *Int. J. Comput. Methods* **2014**, *11*, 1350078. [CrossRef]
34. Kalantari, B. *Polynomial Root-Finding and Polynomiography*; World Scientific: Singapore, 2009.
35. Cordero, A.; Guasp, L.; Torregrosa, J.R. Choosing the most stable members of Kou's family of iterative methods. *J. Comput. Appl. Math.* **2018**, *330*, 759–769. [CrossRef]
36. Herceg, D.; Herceg, D. Eighth order family of iterative methods for nonlinear equations and their basins of attraction. *J. Comput. Appl. Math.* **2018**, *343*, 458–480. [CrossRef]
37. Herceg, D.; Petković, I. Computer visualization and dynamic study of new families of root-solvers. *J. Comput. Appl. Math.* **2022**, *401*, 16. [CrossRef]