symmetry

MDPI

Article

Development of Optimal Iterative Methods with Their Applications and Basins of Attraction

Waikhom Henarita Chanu *, Sunil Panday and G. Thangkhenpau

Department of Mathematics, National Institute of Technology Manipur, Langol, Imphal 795004, India
* Correspondence: henaritawai@gmail.com or waikhomhenarita@nitmanipur.ac.in

Abstract: In this paper, we construct variants of Bawazir's iterative methods for solving nonlinear equations having simple roots. The proposed methods are two-step and three-step methods, with and without memory. The Newton method, weight function and divided differences are used to develop the optimal fourth- and eighth-order without-memory methods while the methods with memory are derivative-free and use two accelerating parameters to increase the order of convergence without any additional function evaluations. The methods without memory satisfy the Kung–Traub conjecture. The convergence properties of the proposed methods are thoroughly investigated using the main theorems that demonstrate the convergence order. We demonstrate the convergence speed of the introduced methods as compared with existing methods by applying the methods to various nonlinear functions and engineering problems. Numerical comparisons specify that the proposed methods are efficient and give tough competition to some well known existing methods.

Keywords: simple roots; nonlinear equation; iterative methods; error

check for updates

Citation: Chanu, W.H.; Panday, S.; Thangkhenpau, G. Development of Optimal Iterative Methods with Their Applications and Basins of Attraction. *Symmetry* **2022**, *14*, 2020. https://doi.org/10.3390/sym14102020

Academic Editors: Juan Luis García Guirao and Alexander Zaslavski

Received: 25 August 2022
Accepted: 19 September 2022
Published: 26 September 2022

1. Introduction

Finding the roots of nonlinear equations is one of the most challenging problems in applied mathematics, engineering and scientific computing. Analytical methods are generally ineffective for finding the roots of a nonlinear equation. Consequently, iterative methods are employed to obtain the approximate roots of nonlinear equations. Many iterative methods for solving nonlinear equations have been developed and studied. Among these, Newton's method is one of the most widely used [1], which is defined as follows:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, 3, \ldots \tag{1}$$

Other well-known iterative approaches for solving nonlinear equations include the Chebyshev [2], Halley [2] and Ostrowski [3], methods. Most of the authors try to improve the order of convergence. As the order of convergence rises, so does the quantity of functional evaluations. As a result, iterative methods' efficiency index falls. The efficiency index [2,3] of an iterative method determines the method's efficiency, which is defined by the formula below:

$$E = \rho^{\frac{1}{\lambda}} \tag{2}$$

where ρ is the order of convergence and λ is the number of functional evaluations per step. Kung–Traub conjectured [2] that the order of convergence of an iterative method without memory is at most $2^{\lambda-1}$. The optimal method is one in which the order of convergence is $2^{\lambda-1}$. In 2022, Panday S. et al. created optimal methods [4]. In 2015, Kumar M. et al. developed a fifth-order derivative-free method [5]. Choubey N. et al. introduced the derivative-free eighth-order method [6] in 2015. Tao Y. et al. developed optimal methods [7]. Neta B. also developed a derivative-free method [8]. Singh M. Kumar et al. developed the eighth-order optimal method in 2021 [9]. In 2021, Said Solaiman O. et al. [10] developed an optimal eighth-order method. Chanu W. H. et al. [11] created a nonoptimal tenth-order method in 2022. This paper presents optimal fourth- and optimal eighth-order methods for solving simple roots of nonlinear equations, with efficiency indice of $4^{1/3} = 1.5874$ and $8^{1/4} = 1.6817$, respectively. The efficiency indice of with-memory methods of orders 5.7 and 11 are $5.7^{1/3} = 1.7863$ and $11^{1/4} = 1.8211$ respectively. The remaining part of the manuscript is structured as follows. In Section 2, we describe the development of methods without memory using divided difference and weight function techniques. The order of convergence of without memory with derivative is analysed in Section 2. The development of derivative free with-memory methods along with convergence analysis are in Section 3. We present numerical tests to compare the proposed methods with other known optimal methods in Section 4. In Section 5, the proposed without memory methods are discussed in the complex plane using the basins of attraction. Finally, Section 6 covers the conclusions of the study.

2. Development of the Methods and Convergence Analysis

In 2021, Bawazir H. M. developed the following nonoptimal seventh-order method [12]

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ z_n &= y_n - \frac{f(y_n)(1+\frac{A}{2})}{f'(y_n)} \\ x_{n+1} &= z_n + \frac{f(z_n)f(y_n)(1+\frac{A}{2})}{(f(z_n)-f(y_n))f'(y_n)} \end{aligned} \tag{3}$$

where $A = \dfrac{f(y_n)(f'(x_n)-f'(y_n))}{f(x_n)f'(y_n)}$.

We take the first and second steps of method (3) and replace $f'(y_n)$ by the divided difference $f[y_n, x_n] = \dfrac{f(y_n)-f(x_n)}{y_n - x_n}$; weighted by a function $Q(t_n)$, we obtain the following fourth-order optimal method.

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ x_{n+1} &= y_n - Q(t_n)\frac{f(y_n)(1+\frac{A}{2})}{f[y_n, x_n]} \end{aligned} \tag{4}$$

where $A = \dfrac{f(y_n)(f'(x_n)-f[y_n, x_n])}{f(x_n)f[y_n, x_n]}$ and $Q : \mathbb{R} \to \mathbb{R}$ is the weight function, which is a sufficiently differentiable function at the point 0 with $t_n = \dfrac{f(y_n)}{f(x_n)}$.

Theorem 1. *Let $f : \mathbb{I} \subset \mathbb{R} \to \mathbb{R}$ be a real-valued, sufficiently differentiable function. Let $\mu \in \mathbb{I}$ be a simple root of f and x_0 be sufficiently close to μ; then, the iterative scheme defined in (4) is of fourth order of convergence if $Q(t_n)$ satisfies $Q(0) = 1$, $Q'(0) = 2$ and $Q''(0) = 9$, and (4) satisfies the following error equation*

$$\epsilon_{n+1} = -K_2K_3\epsilon_n^4 + O[\epsilon_n]^5 \tag{5}$$

Proof of Theorem 1. Let μ be the simple root of $f(x) = 0$ and let $\epsilon_n = x_n - \mu$ be the error of n^{th} iteration. Using Taylor expansion, we obtain

$$f(x_n) = f'(\mu)\left(\epsilon_n + \sum_{i=2}^{4} K_i\epsilon_n^i + O[\epsilon_n]^5]\right). \tag{6}$$

where $K_i = \frac{f^{(i)}(\mu)}{i!f'(\mu)}$

$$f'(x_n) = f'(\mu)\left(1 + \sum_{i=2}^{4} iK_i\epsilon_n^{i-1} + O[\epsilon_n]^5]\right). \tag{7}$$

Using Equations (6) and (7) in the first step of (4), we obtain the following

$$y_n - \mu = K_2\epsilon_n^2 + (-2K_2^2 + 2K_3)\epsilon_n^3 + (4K_2^3 - 7K_2K_3 + 3K_4)\epsilon_n^4 + O[\epsilon_n]^5 \tag{8}$$

Expanding $f(y_n)$ about μ, we obtain

$$f(y_n) = K_2 f'(\mu)\epsilon_n^2 + 2(-K_2^2 + K_3)f'(\mu)\epsilon_n^3 + f'(\mu)\big(5K_2^3 - 7K_2K_3 + 3K_4\big)\epsilon_n^4 + O[\epsilon_n]^5 \tag{9}$$

Using the expansion of $f(x_n)$ and $f(y_n)$, we obtain

$$\begin{aligned}\frac{f(y_n)}{f(x_n)} = {} & K_2\epsilon_n + \big(-3K_2^2 + 2K_3\big)e_n^2 + \big(8K_2^3 - 10K_2K_3\big)e_n^3 + \big(-20K_2^4 \\ & + 37K_2^2K_3 - 8K_3^2 - 14K_2K_4 + 4K_5\big)\epsilon_n^4 + O[\epsilon_n]^5\end{aligned} \tag{10}$$

Moreover,

$$\begin{aligned}f[y_n, x_n] = {} & f'(\mu)(1 + K_2\epsilon_n + (K_2^2 + K_3)\epsilon_n^2 + (-2K_2^3 + 3K_2K_3 + K_4)\epsilon_n^3 \\ & + (4K_2^4 - 8K_2^2K_3 + 2K_3^2 + 4K_2K_4 + K_5)\epsilon_n^4 + O[\epsilon_n]^5)\end{aligned} \tag{11}$$

Using (6), (7), (9) and (11), we obtain

$$A = K_2^2\epsilon_n^2 + (-5K_2^3 + 4K_2K_3)\epsilon_n^3 + (17K_2^4 - 26K_2^2K_3 + 4K_3^2 + 6K_2K_4)\epsilon_n^4 + O[\epsilon_n]^5 \tag{12}$$

Using (9), (11) and (12) in (4), we obtain

$$\begin{aligned}\epsilon_{n+1} = {} & K_2(1 - Q(0))\epsilon_n^2 + \big(2K_3(1 - Q(0)) + K_2^2(-2 + 4Q(0) \\ & - Q'(0))\big)\epsilon_n^3 + \big(3K_4(1 - Q(0)) + K_2K_3(-7 + 14Q(0) \\ & - 4Q'(0)) + \frac{1}{2}K_2^3(8 - 27Q(0) + 14Q'(0) - (Q''(0))\big)\epsilon_n^4 \\ & + O[\epsilon_n]^5\end{aligned} \tag{13}$$

To achieve the fourth order of convergence, we put $Q(0) = 1$, $Q'(0) = 2$ and $Q''(0) = 9$ and obtain the following error equation

$$\epsilon_{n+1} = -K_2K_3\epsilon_n^4 + O[\epsilon_n]^5 \tag{14}$$

From Equation (14), we conclude that the method (4) is of the fourth order of convergence. □

The new eighth-order optimal method is obtained by adding the following equation as the third step to the method (4).

$$x_{n+1} = z_n + \frac{f(z_n)f(y_n)\left(1 + \frac{A}{2}\right)}{\big(f(z_n) - f(y_n)\big)f'(z_n)} \tag{15}$$

where z_n is the second step of method (4). To obtain the optimal method, $f'(z_n)$ is approximated by $h(z_n, y_n, x_n)$ and weighted by a function $Q : \mathbb{R} \to \mathbb{R}$, and the method is given by

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ z_n &= y_n - Q(t_n)\frac{f(y_n)(1+\frac{A}{2})}{f[y_n,x_n]} \\ x_{n+1} &= z_n + G(t_n,s_n)\frac{f(z_n)f(y_n)(1+\frac{A}{2})}{(f(z_n)-f(y_n))h(z_n,y_n,x_n)} \end{aligned} \tag{16}$$

where $h(z_n,y_n,x_n) = f[z_n,y_n] - f'(x_n)$, $A = \dfrac{f(y_n)(f'(x_n) - f[y_n,x_n])}{f(x_n)f[y_n,x_n]}$ and $Q : \mathbb{R} \to \mathbb{R}$ and $G : \mathbb{R}^2 \to \mathbb{R}$ are the weight functions with $t_n = \dfrac{f(y_n)}{f(x_n)}$ and $s_n = \dfrac{f(z_n)}{f(y_n)}$.

Theorem 2. *Let $f : \mathbb{I} \subset \mathbb{R} \to \mathbb{R}$ be a real-valued, sufficiently differentiable function. Let $\mu \in \mathbb{I}$ be a simple root of f and x_0 be sufficiently close to μ; then, the iterative scheme defined in (16) is of the eighth order of convergence if $Q(t_n)$ and $G(t_n,s_n)$ satisfy the following conditions $Q(0) = 1$, $Q'(0) = 2$ and $Q''(0) = 9$, $G(0,0) = 0$, $G^{(1,0)}(0,0) = -2$, $G^{(0,1)}(0,0) = -1$, $G^{(2,0)}(0,0) = -10$, $G^{(1,1)}(0,0) = 0$, $G^{(0,2)}(0,0) = 0$, $G^{(2,1)}(0,0) = -9$, $G^{(4,0)}(0,0) = -Q^{(4)}(0) - 318$ and $G^{(3,0)}(0,0) = -\Big(Q^{(3)}(0) + 15\Big)$. Equation (16) satisfies the following error equation*

$$\begin{aligned} \epsilon_{n+1} = &-\frac{1}{240}\Bigg(K_2K_3\Bigg(-20K_2^2K_3\Big(G^{(3,1)}(0,0)+129\Big) + K_2^4(G^{(5,0)}(0,0) + 100Q^{(3)}(0) \\ &+ Q^{(5)}(0) + 60) + 60K_3^2\Big(G^{(1,2)}(0,0)+4\Big) + 240K_2K_4\Bigg)\Bigg)\epsilon_n^8 + O[\epsilon_n]^9 \end{aligned} \tag{17}$$

Proof of Theorem 2. Considering all the assumptions made in Theorem 1, from Equation (14) we have

$$z_n - \mu = -K_2K_3\epsilon_n^4 + \sum_{j=5}^{8} D_j\epsilon_n^j + O[\epsilon_n]^9. \tag{18}$$

Expanding $f(z_n)$ about μ, we obtain

$$\begin{aligned} f(z_n) =& K_2K_3f'(\mu)\epsilon_n^4 + f'(\mu)\Bigg(2K_2^2K_3 - \frac{1}{6}K_2^4\Big(Q^{(3)}(0) - 75\Big) - 2K_2K_4 - 2K_3^2\Bigg)\epsilon_n^5 \\ &+ \sum_{j=5}^{8} X_j\epsilon_n^j + O[\epsilon_n]^9 \end{aligned} \tag{19}$$

Further,

$$\begin{aligned} h(z_n,y_n,x_n) =& -2(K_2f'(\mu))\epsilon_n + (K_2^2 - 3K_3)f'\epsilon_n^2 + -2((K_2^3 - K_2K_3 + 2K_4)f'(\mu)\epsilon_n^3 \\ &+ \sum_{i}^{8} Y_i\epsilon_n^i. \end{aligned} \tag{20}$$

Using (19) and (20) in the third step of method (16), we obtain

$$\begin{aligned} \epsilon_{n+1} =& -\frac{1}{2}((K_3G(0,0)))\epsilon_n^3 + \frac{1}{12}\epsilon_n^4\Bigg(K_2^3\Big(Q^{(3)}(0) - 75\Big)(-G(0,0)) - \frac{3K_3^2G(0,0)}{K_2} \\ &- 6K_2K_3\Big(G^{(1,0)}(0,0) + 2\Big) + 9K_2K_3G(0,0) - 12K_4G(0,0)\Bigg) + \sum_{i=5}^{8} Z_i\epsilon_n^i. \end{aligned} \tag{21}$$

To eliminate ϵ_n^k, $k = 3,4,5,6,7$, we put $G(0,0) = 0$, $G^{(1,0)}(0,0) = -2$, $G^{(0,1)}(0,0) = -1$, $G^{(2,0)}(0,0) = -10$, $G^{(1,1)}(0,0) = 0$, $G^{(0,2)}(0,0) = 0$, $G^{(2,1)}(0,0) = -9$, $G^{(4,0)}(0,0) = -Q^{(4)}(0) - 318$, $G^{(3,0)}(0,0) = -\Big(Q^{(3)}(0) + 15\Big)$. Then, we obtain

$$\epsilon_{n+1} = -\frac{1}{240}\Bigg(\Big(K_2K_3\Big(-20K_2^2K_3\Big(G^{(3,1)}(0,0)+129\Big) + K_2^4\Big(G^{(5,0)}(0,0)+100Q^{(3)}(0)+Q^{(5)}(0)+60\Big) + 60K_3^2\Big(G^{(1,2)}(0,0)+4\Big)+240K_2K_4\Big)\Big)\Bigg)\epsilon_n^8 + O[\epsilon_n]^9. \tag{22}$$

From Equation (22), we conclude that (16) is of the eighth order of convergence. □

Remark 1. *The methods defined in (4) and (16) have derivatives and are without-memory methods. In the next section, we will develop derivative free with-memory methods in order to obtain a higher efficiency index.*

3. Derivative-Free and with-Memory Methods

In this section, we present derivative-free parametric and with memory iterative methods. Another Bawazir's iterative method is written as [12]

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ z_n &= y_n - \frac{f(y_n)(1+\frac{A}{2})}{f'(y_n)} \\ x_{n+1} &= z_n - \frac{f(z_n)(1+\frac{B}{2})}{f'(z_n)} \end{aligned} \tag{23}$$

where $A = \dfrac{f(y_n)(f'(x_n)-f'(y_n))}{f(x_n)f'(y_n)}$, $B = \dfrac{f(z_n)(f'(y_n)-f'(z_n))}{f(y_n)f'(z_n)}$. This method uses five function evaluation to achieve the twelfth order of convergence. We modify the method given in (23) by adding two parameters γ and β as follows:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n,w_n]+\gamma f(x_n)}, w_n = x_n - \beta f(x_n)^2 \\ x_{n+1} &= y_n - Q(t_n)\frac{f(y_n)(1+\frac{A}{2})}{F(x_n,w_n,y_n)} \end{aligned} \tag{24}$$

where $A = \dfrac{f(y_n)(f[x_n,w_n]-F(x_n,w_n,y_n))}{f(x_n)F(x_n,w_n,y_n)}$, and $Q:\mathbb{R}\to\mathbb{R}$ is the weight function with $t_n = \dfrac{f(y_n)}{f(x_n)}$, $f'(y) \approx F(x_n,w_n,y_n) = 2f[x_n,y_n]-f[x_n,w_n]$ [13] and $f[x,y] = \dfrac{f(x)-f(y)}{x-y}$.

Theorem 3. *Let $f:\mathbb{I}\subset\mathbb{R}\to\mathbb{R}$ be a real-valued, sufficiently differentiable function. Let $\mu\in\mathbb{I}$ be a simple root of f and x_0 be sufficiently close to μ; then, the iterative scheme defined in (24) is of the fourth order of convergence if $Q(t_n)$ satisfies the following conditions $Q(0)=1$, $Q'(0)=0$ and $Q''(0)=0$. The iterative scheme (24) satisfies the following error equation*

$$\epsilon_{n+1} = (\gamma+K_2)\Big(\beta K_2 f'(\mu)^2 - K_3\Big)\epsilon_n^4 + O\Big(e^5\Big) \tag{25}$$

Proof of Theorem 3. Let μ be the simple root of $f(x)=0$ and let $\epsilon_n = x_n-\mu$ be the error of n^{th} iteration. Using Taylor expansion, we obtain

$$f(x_n) = f'(\mu)\Bigg(\epsilon_n + \sum_{i=2}^{4}K_i\epsilon_n^i + O[\epsilon_n]^5]\Bigg). \tag{26}$$

where $K_i = \frac{f^{(i)}(\mu)}{i! f'(\mu)}$ Using (26) in w_n, we obtain

$$\begin{aligned} w_n - \mu = \epsilon_{w,n} =& \epsilon_n - \beta \mathrm{f}'(\mu)^2 \epsilon_n^2 - 2\Big(\beta \mathrm{K}_2 \mathrm{f}'(\mu)^2\Big)\epsilon_n^3 \\ & - \beta \mathrm{f}'(\mu)^2 \Big(\mathrm{K}_2^2 + 2\mathrm{K}_3\Big)\epsilon_n^4 + O[\epsilon_n^5] \end{aligned} \tag{27}$$

By Taylor series expansion, we obtain

$$\begin{aligned} f(w_n) =& f'(\mu)\epsilon_n + f'(\mu)(K_2 - f'(\mu)^2\beta)\epsilon_n^2 + f'(\mu)(K_3 \\ & - 4K_2 f'(\mu)^2\beta)\epsilon_n^3 + f'(\mu)(K_4 - 3K_3 f'(\mu)^2\beta \\ & - (K_2^2 + 2K_3) f'(\mu)^2\beta + K_2(-4K_2 f'(\mu)^2\beta \\ & + f'(\mu)^4\beta^2))\epsilon_n^4 + O[\epsilon^5] \end{aligned} \tag{28}$$

Using (26) and (28) in the first step of (24), we obtain

$$\begin{aligned} y_n - \mu = \epsilon_{y,n} =& f'(\mu)(\gamma + \mathrm{K}_2)\epsilon_n^2 - f'(\mu)\bigg(\gamma^2 + 2\mathrm{K}_2^2 + 2\gamma \mathrm{K}_2 \\ & + \beta \mathrm{K}_2 f'(\mu)^2 - 2\mathrm{K}_3 + \beta\gamma f'(\mu)^2\bigg)\epsilon_n^3 + O[\epsilon_n^4] \end{aligned} \tag{29}$$

Using Taylor series expansion, we obtain

$$\begin{aligned} f(y_n) =& f'(\mu)(K_2 + \gamma)\epsilon_n^2 - f'(\mu)(2K_2^2 - 2K_3 + K_2 f'(\mu)^2\beta \\ & + 2K_2\gamma + f'(\mu)^2\beta\gamma + \gamma^2)\epsilon_n^3 + O[\epsilon_n^4] \end{aligned} \tag{30}$$

Using (26) and (30), we obtain

$$\begin{aligned} t_n = \frac{f(y_n)}{f(x_n)} =& (\gamma + K_2)\epsilon_n + \bigg(- 3K_2^2 - K_2\Big(3\gamma + \beta f'(\mu)^2\Big) \\ & + 2K_3 - \gamma\Big(\gamma + \beta f'(\mu)^2\Big)\bigg)\epsilon_n^2 + \ldots + O[\epsilon_n^4] \end{aligned} \tag{31}$$

Using (26), (28), (30) and (31) in second step of (24), we obtain

$$x_{n+1} - \mu = \epsilon_{n+1} = -(K_2 + \gamma)(1 - Q(0))\epsilon_n^2 + P_3\epsilon_n^3 + P_4\epsilon_n^4 + O[\epsilon_n^5] \tag{32}$$

$P_3 = K_2^2(-Q'(0) + 2Q(0) - 2) + \beta K_2 f'(\mu)^2 (Q(0) - 1) + 2\gamma K_2(-Q'(0) + Q(0) - 1) - 2K_3$ $(Q(0) - 1) + \beta\gamma f'(\mu)^2 (Q(0) - 1) + \gamma^2(-Q'(0) + Q(0) - 1)$. etc. Putting $Q(0) = 1$, $Q'(0) = 0$ and $Q''(0) = 0$, Equation (32) becomes

$$x_{n+1} - \mu = \epsilon_{n+1} = (-K_3 + K_2 f'(\mu)^2\beta)(K_2 + \gamma)\epsilon_n^4 + O[\epsilon_n^5] \tag{33}$$

From Equation (33), we can conclude that the method (24) has fourth order of convergence, which completes the proof of Theorem 3. □

The eighth-order method is given as follows:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n, w_n] + \gamma f(x_n)}, w_n = x_n - \beta f(x_n)^2 \\ z_n &= y_n - Q(t_n)\frac{f(y_n)(1 + \frac{A}{2})}{F(x_n, w_n, y_n)} \\ x_{n+1} &= z_n - G(r_n, s_n)\frac{f(z_n)(1 + \frac{B}{2})}{H(x_n, w_n, y_n, z_n)} \end{aligned} \tag{34}$$

where $A = \dfrac{f(y_n)\big(f[x_n,w_n] - F(x_n,w_n,y_n)\big)}{f(x_n)F(x_n,w_n,y_n)}$, $B = \dfrac{f(z_n)\big(F(x_n,w_n,y_n) - H(x_n,w_n,y_n,z_n)\big)}{f(y_n)H(x_n,w_n,y_n,z_n)}$. $Q : \mathbb{R} \to \mathbb{R}$ & $G : \mathbb{R}^2 \to \mathbb{R}$ are the weight functions with $t_n = \dfrac{f(y_n)}{f(x_n)}$, $r_n = \dfrac{f(z_n)}{f(x_n)}$ and $s_n = \dfrac{f(z_n)}{f(y_n)}$, $f'(z_n) \approx H(x_n,w_n,y_n,z_n) = f[x_n,z_n] + \big(f[w_n,x_n,y_n] - f[y_n,x_n,z_n]\big)(x_n - z_n)$ [13].

Theorem 4. *Let $f : \mathbb{I} \subset \mathbb{R} \to \mathbb{R}$ be a real-valued, sufficiently differentiable function. Let $\mu \in \mathbb{I}$ be a simple root of f and x_0 be sufficiently close to μ; then, the iterative scheme defined in (34) is of eighth order of convergence if $Q(t_n)$ and $G(r_n,s_n)$ satisfy the following conditions $Q(0) = 1$, $Q'(0) = 0$, $Q''(0) = 0$, $G(0,0) = 1$, $G^{(1,0)}(0,0) = 0$, $G^{(0,1)}(0,0) = 0$ and $G^{(0,2)}(0,0) = -1$. The iterative scheme (34) satisfies the following error equation*

$$\epsilon_{n+1} = K_4(\gamma + K_2)^2\Big(\beta K_2 f'(\mu)^2 - K_3\Big)\epsilon_n^8 + O\Big(\epsilon_n^9\Big) \tag{35}$$

Proof of Theorem 4. Considering all the assumptions made in Theorem 3, we have from (33),

$$z_n - \mu = \epsilon_{n,z} = (-K_3 + K_2 f'(\mu)^2\beta)(K_2 + \gamma)\epsilon_n^4 + \sum_{j=5}^{8} C_j\epsilon_n^j + O[\epsilon_n^9] \tag{36}$$

where C_j's are constants formed by $K_i's$, β and γ.
Using Taylor expansion, we obtain

$$f(z_n) = f'(\mu)(-K_3 + K_2 f'(\mu)^2\beta)(K_2 + \gamma)\epsilon_n^4 + \sum_{j=5}^{8} f'(\mu)B_j\epsilon_n^j + O[\epsilon_n^9] \tag{37}$$

where B_j's are constants formed by $K_i's$, β and γ. Using (26), (28), (30) and (37) in third step of (34), we obtain

$$x_{n+1} - \mu = \epsilon_{n+1} = \big(K_3 - K_2 f'(\mu)^2\beta\big)(K_2 + \gamma)(G(0,0) - 1)\epsilon_n^4 + \sum_{j=5}^{8} M_j\epsilon_n^j + O[\epsilon_n^8] \tag{38}$$

where M_j's are constants formed by $K_i's$, β and γ. Putting $G(0,0) = 1$, $G^{(1,0)}(0,0) = 0$, $G^{(0,1)}(0,0) = 0$, $G^{(0,2)}(0,0) = -1$, we obtain the following:

$$\epsilon_{n+1} = x_{n+1} - \mu = K_4(\gamma + K_2)^2\Big(\beta K_2 f'(\mu)^2 - K_3\Big)\epsilon_n^8 + O\Big(\epsilon_n^9\Big) \tag{39}$$

Thus, the proof is complete. □

Development of with Memory Methods

We are going to develop with-memory methods from (24) and (34) using the two parameters. From Equations (25) and (35), we clearly see that the order of convergence of the method (34) is sixth and evelenth if $\beta = \dfrac{K_3}{K_2 f'(\mu)}$ and $\gamma = -K_2$. With the choice $\beta = \dfrac{K_3}{K_2 f'(\mu)} = \dfrac{f'''(\mu)}{3f'(\mu)^2 f''(\mu)}$ and $\gamma = -K_2 = -\dfrac{f''(\mu)}{2f'(\mu)}$, the error Equation (25) becomes

$$\epsilon_{n+1} = \frac{(K_2^2 - 2K_3)(2K_2^2K_3 + 3K_3^2 - 2K_2K_4)\epsilon_n^6}{k_2} + O[\epsilon_n^7] \tag{40}$$

and the error Equation (35) becomes

$$\epsilon_{n+1} = \frac{(K_2^2 - 2K_3)^2 K_4(2K_2^2K_3 + 3K_3^2 - 2K_2K_4)\epsilon_n^{11}}{K_2} + O[\epsilon_n^{12}]. \tag{41}$$

In order to obtain with-memory method, we choose $\beta = \beta_n$ and $\gamma = \gamma_n$, as the iteration proceeds by the formulas $\beta_n = \frac{\bar{f}'''(\mu)}{3\bar{f}''(\mu)\bar{f}'(\mu)^2}$ and $\gamma_n = -\frac{\bar{f}''(\mu)}{2\bar{f}'(\mu)}$. In method (24), we use the following approximation

$$\beta_n = \frac{\bar{f}'''(\mu)}{3\bar{f}''(\mu)\bar{f}'(\mu)^2} \approx \frac{N_3'''(x_n)}{3N_3''(x_n)^2 N_3'(x_n)} \tag{42}$$

$$\gamma_n = -\frac{\bar{f}''(\mu)}{2\bar{f}'(\mu)} \approx -\frac{N_4''(w_n)}{2N_4'(w_n)} \tag{43}$$

where $N_3(u) = N_3(u; x_n, y_{n-1}, x_{n-1}, w_{n-1})$ and $N_4(u) = N_4(u; w_n, x_n, y_{n-1}, x_{n-1}, w_{n-1})$ are Newton's interpolating polynomial of third and fourth degrees, respectively. We obtain the following with memory iterative method:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n, w_n] + \gamma_n f(x_n)}, w_n = x_n - \beta_n f(x_n)^2 \\ x_{n+1} &= y_n - Q(t_n)\frac{f(y_n)(1 + \frac{A}{2})}{G(x_n, w_n, y_n)} \end{aligned} \tag{44}$$

For method (34), we use the following approximation

$$\beta_n = \frac{\bar{f}'''(\mu)}{3\bar{f}''(\mu)\bar{f}'(\mu)^2} \approx \frac{N_4'''(x_n)}{3N_4'(x_n)^2 N_4''(x_n)} \tag{45}$$

$$\gamma_n = -\frac{\bar{f}''(\mu)}{2\bar{f}'(\mu)} \approx -\frac{N_5''(w_n)}{2N_5'(w_n)} \tag{46}$$

where $N_4(u) = N_4(u; x_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1})$ and $N_5(u) = N_5(u; w_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1})$ are Newton's interpolating polynomial of fourth and fifth degree, respectively. We obtain the following with-memory iterative method:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n, w_n] + \gamma_n f(x_n)}, w_n = x_n - \beta_n f(x_n)^2 \\ z_n &= y_n - Q(t_n)\frac{f(y_n)(1 + \frac{A}{2})}{F(x_n, w_n, y_n)} \\ x_{n+1} &= z_n - F(r_n, s_n)\frac{f(z_n)(1 + \frac{B}{2})}{H(x_n, w_n, y_n, z_n)} \end{aligned} \tag{47}$$

Remark 2. *Accelerating methods obtained by recursively calculated free parameter may also be called self-accelerating methods. The initial value β_0 and γ_0 should be chosen before starting the iterative process [14].*

We are going to analyse the convergence behaviours of the with-memory methods. If the sequence $\{x_n\}$ converges to the root μ of f with the order p, we write $\epsilon_{n+1} \sim \epsilon_n^p$, where $\epsilon_n = x_n - \mu$. To prove the order of convergence of methods (44) and (47), we use the following lemma, introduced in [15].

Lemma 1. *If* $\beta_n = \frac{N_3'''(x_n)}{3N_3'(x_n)^2 N_3''(x_n)}$ *and* $\gamma_n = -\frac{N_4''(w_n)}{2N_4'(w_n)}$, $n = 1, 2, 3, ...,$ *the estimates*

$$(-K_3 + \beta_n K_2 f'(\mu)^2) \sim \epsilon_{n-1,y}\, \epsilon_{n-1,w}\, \epsilon_{n-1}$$

and

$$K_2 + \gamma_n \sim \epsilon_{n-1,y}\, \epsilon_{n-1,w}\, \epsilon_{n-1}$$

hold.

Let us consider the following theorems.

Theorem 5. *If an initial guess x_0 is sufficiently close to the simple root μ of $f(x) = 0$, f is real sufficiently differentiable function; then, the R-order of convergence of the method (44) is at least 5.7075.*

Proof. Let $\{x_n\}$ be a sequence of approximations generated by the with-memory iterative method defined in (44). If the sequence converges to the root μ of f with order q, we obtain the following:

$$\epsilon_{n+1} \sim \epsilon_n^q, \text{ where } \epsilon_n = x_n - \mu \tag{48}$$

$$\epsilon_{n+1} \sim (\epsilon_{n-1}^q)^q = \epsilon_{n-1}^{q^2} \tag{49}$$

Let us assume that the iterative sequences w_n and y_n have the orders q_1 and q_2, respectively. Then, Equation (48) gives the following:

$$\epsilon_{n,w} \sim (\epsilon_n^{q_1}) = \epsilon_{n-1}^{qq_1} \tag{50}$$

$$\epsilon_{n,y} \sim (\epsilon_n^{q_2}) = \epsilon_{n-1}^{qq_2} \tag{51}$$

By Theorem 3, we can write

$$\epsilon_{n,w} \sim \epsilon_n \tag{52}$$
$$\epsilon_{n,y} \sim (K_2 + \gamma_n)\epsilon_n \tag{53}$$
$$\epsilon_{n+1} \sim (-K_3 + K_2 f'(\mu)^2 \beta_n)(K_2 + \gamma_n)\epsilon_n^4 \tag{54}$$

Using Lemma 1, we obtain the following:

$$\epsilon_{n,w} \sim \epsilon_n \sim \epsilon_{n-1}^q \tag{55}$$
$$\epsilon_{n,y} \sim (K_2 + \gamma_n)\epsilon_n \sim (\epsilon_{n-1,y}\, \epsilon_{n-1,w}\, \epsilon_{n-1})\epsilon_n^2 \sim \epsilon_{n-1}^{2q+q_1+q_2+1} \tag{56}$$
$$\epsilon_{n+1} \sim (-K_3 + K_2 f'(\mu)^2 \beta_n)(K_2 + \gamma_n)\epsilon_n^4 \sim (\epsilon_{n-1,y}\, \epsilon_{n-1,w}\, \epsilon_{n-1})^2 \epsilon_n^4 \sim \epsilon_{n-1}^{4q+2q_1+2q_2+1} \tag{57}$$

Comparing the power of ϵ_{n-1} of Equations (50)–(55), (51)–(56) and (49)–(57), we obtain the following system of equations

$$qq_1 - q = 0 \tag{58}$$
$$qq_1 - 2q - q_1 - q_2 - 1 = 0 \tag{59}$$
$$qq_1 - 4q - 2q_1 - 2q_2 - 2 = 0 \tag{60}$$

By solving this system of equations, we obtain $q_1 = 1$, $q_2 = 2.8507$ and $q = 5.7015$. Thus, the proof is complete. □

Lemma 2. *If* $\beta_n = \dfrac{N_4'''(x_n)}{3N_4'(x_n)^2 N_4''(x_n)}$ *and* $\gamma_n = -\dfrac{N_5''(w_n)}{2N_5'(w_n)}$, $n = 1, 2, 3, \ldots$, *the estimates*

$$(-K_3 + \beta_n K_2 f'(\mu)^2) \sim \epsilon_{n-1,z}\, \epsilon_{n-1,y}\, \epsilon_{n-1,w}\, \epsilon_{n-1}$$

and

$$K_2 + \gamma_n \sim \epsilon_{n-1,z}\, \epsilon_{n-1,y}\, \epsilon_{n-1,w}\, \epsilon_{n-1}$$

hold.

Theorem 6. *If an initial guess x_0 is sufficiently close to the simple root μ of $f(x) = 0$, f is real sufficiently differentiable function; then, the R-order of convergence of the method (47) is at least 11.*

Proof. Let $\{x_n\}$ be a sequence of approximations generated by the with-memory iterative method defined in (44). If the sequence converges to the root μ of f with order q, we obtain the following equation:

$$\epsilon_{n+1} \sim \epsilon_n^q, \ \ where\ \epsilon_n = x_n - \mu \tag{61}$$

$$\epsilon_{n+1} \sim (\epsilon_{n-1}^q)^q = \epsilon_{n-1}^{q^2} \tag{62}$$

Let us assume that the iterative sequences w_n, y_n and z_n have the order q_1, q_2 and q_3, respectively. Then, Equation (61) gives the following:

$$\epsilon_{n,w} \sim (\epsilon_n^{q_1}) = \epsilon_{n-1}^{qq_1} \tag{63}$$

$$\epsilon_{n,y} \sim (\epsilon_n^{q_2}) = \epsilon_{n-1}^{qq_2} \tag{64}$$

$$\epsilon_{n,z} \sim (\epsilon_n^{q_2}) = \epsilon_{n-1}^{qq_3} \tag{65}$$

By Theorem 4, we can write

$$\epsilon_{n,w} \sim \epsilon_n \tag{66}$$

$$\epsilon_{n,y} \sim (K_2 + \gamma_n)\epsilon_n \tag{67}$$

$$\epsilon_{n,z} \sim (-K_3 + K_2 f'(\mu)^2 \beta_n)(K_2 + \gamma_n)\epsilon_n^4 \tag{68}$$

$$\epsilon_{n+1} \sim K_4(\gamma_n + K_2)^2 \Big(\beta_n K_2 f'(\mu)^2 - K_3\Big)\epsilon_n^8 \tag{69}$$

Using Lemma 1, we obtain the following:

$$\epsilon_{n,w} \sim \epsilon_n \sim \epsilon_{n-1}^q \tag{70}$$

$$\epsilon_{n,y} \sim (K_2 + \gamma_n)\epsilon_n \sim (\epsilon_{n-1,z}\, \epsilon_{n-1,y}\, \epsilon_{n-1,w}\, \epsilon_{n-1})\epsilon_n^2 \sim \epsilon_{n-1}^{2q+q_1+q_2+q_3+1} \tag{71}$$

$$\begin{aligned}\epsilon_{n,z} &\sim (-K_3 + K_2 f'(\mu)^2 \beta_n)(K_2 + \gamma_n)\epsilon_n^4 \sim (\epsilon_{n-1,z}\, \epsilon_{n-1,y}\, \epsilon_{n-1,w}\, \epsilon_{n-1})^2\epsilon_n^4 \\ &\sim \epsilon_{n-1}^{4q+2q_1+2q_2+2q_3+2}\end{aligned} \tag{72}$$

$$\begin{aligned}\epsilon_{n+1} &\sim K_4(\gamma_n + K_2)^2 \Big(\beta_n K_2 f'(\mu)^2 - K_3\Big)\epsilon_n^8 \sim (\epsilon_{n-1,z}\, \epsilon_{n-1,y}\, \epsilon_{n-1,w}\, \epsilon_{n-1})^3\epsilon_n^8 \\ &\sim \epsilon_{n-1}^{8q+3q_1+3q_2+3q_3+3}\end{aligned} \tag{73}$$

Comparing the power of ϵ_{n-1} of Equations (63)–(70), (64)–(71), (65)–(72) and (62)–(73), we obtain the following system of equations:

$$qq_1 - q = 0 \tag{74}$$

$$qq_1 - 2q - q_1 - q_2 - q_3 - 1 = 0 \tag{75}$$

$$qq_1 - 4q - 2q_1 - 2q_2 - 2q_3 - 2 = 0 \tag{76}$$

$$qq_1 - 8q - 3q_1 - 3q_2 - 3q_3 - 3 = 0 \tag{77}$$

By solving this system of equations, we obtain $q_1 = 1$, $q_2 = 3$, $q_3 = 6$ and $q = 11$. Thus, the proof is complete. □

4. Numerical Results

In this section, we consider the peculiar attitude of the introduced iterative methods (4) and (16) over the existing methods having the same order of convergence. To demonstrate the behaviours of the newly defined methods, we apply the methods to several numerical examples. For comparison, we consider the following methods:

Fourth-order method (M4th(a)) introduced by Chun et al. [16]:

$$y_n = x_n - \frac{2}{3}\frac{f(x_n)}{f'(x_n)}$$
$$x_{n+1} = x_n - \frac{16f(x_n)f'(x_n)}{-5f'(x_n)^2 + 30f'(x_n)f'(y_n) - 9f'(y_n)} \tag{78}$$

Fourth-order method (M4th(b)) introduced by Singh et al. [17]:

$$y_n = x_n - \frac{2}{3}\frac{f(x_n)}{f'(x_n)}$$
$$x_{n+1} = x_n - \left(\frac{17}{8} - \frac{9f'(y_n)}{4f'(x_n)} + \frac{9}{8}\left(\frac{f'(x_n)}{f'(y_n)}\right)^2\right)\left(\frac{7}{4} - \frac{3}{4}\frac{f'(y_n)}{f'(x_n)}\right)\frac{f(x_n)}{f'(x_n)} \tag{79}$$

In the year 2019, Francisco et al. developed the following method (M4th(c)) [18]:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$
$$x_{n+1} = x_n - \frac{f^2(x_n) + f(x_n)f(y_n) + 2f^2(y_n)}{f(x_n)f'(x_n)} \tag{80}$$

Ekta et al. introduced the following method (M4th(d)) [19] in 2020:

$$y_n = x_n - \frac{2}{3}\frac{f(x_n)}{f'(x_n)}$$
$$x_{n+1} = x_n - \frac{4f(x_n)}{f'(x_n) + 3f'(y_n)}\left(1 + \left(\frac{f(x_n)}{f'(x_n)}\right)^3\right) - \frac{9}{16}\left(\frac{g(x_n)}{f'(x_n)}\right)^2\left(\frac{f(x_n)}{f'(x_n)}\right)^3 \tag{81}$$

where $g(x_n) = \dfrac{f'(x_n)(f'(x_n) - f'(y_n))}{f(x_n)}$
Eighth-order method (M8th(a)) developed by Petkovic et al. [20] is given as follows:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$
$$z_n = x_n - \left(t_n^2 - \frac{f(x_n)}{f(y_n) - f(x_n)}\frac{f(x_n)}{f'(x_n)}\right) \tag{82}$$
$$x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)}\left(\phi(t_n) + \frac{f(z_n)}{f(y_n) - f(z_n)} + \frac{4f(z_n)}{f(x_n)}\right) \tag{83}$$

where $\phi(t_n) = 1 + 2t_n + 2t_n^2 - t_n^3$ with $t_n = \dfrac{f(y_n)}{f(x_n)}$. Cordero A. et al. developed the following eighth-order method (M8th(b)) [21]:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)},$$
$$z_n = y_n - \frac{f(x_n)^2}{(f(x_n) - f(y_n))^2}\frac{f(y_n)}{f'(x_n)}$$
$$x_{n+1} = z_n - (H(t_n, s_n))\frac{f(z_n)}{f'(x_n)}, \tag{84}$$

where $H(t_n, s_n) = 1 + 2t_n + 4t_n^2 + 6t_n^3 + s_n + 4t_n s_n$ with $t_n = \frac{f(y_n)}{f(x_n)}$ and $s_n = \frac{f(z_n)}{f(y_n)}$. Another eighth-order method (M8th(c)) developed by Cordero A. et al. [21] is written as follows:

$$
\begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
z_n &= y_n - \frac{f(x_n)^2}{(f(x_n) - f(y_n))^2} \frac{f(y_n)}{f'(x_n)} \\
x_{n+1} &= z_n - (H(t_n, s_n))G(v_n) \frac{f(z_n)}{f'(x_n)},
\end{aligned}
\tag{85}
$$

where $H(t_n, s_n) = 1 + 2t_n + 4t_n^2 + 6t_n^3 + s_n + 2t_n s_n$ and $G(v_n) = 1 + 2v_n$ with $t_n = \frac{f(y_n)}{f(x_n)}$, $s_n = \frac{f(z_n)}{f(y_n)}$ and $v_n = \frac{f(z_n)}{f(x_n}$. Abbas H. M. et al. developed the following eighth-order method (M8th(d)) [22]:

$$
\begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\
z_n &= x_n + (\beta - 1) \frac{f(x_n) f(f(x_n) - f(y_n))}{f'(x_n)(f(x_n) - 2f(y_n))} \\
&\quad - \beta \left(\frac{f(x_n)}{f'(x_n)} + \frac{f(y_n)\left(f(x_n)^3 + f(y_n)^2 f(x_n) + \frac{1}{2} f(y_n)^3\right)\left(f(x_n + f(y_n))\right)^2}{f'(x_n) f(x_n)^5} \right) \\
x_{n+1} &= z_n - \frac{f(z_n)}{q'(z_n)}.
\end{aligned}
\tag{86}
$$

where $q'(z_n) = a_1 + 2a_2(z - x_n) + 3a_3(z_n - x_n)$,

$$
\begin{aligned}
a_1 &= f'(x_n) \\
a_2 &= \frac{f[y_n, x_n, x_n](z_n - x_n) - f[z_n, x_n, x_n](y_n - x_n)}{z_n - y_n} \\
a_3 &= \frac{f[z_n, x_n, x_n] - f[y_n, x_n, x_n]}{z_n - y_n}
\end{aligned}
$$

The following nonlinear equations are taken as test functions, and their corresponding initial guesses are also given:

Example 1: $f_1(x) = e^{6x} + 0.1441e^{2x} - 2.079e^{4x} - 0.333$, $x_0 = -0.2$
Example 2: $f_2(x) = sin^2 x - x^2 + 1$, $x_0 = 1.4$
Example 3: $f_3(x) = e^{(x^3 - x)} - cos(x^2 - 1) + x^3 + 1$, $x_0 = -1.65$
Example 4: $f_4(x) = sin(-3xcos(\sqrt{x}))$, $x_0 = 0.2$
Example 5: $f_5(x) = x^3 - 3x^2 2^{-x} + 3x 4^{-x} - 8^{-x}$, $x_0 = 0.8$
Example 6: $f_6(x) = (sinx - \frac{\sqrt{2}}{2})(x + 1)$, $x_0 = 0.8$

In Tables 1–6, we provide the errors of two consecutive iterations $\mid x_n - x_{n-1} \mid$ after the fourth iteration; modulus value of approximate root after fourth iteration, i.e., $\mid x_n \mid$ with 17-significance digits; and the residual error, i.e., $|f(x_n)|$ after fourth iteration. We provide the computational order of convergence [23], which is formulated by

$$
COC = \frac{log \mid \frac{f(x_n)}{f(x_{n-1})} \mid}{log \mid \frac{f(x_{n-1})}{f(x_{n-2})} \mid}
\tag{87}
$$

We also provide the CPU running time for each method. The elapsed CPU times are computed by selecting $|f(x_n)| \leq 10^{-1000}$ as the stopping condition. Note that CPU running time is not unique and depends entirely on the computer's specification; however, here, we present an average of three performances to ensure the robustness of the methods. The results are carried out with Mathematica 12.2 software on a 2.30 GHz Intel(R) Core(TM) i3-8145U CPU with 4 GB of RAM running Windows 10.

Remark 3. *For the methods defined in (4) (NPM4th) and (16) (NPM8th), we chose the following weight functions* $Q(t_n) = 1 + 2t_n + \frac{9}{2}t_n^2$ *and* $G(t_n, s_n) = 2t_n + s_n + 5t_n^2 + \frac{5}{2}t_n^3 + \frac{9}{2}t_n^2 s_n + \frac{53}{4}t_n^4$*. For the methods defined in (24) (NPMDF4th) and (34) (NPMDF8th), we chose* $Q(t_n) = 1$ *and* $G(r_n, s_n) = 1 - \frac{1}{2}s^2$*. With-memory methods (44) and (47) are denoted NPMWM1 and NPMWM2, respectively, in the tables.*

From the results in Tables 1–6, we observe that the newly presented methods are highly competitive, with the errors obtained in the different results being highly accurate as compared with the other existing methods and better than them in all cases.

Table 1. Convergence behaviour on f_1.

Methods	$\mid x_n \mid$	$\mid x_n - x_{n-1} \mid$	$\mid f(x_n) \mid$	COC	CPU
Without Memory					
M4th(a)	0.16960654770953905	1.65×10^{-4}	5.51×10^{-13}	3.37	0.535
M4th(b)	0.16960643807598997	6.79×10^{-4}	2.15×10^{-10}	2.90	0.424
M4th(c)	0.16960507315213785	1.18×10^{-3}	2.88×10^{-9}	2.54	0.535
M4th(d)	0.16960625449338888	8.18×10^{-3}	5.75×10^{-10}	2.76	0.465
NPM4th	0.16960654801221716	1.29×10^{-3}	4.11×10^{-14}	3.72	0.422
NPMDF4th	0.16960654801221716	5.08×10^{-23}	1.25×10^{-87}	4	0.404
M8th(a)	0.16960654799121610	4.61×10^{-9}	8.56×10^{-56}	7.86	0.495
M8th(b)	0.16960654799121610	7.51×10^{-10}	2.41×10^{-62}	7.90	0.585
M8th(c)	0.16960654799121610	5.60×10^{-11}	1.73×10^{-71}	7.93	0.497
M8th(d)	0.16960654799121609	1.35×10^{-3}	1.34×10^{-20}	7.40	0.498
NPM8th	0.16960654799121610	1.42×10^{-15}	1.86×10^{-109}	7.94	0.485
NPMDF8th	0.16960654799121610	6.57×10^{-174}	1.92×10^{-1378}	8	0.491
With Memory	$\beta = 0.01$	$\gamma = -1$			
NPMWM1	0.16960654799121610	8.0433×10^{-8}	2.7561×10^{-30}	4.6	0.172
NPMWM2	0.16960654799121610	3.2484×10^{-32}	6.8778×10^{-298}	9.76	0.092

Table 2. Convergence behaviour on f_2.

Methods	$\mid x_n \mid$	$\mid x_n - x_{n-1} \mid$	$\mid f(x_n) \mid$	COC	CPU
Without Memory					
M4th(a)	1.4044916482153412	4.95×10^{-152}	1.31×10^{-605}	4	0.354
M4th(b)	1.4044916482153412	4.68×10^{-148}	1.62×10^{-589}	4	0.256
M4th(c)	1.4044916482153412	4.47×10^{-143}	2.33×10^{-569}	4	0.348
M4th(d)	1.4044916482153412	2.58×10^{-152}	9.25×10^{-607}	4	0.364
NPM4th	1.4044916482153412	2.23×10^{-173}	4.22×10^{-692}	4	0.234
NPMDF4th	1.4044916482153412	1.33×10^{-151}	8.16×10^{-604}	8	0.206
M8th(a)	1.4044916482153412	9.16×10^{-1116}	1.84×10^{-8919}	8	0.254
M8th(b)	1.4044916482153412	7.48×10^{-1130}	2.33×10^{-9032}	8	0.374
M8th(c)	1.4044916482153412	2.12×10^{-1139}	7.07×10^{-9109}	8	0.253
M8th(d)	1.4044916482153412	3.16×10^{-1157}	1.00×10^{-9251}	8	0.254
NPM8th	1.4044916482153412	3.78×10^{-1291}	4.14×10^{-10325}	8	0.136
NPMDF8th	1.4044916482153412	5.34×10^{-1169}	3.18×10^{-9346}	8	0.246
With Memory	$\beta = 0.01$	$\gamma = -1$			
NPMWM1	1.4044916482153412	9.7384×10^{-317}	1.0013×10^{-1631}	5.15	0.024
NPMWM2	1.4044916482153412	2.0364×10^{-2087}	$0.1234 \times 10^{-20000}$	9.8	0.022

Table 3. Convergence behaviour on f_3.

Methods	$\mid x_n \mid$	$\mid x_n - x_{n-1} \mid$	$\lvert f(x_n) \rvert$	**COC**	**CPU**
Without Memory					
M4th(a)	1	4.52×10^{-29}	1.00×10^{-113}	4.00	0.126
M4th(b)	1	7.77×10^{-29}	9.90×10^{-113}	4.00	0.145
M4th(c)	1	3.78×10^{-46}	7.08×10^{-182}	4.00	0.124
M4th(d)	1	9.66×10^{-13}	7.00×10^{-48}	4.00	0.133
NPM4th	1	6.94×10^{-46}	4.35×10^{-183}	4.00	0.156
NPMDF4th	1	6.94×10^{-36}	4.35×10^{-103}	4.00	0.156
M8th(a)	1	9.19×10^{-294}	3.21×10^{-2344}	8.00	0.146
M8th(b)	1	9.35×10^{-285}	5.80×10^{-2272}	8.00	0.138
M8th(c)	1	3.42×10^{-285}	1.50×10^{-2275}	8.00	0.106
M8th(d)	1	2.09×10^{-303}	2.47×10^{-2422}	8.00	0.096
NPM8th	1	4.95×10^{-309}	1.62×10^{-2466}	8.00	0.086
NPMDF8th	1	6.94×10^{-246}	4.35×10^{-1083}	8.00	0.156
With Memory	$\beta = 0.01$	$\gamma = -1$			
NPMWM1	1	1.9552×10^{-22}	2.8585×10^{-112}	5.15	0.029
NPMWM2	1	1.9867×10^{-157}	2.9811×10^{-1508}	9.62	0.050

Table 4. Convergence behaviour on f_4.

Methods	$\mid x_n \mid$	$\mid x_n - x_{n-1} \mid$	$\lvert f(x_n) \rvert$	**COC**	**CPU**
M4th(a)	0	3.20×10^{-35}	2.30×10^{-138}	4	0.309
M4th(b)	0	1.74×10^{-35}	2.35×10^{-139}	4	0.308
M4th(c)	0	3.46×10^{-28}	5.87×10^{-110}	4	0.496
M4th(d)	0	6.96×10^{-28}	1.40×10^{-108}	4	0.336
NPM4th	0	4.82×10^{-36}	1.18×10^{-141}	4	0.203
NPMDF4th	0	8.84×10^{-33}	8.39×10^{-128}	4	0.291
M8th(a)	0	2.82×10^{-148}	5.84×10^{-1180}	8	0.226
M8th(b)	0	2.52×10^{-176}	4.20×10^{-1404}	8	0.336
M8th(c)	0	1.39×10^{-184}	2.76×10^{-1470}	8	0.406
M8th(d)	0	2.53×10^{-195}	3.06×10^{-1557}	8	0.386
NPM8th	0	9.63×10^{-213}	8.01×10^{-1696}	8	0.276
NPMDF8th	0	5.56×10^{-230}	4.70×10^{-1834}	8	0.323
With Memory	$\beta = 0.01$	$\gamma = -1$			
NPMWM1	0	12.0284×10^{-62}	1.4865×10^{-318}	5.16	0.035
NPMWM2	0	1.0723×10^{-434}	2.9811×10^{-4000}	9.62	0.043

Table 5. Convergence behaviour on f_5.

Methods	$\mid x_n \mid$	$\mid x_n - x_{n-1} \mid$	$\lvert f(x_n) \rvert$	**COC**	**CPU**
Without Memory					
M4th(a)	0.94679089869251303	1.10×10^{-16}	1.19×10^{-63}	4	0.676
M4th(b)	0.94679089869251303	1.61×10^{-14}	8.70×10^{-55}	4	0.587
M4th(c)	0.94679089869251303	1.21×10^{-23}	4.96×10^{-95}	4	0.596
M4th(d)	0.94679089869251303	2.44×10^{-28}	5.58×10^{-110}	4	0.477
NPM4th	0.94679089869251303	1.93×10^{-49}	2.50×10^{-195}	4	0.406
NPMDF4th	0.94679089869251303	2.81×10^{-50}	2.44×10^{-198}	4	0.450
M8th(a)	Divergence				
M8th(b)	0.94679089869251303	3.21×10^{-304}	6.99×10^{-2426}	8	0.636
M8th(c)	0.94679089869251303	9.50×10^{-286}	3.03×10^{-2278}	8	0.646
M8th(d)	0.94679089869251303	1.39×10^{-300}	4.44×10^{-2397}	8	0.597
NPM8th	0.94679089869251303	1.12×10^{-309}	4.48×10^{-2471}	8	0.424
NPMDF8th	0.94679089869251303	1.80×10^{-255}	9.51×10^{-2038}	8	0.571
With Memory	$\beta = 0.01$	$\gamma = -1$			
NPMWM1	0.94679089869251303	12.0284×10^{-62}	1.4865×10^{-318}	5.16	0.034
NPMWM2	0.94679089869251303	1.0723×10^{-434}	2.9811×10^{-4000}	9.62	0.024

Table 6. Convergence behaviour on f_6.

Methods	$\mid x_n \mid$	$\mid x_n - x_{n-1} \mid$	$\lvert f(x_n) \rvert$	COC	CPU
Without Memory					
M4th(a)	0.78539816339744831	5.62×10^{-143}	2.72×10^{-571}	4	0.386
M4th(b)	0.78539816339744831	6.37×10^{-143}	4.53×10^{-571}	4	0.276
M4th(c)	0.78539816339744831	1.23×10^{-152}	8.13×10^{-610}	4	0.256
M4th(d)	0.78539816339744831	2.23×10^{-118}	3.05×10^{-471}	4	0.266
NPM4th	0.78539816339744831	6.85×10^{-153}	7.50×10^{-611}	4	0.126
NPMDF4th	0.78539816339744831	1.53×10^{-124}	3.60×10^{-496}	4	0.185
M8th(a)	0.78539816339744831	6.32×10^{-1110}	1.19×10^{-8877}	8	0.136
M8th(b)	0.78539816339744831	3.07×10^{-1110}	5.99×10^{-8879}	8	0.256
M8th(c)	0.78539816339744831	6.51×10^{-1111}	2.40×10^{-8884}	8	0.166
M8th(d)	0.78539816339744831	4.56×10^{-1113}	9.48×10^{-8904}	8	0.276
NPM8th	0.78539816339744831	2.30×10^{-1113}	5.77×10^{-8904}	8	0.126
NPMDF8th	0.78539816339744831	1.31×10^{-1056}	3.52×10^{-8449}	8	0.242
With Memory	$\beta = 0.01$	$\gamma = -1$			
NPMWM1	0.78539816339744831	1.0051×10^{-210}	6.6067×10^{-1085}	5.16	0.013
NPMWM2	0.78539816339744831	1.6485×10^{-1566}	$5.2790 \times 10^{-15073}$	9.62	0.024

Applications on Real-World Problem

Here, we take some real-world problems from other papers:

Problem 1. *Projectile Motion Problem: This problem expresses motion of the projectile, it is represented by the following nonlinear equation (see more details in [7])*

$$f(x) = h + \frac{v^2}{2g} - \frac{gx^2}{2v^2} - w(x) \tag{88}$$

where h is height of the tower from which the projectile is launched, v is initial velocity of the projectile, g is acceleration due to gravity and $w(x)$ is the impact function. In particular, we choose $w(x) = 0.4x$, $h = 10\ m$, $v = 20\ m/s$, $g = 9.8\ m/s^2$ and $x_0 = 30$.

Table 7 shows that the convergence behaviour of newly introduced methods performs better than that of the other existing methods.

Table 7. Convergence behaviour on projectile motion problem.

Methods	$\mid x_n \mid$	$\mid x_n - x_{n-1} \mid$	$\lvert f(x_n) \rvert$	COC	CPU
Without Memory					
M4th(a)	14.614565956915786	5.18×10^{-27}	1.55×10^{-109}	4	0.086
M4th(b)	14.614565956915786	3.74×10^{-24}	6.34×10^{-98}	4	0.068
M4th(c)	14.614565956915786	8.85×10^{-22}	3.30×10^{-88}	4	0.076
M4th(d)	Divergence				
NPM4th	14.614565956915786	1.98×10^{-40}	1.25×10^{-203}	5	0.046
NPMDF4th	14.614565956915786	1.47×10^{-26}	7.49×10^{-104}	4.00	0.021
M8th(a)	14.614565956915786	1.02×10^{-162}	1.01×10^{-1304}	8	0.096
M8th(b)	14.614565956915786	8.86×10^{-169}	2.21×10^{-1353}	8	0.066
M8th(c)	14.614565956915786	9.14×10^{-177}	2.10×10^{-1417}	8	0.038
M8th(d)	14.614565956915786	2.86×10^{-188}	1.05×10^{-1509}	8	0.056
NPM8th	14.614565956915786	3.67×10^{-262}	1.09×10^{-2364}	9	0.026
NPMDF8th	14.614565956915786	1.50×10^{-77}	6.71×10^{-693}	9.00	0.041
With Memory	$\beta = 0.01$	$\gamma = -1$			
NPMWM1	14.614713726401837	1.0051×10^{-21}	6.6067×10^{-105}	5.16	0.034
NPMWM2	14.614713726401837	1.6485×10^{-156}	5.2790×10^{-1503}	9.62	0.023

Problem 2. *Height of a moving object: An object falling vertically through the air is subjected to viscous resistance as well as the force of gravity (see [24] Ch2, p-66). Let us assume that the*

object with mass m is dropped from a height s_0 and that the height of the object after t seconds is represented by the following equation:

$$s(t) = s_0 - \frac{mg}{k}t + \frac{m^2 g}{k^2}(1 - e^{-\frac{kt}{m}}) \tag{89}$$

where k represents coefficient of air resistance in lb-s/ft and g is the acceleration due to gravity. To solve Equation (89), we choose $s_0 = 300\ ft$, $m = 0.25\ lb$ and $k = 0.1$ lb-s/ft. We have to find the time taken for the object to reach the ground. We rewrite Equation (89) in the following nonlinear form

$$f(x) = 300 - 80.425x + 2.010625\left(1 - e^{-\dfrac{x}{2.5}}\right),\ x_0 = 3 \tag{90}$$

Table 8 shows that the convergence behaviour of newly introduced methods performs better than that of the other existing methods.

Table 8. Convergence behaviour on height of a moving object problem.

Methods	$\mid x_n \mid$	$\mid x_n - x_{n-1} \mid$	$\lvert f(x_n) \rvert$	COC	CPU
Without Memory					
M4th(a)	3.7496042636030085	6.06×10^{-137}	7.44×10^{-550}	4	0.326
M4th(b)	3.7496042636030085	6.08×10^{-137}	7.55×10^{-550}	4	0.260
M4th(c)	3.7496042636030085	3.91×10^{-165}	5.10×10^{-664}	4	0.233
M4th(d)	3.7496042636030085	9.97×10^{-9}	7.92×10^{-31}	4	0.186
NPM4th	3.7496042636030085	2.55×10^{-165}	9.14×10^{-665}	4	0.196
NPMDF4th	3.7496042636030085	2.55×10^{-105}	9.14×10^{-605}	4	0.196
M8th(a)	3.7496042636030085	4.04×10^{-1192}	4.35×10^{-9546}	8	0.206
M8th(b)	3.7496042636030085	1.00×10^{-1196}	1.42×10^{-9582}	8	0.232
M8th(c)	3.7496042636030085	5.23×10^{-1197}	7.71×10^{-9585}	8	0.226
M8th(d)	3.7496042636030085	1.88×10^{-1193}	8.96×10^{-9557}	8	0.196
NPM8th	3.7496042636030085	1.40×10^{-1196}	2.04×10^{-9586}	8	0.167
NPMDF8th	3.7496042636030085	1.40×10^{-1194}	2.04×10^{-9580}	8	0.167
With Memory	$\beta = 0.01$	$\gamma = -1$			
NPMWM1	3.7496042636030085	1.2464×10^{-19}	1.1482×10^{-102}	5.16	0.023
NPMWM2	3.7496042636030085	9.8794×10^{-60}	5.7594×10^{-583}	9.71	0.026

Problem 3. *Fractional Conversion: Fractional conversion of nitrogen hydrogen feed to ammonia at 500 °C temperature and 250 atm. pressure is given by the following nonlinear equation (see [25,26]):*

$$f(x) = -0.186 - \frac{8x^2(x-4)^2}{4(x-2)^3} \tag{91}$$

Equation (91) can be reduced to a polynomial of degree four

$$f(x) = x^4 - 7.79075x^3 + 14.7445x^2 + 2.511x - 0.164,\ x_0 = 0.22 \tag{92}$$

Table 9 shows that the convergence behaviour of the newly introduced methods performs better than that of the other existing methods.

Table 9. Convergence behaviour on fractional conversion problem.

Methods	$\mid x_n \mid$	$\mid x_n - x_{n-1} \mid$	$\lvert f(x_n)\rvert$	COC	CPU
Without Memory					
M4th(a)	0.27775954284172066	6.76×10^{-68}	4.79×10^{-268}	4	0.023
M4th(b)	0.27775954284172066	3.95×10^{-65}	7.59×10^{-257}	4	0.020
M4th(c)	0.27775954284172066	3.22×10^{-60}	5.16×10^{-237}	4	0.019
M4th(d)	0.27775954284172066	6.90×10^{-64}	6.90×10^{-252}	4	0.023
NPM4th	0.27775954284172066	2.62×10^{-82}	3.09×10^{-326}	4	0.019
NPMDF4th	0.27775954284172066	1.88×10^{-10}	2.70×10^{-38}	4.14	0.029
M8th(a)	0.27775954284172066	1.42×10^{-447}	1.80×10^{-3572}	8	0.036
M8th(b)	0.27775954284172066	4.61×10^{-454}	1.82×10^{-3624}	8	0.036
M8th(c)	0.27775954284172066	8.78×10^{-463}	2.38×10^{-3694}	8	0.036
M8th(d)	0.27775954284172066	1.14×10^{-360}	3.29×10^{-2870}	8	0.037
NPM8th	0.27775954284172066	3.88×10^{-495}	1.26×10^{-3953}	8	0.035
NPMDF8th	0.27775954284172066	2.13×10^{-15}	1.32×10^{-115}	8.14	0.046
With Memory	$\beta = 0.01$	$\gamma = -1$			
NPMWM1	0.27775954284172066	8.6899×10^{-157}	1.3916×10^{-810}	5.16	0.027
NPMWM2	0.27775954284172066	3.1158×10^{-1278}	$1.2697 \times 10^{-14052}$	11.0	0.032

Problem 4. *Open channel flow: Open channel flow is a problem to find the depth of water in a rectangular channel for a given quantity of water; the problem is represented by the following nonlinear equation (see [25,27]):*

$$f(x) = \frac{\sqrt{s}bx}{n}\left(\frac{bx}{b+2x}\right)^{\frac{2}{3}} - F \tag{93}$$

where F represents water flow, which is formulated as $F = \frac{\sqrt{s}bx}{n} r^{\frac{2}{3}}$. *s is the slope of the channel, a is the area the channel, r is the hydraulic radius of the channel, n is Manning's roughness coefficient and b is the width of the channel. Taking the different values of the parameters as* $F = 14.15\ m^3/s$ *b = 4.572 m, s = 0.017 and n = 0.0015, we obtain the following equation*

$$f(x) = \frac{0.5961x}{0.0015}\left(\frac{4.572x}{4.572+2x}\right)^{\frac{2}{3}} - 14.15, \quad x_0 = 0.4 \tag{94}$$

Table 10 shows that convergence behaviour of newly introduced method performs better than that of the other existing methods.

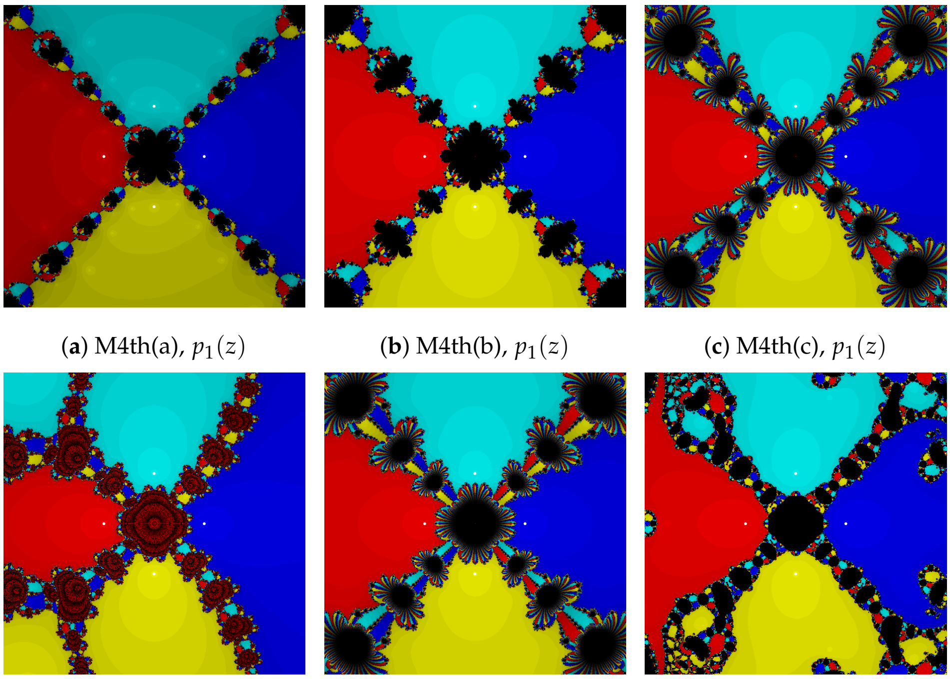
Table 10. Convergence behaviour on open channel flow problem.

Methods	$\mid x_n \mid$	$\mid x_n - x_{n-1} \mid$	$\lvert f(x_n)\rvert$	COC	CPU
Without Memory					
M4th(a)	0.13839748098511792	2.11×10^{-27}	9.06×10^{-104}	4	0.151
M4th(b)	0.13839748098511792	1.66×10^{-25}	4.74×10^{-96}	4	0.066
m4th(c)	0.13839748098511792	9.11×10^{-24}	6.68×10^{-89}	4	0.055
M4th(d)	0.13839748098511792	2.58×10^{-26}	3.50×10^{-99}	4	0.062
NPM4th	0.13839748098511792	1.84×10^{-31}	1.09×10^{-120}	4	0.051
NPMDF4th	0.13839748098511792	1.88×10^{-30}	2.70×10^{-138}	4.14	0.029
M8th(a)	0.13839748098511792	7.45×10^{-164}	5.03×10^{-1299}	8	0.066
M8th(b)	0.13839748098511792	1.14×10^{-165}	1.20×10^{-1313}	8	0.060
M8th(c)	0.13839748098511792	3.85×10^{-172}	1.50×10^{-1365}	8	0.063
M8th(d)	0.13839748098511792	7.04×10^{-190}	8.13×10^{-1508}	8	0.061
NPM8th	0.13839748098511792	3.49×10^{-197}	3.29×10^{-1567}	8	0.060
NPMDF8th	0.13839748098511792	2.13×10^{-156}	1.32×10^{-1515}	8.14	0.046
With Memory	$\beta = 0.01$	$\gamma = -1$			
NPMWM1	0.13839748098511792	8.7820×10^{-25}	4.0868×10^{-121}	5.16	0.029
NPMWM2	0.13839748098511792	2.5875×10^{-65}	8.9327×10^{-618}	9.62	0.034

5. Basins of Attraction

In this section, we discuss the dynamical behaviours of the without-memory iterative methods in the complex plane. This gives useful information about the stability and reliability of the iterative methods. Here, we compare the stability of the introduced methods with other methods. For the comparison, we apply the iterative methods to the complex polynomial of orders four and three, $p_1(z) = z^4 - 1$ and $p_2(z) = z^3 + z$. We take a square $D = [-3, 3] \times [-3, 3] \in \mathbb{C}$ of 601×601 grid points and lay on a colour to each point $z \in D$, according to the roots corresponding to which the method starting from z converges. The roots of the polynomial are represented by the white dots. We spot the point z—where the methods diverge from a root with the tolerance 10^{-4} and a maximum iteration 100—as black, and these black points are considered as divergent points. In the basins of attraction of each iterative method, a brighter colour region indicates that the iterative method converges to the root in the minimum number of iterations and a darker region indicates that the method needs more iterations to converge towards the root.

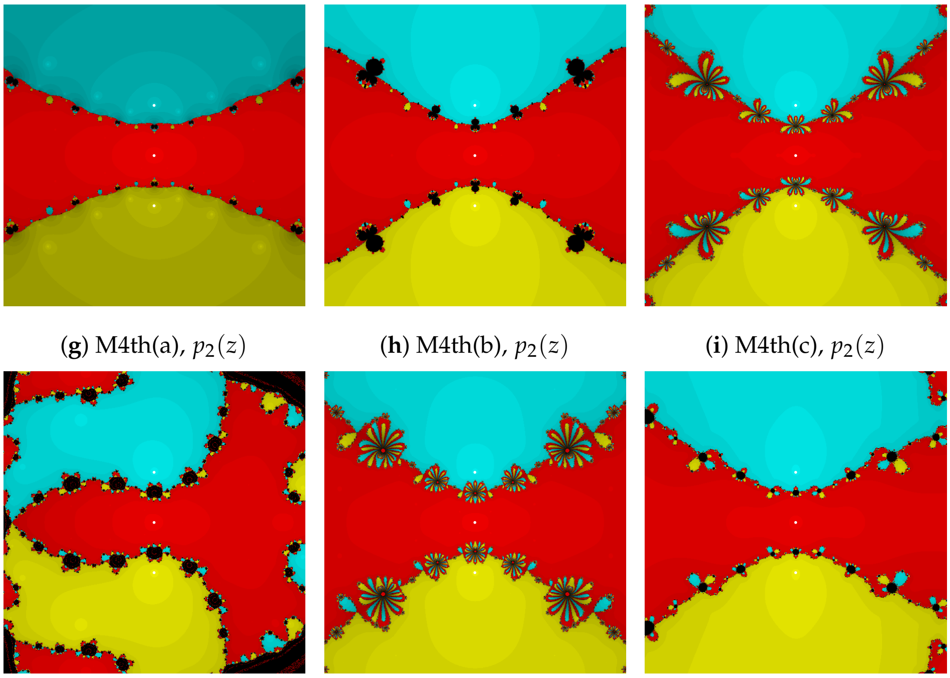
The basins of attraction of fourth-order iterative methods on polynomials $p_1(z)$ and $p_2(z)$ are given in Figure 1. Figures 2 and 3 are the basins of attraction of eighth-order iterative methods on polynomials $p_1(z)$ and $p_2(z)$, respectively. From the figures, we can observe that the newly presented methods produce competitive basins and perform better than the other methods in some cases.



(**a**) M4th(a), $p_1(z)$ (**b**) M4th(b), $p_1(z)$ (**c**) M4th(c), $p_1(z)$

(**d**) M4th(d), $p_1(z)$ (**e**) NPM4th, $p_1(z)$ (**f**) NPMDF4th, $p_1(z)$

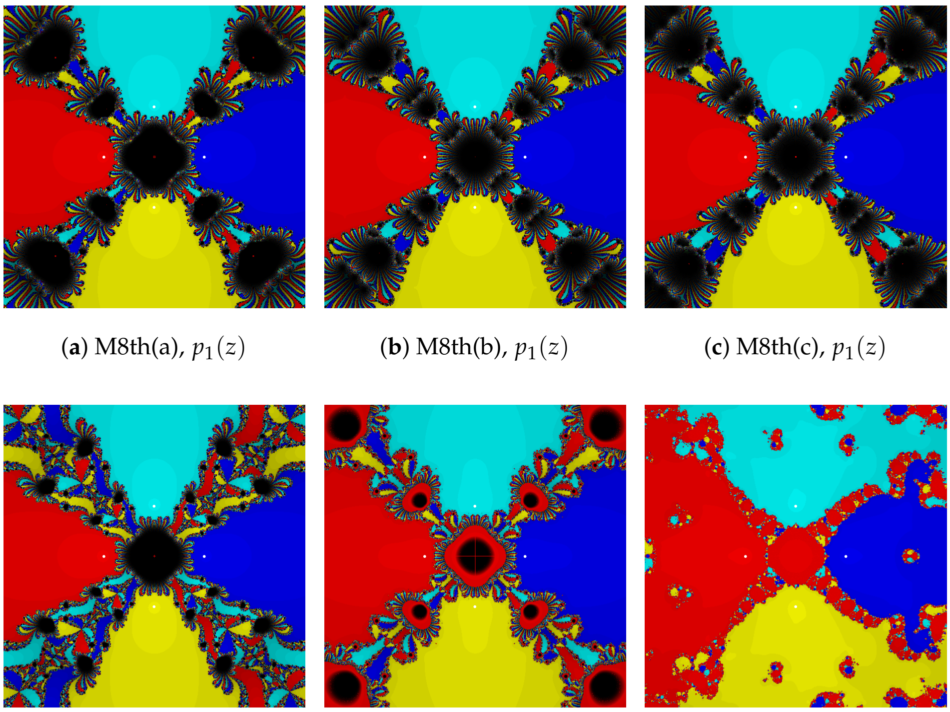
Figure 1. *Cont.*



(**g**) M4th(a), $p_2(z)$ (**h**) M4th(b), $p_2(z)$ (**i**) M4th(c), $p_2(z)$

(**j**) M4th(d), $p_2(z)$ (**k**) NPM4th, $p_2(z)$ (**l**) NPMDF4th, $p_2(z)$

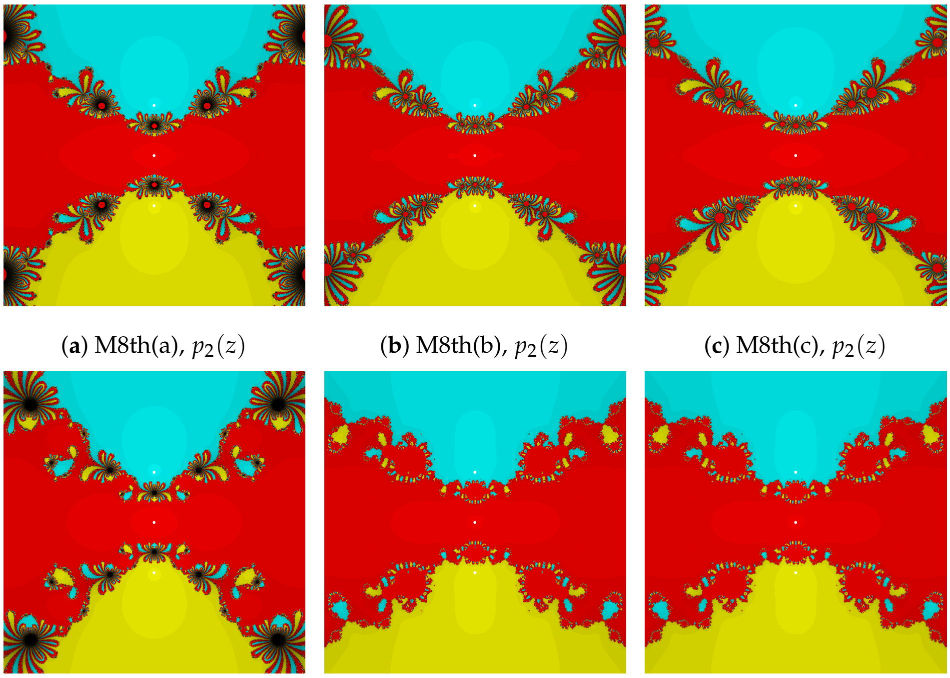
Figure 1. Basins of attraction for fourth-order methods for $p_1(z)$ and $p_2(z)$.



(**a**) M8th(a), $p_1(z)$ (**b**) M8th(b), $p_1(z)$ (**c**) M8th(c), $p_1(z)$

(**d**) M8th(d), $p_1(z)$ (**e**) NPM8th, $p_1(z)$ (**f**) NPMDF8th, $p_1(z)$

Figure 2. Basins of attraction for eighth-order methods for $p_1(z)$.



(**a**) M8th(a), $p_2(z)$ (**b**) M8th(b), $p_2(z)$ (**c**) M8th(c), $p_2(z)$

(**d**) M8th(d), $p_2(z)$ (**e**) NPM8th, $p_2(z)$ (**f**) NPMDF8th, $p_2(z)$

Figure 3. Basins of attraction for eighth-order methods for $p_2(z)$.

6. Conclusions

We have introduced the fourth and eighth-order without-memory iterative methods and with-memory methods of orders 5.7 and 11. The weight function and divided difference techniques are used to develop the without-memory methods. The derivative-free with-memory iterative methods are developed using two accelerating parameters, which are computed using Newton interpolating polynomials, thereby increasing the order of convergence from 4 to 5.7 for two-step and from 8 to 11 for three-step methods without any additional function evaluation. The presented methods are compared with other existing methods using some examples of nonlinear equations. The results given in the tables clarify the competitive nature of the presented methods in comparison with the existing methods and will be valuable in finding an adequate estimate of the exact solution of nonlinear equations. The current work can be extended to find solutions of multivariate nonlinear equations.

Author Contributions: Conceptualization, W.H.C., G.T. and S.P.; methodology, W.H.C. and S.P.; software, W.H.C. and G.T.; validation,W.H.C., G.T. and S.P.; formal analysis, S.P.; investigation, W.H.C., S.P. and G.T.; resources, W.H.C.; data curation, W.H.C. and G.T.; writing—original draft preparation, W.H.C.; writing—review and editing, W.H.C., G.T. and S.P.; visualization, G.T.; supervision, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Jamaludin, N.A.A.; Nik Long, N.M.A.; Salimi, M.; Sharifi, S. Review of Some Iterative Methods for Solving Nonlinear Equations with Multiple Zeros. *Afrika Matematika* **2019**, *30*, 355–369. [CrossRef]
2. Traub, J.F. *Iterative Methods for the Solution of Equations*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1964.
3. Ostrowski, A.M. *Solution of Equations in Euclidean and Banach Space*; Academic Press: New York, NY, USA, 1973.
4. Panday, S.; Sharma, A.; Thangkhenpau, G. Optimal fourth and eighth-order iterative methods for non-linear equations. *J. Appl. Math. Comput.* **2022**, 1–19. [CrossRef]

5. Kumar, M.; Singh, A.K.; Srivastava, A. A New Fifth Order Derivative Free Newton-Type Method for Solving Nonlinear Equations. *Appl. Math. Inf. Sci.* **2015**, *9*, 1507–1513.
6. Choubey, N.; Jaiswal, J.P. A Derivative-Free Method of Eighth-Order For Finding Simple Root of Nonlinear Equations. *Commun. Numer. Anal.* **2015**, *2*, 90–103. [CrossRef]
7. Tao, Y.; Madhu, K. Optimal fourth, eighth and sixteenth Order Methods by using Divided Difference Techniques and Their Basins of Attractions and Its Applications. *Mathematics* **2019**, *7*, 322. [CrossRef]
8. Neta, B. A Derivative-Free Method to Solve Nonlinear Equations. *Mathematics* **2021**, *9*, 583. [CrossRef]
9. Singh, M.K.; Singh, A.K. The Optimal Order Newton's Like Methods with Dynamics. *Mathematics* **2021**, *9*, 527. [CrossRef]
10. Solaiman, O.S.; Hashim, I. Optimal Eighth-Order Solver for Nonlinear Equations with Applications in Chemical Engineering. *Intell. Autom. Soft Comput.* **2021**, *13*, 87–93. [CrossRef]
11. Chanu, W.H.; Panday, S. Excellent Higher Order Iterative Scheme for Solving Non-linear Equations. *IAENG Int. J. Appl. Math.* **2022**, *52*, 1–7.
12. Bawazir, H.M. Seventh and Twelfth-Order Iterative Methods for Roots of Nonlinear Equations. *Hadhramout Univ. J. Nat. Appl. Sci.* **2021**, *18*, 2.
13. Torkashvand, V.; Kazemi, M.; Moccari, M. Sturcture A Family of Three-Step With-Memory Methods for Solving nonlinear Equations and Their Dynamics. *Math. Anal. Convex Optim.* **2021**, *2*, 119–137.
14. Lotfi, T.; Soleymani, F.; Noori, Z.; Kılıçman, A.; Khaksar Haghani, F. Efficient Iterative Methods with and without Memory Possessing High Efficiency Indices. *Discret. Dyn. Nat. Soc.* **2014**, *2014*, 912796. [CrossRef]
15. Dzunic, J. On Efficient Two-Parameter Methods for Solving Nonlinear Equations. *Numer. Algorithms* **2012**, *63*, 549–569. [CrossRef]
16. Chun, C.; Lee, M.Y.; Neta, B.; Dzunic, J. On optimal fourth-order iterative methods free from second derivative and their dynamics. *Appl. Math. Comput.* **2012**, *218*, 6427–6438. [CrossRef]
17. Singh, A.; Jaiswal, J.P. Several new third-order and fourth-order iterative methods for solving nonlinear equations. *Int. J. Eng. Math.* **2014**, *2014*, 828409. [CrossRef]
18. Chicharro, F.I.; Cordero, A.; Garrido, N.; Torregrosa, J.R. Wide stability in a new family of optimal fourth-orderiterative methods. *Comput. Math. Methods* **2019**, *1*, e1023. [CrossRef]
19. Sharma, E.; Panday, S.; Dwivedi, M. New Optimal Fourth Order Iterative Method for Solving Nonlinear Equations. *Int. J. Emerg. Technol.* **2020**, *11*, 755–758.
20. Petkovic, M.S.; Neta, B.; Petkovic, L.D.; Dzunic, J. *Multipoint Methods for Solving Nonlinear Equations*; Elsevier: Amsterdam, The Netherlands, 2012.
21. Cordero, A.; Lotfi, T.; Mahdiani, K.; Torregrosa, J.R. Two Optimal General Classes of Iterative Methods with Eighth-order. *Acta. Appl. Math.* **2014**, *134*, 64–74. [CrossRef]
22. Abbas, H.M.; Al-Subaihi, I.A. A New Family of Optimal Eighth-Order Iterative Method for Solving Nonlinear Equations. *Appl. Math. Comput.* **2022**, *8*, 10–17.
23. Weerakoon, S.; Fernando, T.G.I. A variant of Newton's method with accelerated third convergence. *Appl. Math. Lett.* **2000** *13*, 87–93. [CrossRef]
24. Richard, L.B.; Douglas, J.F. *Numerical Analysis*, 9th ed.; India Brooks/Cole Cengage Larning: Boston, MA, USA, 2019.
25. Rehman, M.A.; Naseem, A.; Abdeljawad, T. Some Novel Sixth-Order Schemes for Computing Zeros of Nonlinear Scalar Equations and Their apllications in Engineering. *J. Funct. Spaces* **2021**, *2021*, 5566379.
26. Balaji, G.V.; Seader, J.D. Application of interval Newton's method to chemical engineering problems. *Reliab. Comput.* **1995**, *1*, 215–223. [CrossRef]
27. Manning, R. On the flow of water in open channels and pipes. *Trans. Inst. Civ. Eng. Irel.* **1891**, *20*, 161–207.