mathematics

MDPI

Article

A Novel *n*-Point Newton-Type Root-Finding Method of High Computational Efficiency

Xiaofeng Wang

School of Mathematical Sciences, Bohai University, Jinzhou 121000, China; wangxiaofeng@qymail.bhu.edu.cn

Abstract: A novel Newton-type n-point iterative method with memory is proposed for solving nonlinear equations, which is constructed by the Hermite interpolation. The proposed iterative method with memory reaches the order $(2^n + 2^{n-1} - 1 + \sqrt{2^{2n+1} + 2^{2n-2} + 2^n + 1})/2$ by using n variable parameters. The computational efficiency of the proposed method is higher than that of the existing Newton-type methods with and without memory. To observe the stability of the proposed method, some complex functions are considered under basins of attraction. Basins of attraction show that the proposed method has better stability and requires a lesser number of iterations than various well-known methods. The numerical results support the theoretical results.

Keywords: nonlinear equation; iterative methods; optimal convergence order; stability

MSC: 65H05; 65B99

1. Introduction

Finding a solution a of nonlinear equation $f(x) = 0$, where $f : D \subseteq \mathbb{R} \to \mathbb{R}$ is a sufficiently differentiable function in an open set D, is a difficult problem in the field of numerical analysis. Multipoint iterative methods with high computational efficiency were introduced by Traub [1] and Petković [2], which are very suitable for finding the solution of nonlinear equations. The design methods for multipoint iterative method include: the weight function method [3–5], the interpolation method [6,7], the rational function method [8,9], the undetermined coefficient method [10], the inverse interpolation function method [11,12] and the symbolic computation method [13]. Using these methods, many efficient multipoint iterative methods have been proposed for solving nonlinear equations, see [14–20] and the references therein. In those methods, the n-point iterative method is worth studying because of its high computational efficiency. The Kung–Traub's method [14], Zheng's method [15], Petković–Džunić's method [16] and Wang–Zhang's method [17] are well-known derivative-free n-point iterative methods. Furthermore, some efficient n-point Newton-type iterative methods with and without memory have been proposed. Kung and Traub [14] proposed an optimal 2^nth order Newton-type method as follows:

$$\begin{cases} y_{k,1} = t_k - f(t_k)/f'(t_k), \\ y_{k,j} = S_j(0), j = 2, \ldots, n, \\ t_{k+1} = y_{k,n}, \end{cases} \tag{1}$$

where the inverse interpolating polynomial is $S_j(y)$ such that $S_j(f(t_k)) = t_k, S_j'(f(t_k)) = 1/f'(t_k)$, $S_j(f(y_{k,j})) = y_{k,j}, (j = 2, \ldots, n)$. Petković [18] derived the following n-point Newton-type method:

check for updates

Citation: Wang, X. A Novel n-Point Newton-Type Root-Finding Method of High Computational Efficiency. *Mathematics* **2022**, *10*, 1144. https://doi.org/10.3390/math10071144

Academic Editors: Alicia Cordero Barbero and Juan R. Torregrosa

Received: 10 March 2022
Accepted: 31 March 2022
Published: 2 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

$$\begin{cases} \phi_1(t) = t - \frac{f(t)}{f'(t)}, \\ \phi_2(t) = \psi_f(t), \\ \phi_3(t) = \phi_2(t) - \frac{f(\phi_2(t))}{h'_{(2)}(\phi_2(t))}, \\ \cdots, \\ \phi_n(t) = \phi_{n-1}(t) - \frac{f(\phi_{n-1}(t))}{h'_{(n-1)}(\phi_{(n-1)}(t))}, \end{cases} \tag{2}$$

where $h'_i(t), (i = 2, 3, \cdots, n-1.)$ is the Hermite interpolation polynomial satisfying the conditions $f'(\phi_0) = h'_{m+1}(\phi_0), f(\phi_j) = h_{m+1}(\phi_j)$ *and* $(j = 0, 1, \cdots, n-1.)$, and $\psi_f(t)$ is an iterative function with an optimal order of 4. Cordero et al. [19] studied the stability of method (2) for $n = 2, 3$. In [20], we obtained the following one-parameter n-point Newton-type method without memory:

$$\begin{cases} t_{k,1} = t_{k,0} - \frac{f(t_{k,0})}{f'(t_{k,0}) + Lf(t_{k,0})}, \\ t_{k,2} = t_{k,1} - \frac{f(t_{k,1})}{f[t_{k,1}, t_{k,0}] + f[t_{k,1}, t_{k,0}, t_{k,0}](t_{k,1} - t_{k,0})}, \\ \cdots, \\ t_{k,n} = t_{k,n-1} - \frac{f(t_{k,n-1})}{N(t_{k,n-1}, t_{k,n-2}, \cdots, t_{k,1}, t_{k,0})}, \end{cases} \tag{3}$$

where $N(t_{k,n-1}, t_{k,n-2}, \cdots, t_{k,1}, t_{k,0}) = f[t_{k,n-1}, t_{k,n-2}] + \cdots + f[t_{k,n-1}, t_{k,n-2}, \cdots, t_{k,1}, t_{k,0}, t_{k,0}](t_{k,n-1} - t_{k,n-2}) \cdots (t_{k,n-1} - t_{k,0}), t_{k,0} = t_k$, and $L \in \mathbb{R}$ is a constant. By replacing parameter L in (3) with a variable parameter L_k, method (3) can be transformed to the following one-parameter n-point Newton-type iterative method with memory:

$$\begin{cases} t_{k,1} = t_{k,0} - \frac{f(t_{k,0})}{f'(t_{k,0}) + L_k f(t_{k,0})}, \\ t_{k,2} = t_{k,1} - \frac{f(t_{k,1})}{f[t_{k,1}, t_{k,0}] + f[t_{k,1}, t_{k,0}, t_{k,0}](t_{k,1} - t_{k,0})}, \\ \cdots, \\ t_{k,n} = t_{k,n-1} - \frac{f(t_{k,n-1})}{N(t_{k,n-1}, t_{k,n-2}, \cdots, t_{k,1}, t_{k,0})}, \end{cases} \tag{4}$$

where $N(t_{k,n-1}, t_{k,n-2}, \cdots, t_{k,1}, t_{k,0}) = f[t_{k,n-1}, t_{k,n-2}] + \cdots + f[t_{k,n-1}, t_{k,n-2}, \cdots, t_{k,1}, t_{k,0}, t_{k,0}](t_{k,n-1} - t_{k,n-2}) \cdots (t_{k,n-1} - t_{k,0}), t_{k,0} = t_k$, $L_k = -H''_4(y_{k,0})/(2f'(y_{k,0}))$ and $H_4(x)$ is a Hermite's interpolating polynomial with a degree of 4. Method (4) improved the convergence order of method (3) without any additional functional evaluations, which implied that the variable parameter can improve the computational efficiency of the Newton-type iterative method. Thus, we concluded that the convergence order of n-point Newton-type iterative method can be improved further by improving the number of variable parameters in the iterative scheme.

The aims of this work are to improve the computational efficiency and convergence order of n-point Newton-type method and produce a general n-point Newton-type iterative method with memory for solving nonlinear equations. The paper is organized as follows: In Section 2, we propose a general n-point Newton-type iterative method with optimal order 2^n by using Hermite's interpolation polynomial. Based on the optimal n-point Newton-type iterative method, a general n-point Newton-type iterative method with memory is proposed by using n variable parameters in Section 3. The convergence order of the n-point Newton-type iterative method with memory is analyzed, which is higher than that of the existing Newton-type iterative methods. In Section 4, the stability of the presented iterative method is analyzed with the help of the basins of attraction. Several numerical tests are made to confirm the theoretical results in Section 5. Conclusions are given in Section 6.

2. The n-Parameter n-Point Newton-Type Method with Optimal Order 2^n

Combining the first step of method (3) with Newton's method [21,22], we construct the following one-parameter Newton-type method:

$$\begin{cases} y_k = t_k - \frac{f(t_k)}{L_1 f(t_k) + f'(t_k)}, \\ t_{k+1} = y_k - \frac{f(y_k)}{f'(y_k)}, \end{cases} \tag{5}$$

where $L_1 \in \mathbb{R}$ is a real parameter. To reduce the computational cost of method (5), we approximate $f(t)$ by Hermite's interpolation polynomial $H_3(t)$. Interpolation polynomial $H_3(t)$ is given by:

$$H_3(t) = f(y_k) + f[y_k, t_k](t - y_k) + f[y_k, t_k, t_k](t - y_k)(t - t_k) + L_2(t - y_k)(t - t_k)^2, \tag{6}$$

such that $H_3(t_k) = f(t_k), H_3(y_k) = f(y_k)$, $H_3'(t_k) = f'(t_k)$ and $L_2 \in \mathbb{R}$. The derivative of $H_3(t)$ at y_k is:

$$H_3'(y_k) = f[y_k, t_k] + f[y_k, t_k, t_k](y_k - t_k) + L_2(y_k - t_k)^2. \tag{7}$$

Replacing $f'(y_k)$ with $H_3'(y_k)$ in (5), we obtain a two-parameter Newton-type method:

$$\begin{cases} y_k = t_k - \frac{f(t_k)}{L_1 f(t_k) + f'(t_k)}, \\ t_{k+1} = y_k - \frac{f(y_k)}{f[y_k, t_k] + f[y_k, t_k, t_k](y_k - t_k) + L_2(y_k - t_k)^2}, \end{cases} \tag{8}$$

where $f[y_k, t_k, t_k] = \frac{f[y_k, t_k] - f'(t_k)}{y_k - t_k}$ and $L_2 \in \mathbb{R}$ are real parameters. Combining method (8) with Newton's method, we obtain the following three-step method:

$$\begin{cases} y_k = t_k - \frac{f(t_k)}{L_1 f(t_k) + f'(t_k)}, \\ z_k = y_k - \frac{f(y_k)}{f[y_k, t_k] + f[y_k, t_k, t_k](y_k - t_k) + L_2(y_k - t_k)^2}, \\ t_{k+1} = z_k - \frac{f(z_k)}{f'(z_k)}. \end{cases} \tag{9}$$

We approximate $f(t)$ in (9) by the following interpolation polynomial $H_4(t)$ of a degree of four:

$$\begin{aligned} H_4(t) &= f(z_k) + f[z_k, y_k](t - z_k) + f[z_k, y_k, t_k](t - z_k)(t - y_k) \\ &+ f[z_k, y_k, t_k, t_k](t - z_k)(t - y_k)(t - t_k) + L_3(t - z_k)(t - y_k)(t - t_k)^2. \end{aligned} \tag{10}$$

$H_4(t)$ satisfies interpolation conditions $H_4(t_k) = f(t_k)$, $H_4(y_k) = f(y_k)$, $H_4(z_k) = f(z_k)$ and $H_4'(t_k) = f'(t_k)$. The derivative of $H_4(t)$ at z_k is:

$$\begin{aligned} H_4'(z_k) = f[z_k, y_k] + f[z_k, y_k, t_k](z_k - y_k) + f[z_k, y_k, t_k, t_k] \times \\ (z_k - y_k)(z_k - t_k) + L_3(z_k - y_k)(z_k - t_k)^2, \end{aligned} \tag{11}$$

where $L_3 \in \mathbb{R}$. Replacing $f'(z_k)$ with $H_4'(z_k)$ in (9), we obtain the following three-parameter method:

$$\begin{cases} y_k = t_k - \frac{f(t_k)}{L_1 f(t_k) + f'(t_k)}, \\ z_k = y_k - \frac{f(y_k)}{f[y_k, t_k] + f[y_k, t_k, t_k](y_k - t_k) + L_2(y_k - t_k)^2}, \\ t_{k+1} = z_k - \frac{f(z_k)}{H_4'(z_k)}, \end{cases} \tag{12}$$

where $H_4'(z_k) = f[z_k, y_k] + f[z_k, y_k, t_k](z_k - y_k) + f[z_k, y_k, t_k, t_k](z_k - y_k)(z_k - t_k) + L_3(z_k - y_k)(z_k - t_k)^2$ and $L_i \in \mathbb{R}, (i = 1, 2, 3)$.

Furthermore, we construct the following n-parameter n-point Newton-type method without memory:

$$\begin{cases} t_{k,1} = t_{k,0} - \frac{f(t_{k,0})}{L_1 f(t_{k,0}) + f'(t_{k,0})}, \\ t_{k,2} = t_{k,1} - \frac{f(t_{k,1})}{f[t_{k,1}, t_{k,0}] + f[t_{k,1}, t_{k,0}, t_{k,0}](t_{k,1} - t_{k,0}) + L_2(t_{k,1} - t_{k,0})^2}, \\ t_{k,3} = t_{k,2} - \frac{f(t_{k,2})}{H_4'(t_{k,2})}, \\ \cdots, \\ t_{k,n} = t_{k,n-1} - \frac{f(t_{k,n-1})}{H_{n+1}'(t_{k,n-1})}, \end{cases} \tag{13}$$

where

$$H_{n+1}'(t_{k,n-1}) = f[t_{k,n-1}, t_{k,n-2}] + f[t_{k,n-1}, t_{k,n-2}, \cdots, t_{k,1}, t_{k,0}, t_{k,0}] \prod_{j=2}^{n} (t_{k,n-1} - t_{k,n-j})$$

$$+ \sum_{j=3}^{n} \left\{ f[t_{k,n-1}, t_{k,n-2}, \cdots, t_{k,n-j}] \prod_{i=2}^{j-1} (t_{k,n-1} - t_{k,n-i}) \right\} + L_{n-1}(t_{k,n-1} - t_{k,0}) \prod_{t=2}^{n} (t_{k,n-1} - t_{k,n-t}),$$

$t_{k,0} = t_k$ and $L_i \in \mathbb{R}, (i = 1, 2 \cdots, n)$. The Petković's method (2) and Wang's method (3) are two special cases of method (13) for $L_i = 0, (i = 1, 2, \cdots n)$ and $L_i = 0, (i = 2, 3, \cdots n)$, respectively.

Theorem 1. *Let $a \in \mathbb{R}$ be a zero of a sufficiently differentiable function $f : D \subseteq \mathbb{R} \to \mathbb{R}$ in an open set D. Assume that initial approximation t_0 is sufficiently close to a. Then, method (8) reaches an optimal convergence order four and satisfies the following:*

$$e_{k+1} = (c_2 + L_1)[c_2(c_2 + L_1) - c_3 + \frac{L_2}{f'(a)}]e_k^4 + O(e_k^5). \tag{14}$$

where $c_n = (1/n!) f^{(n)}(a)/f'(a), n \geq 2$.

Proof. Let $e_k = t_k - a$, $ey_k = y_k - a$, $ez_k = z_k - a$ and $c_n = (1/n!) f^{(n)}(a)/f'(a), n \geq 2$. Using the Taylor expansion of f at a, we obtain:

$$f(t_k) = f'(a)(e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + c_5 e_k^5 + c_6 e_k^6 + c_7 e_k^7 + c_8 e_k^8 + O(e_k^9)), \tag{15}$$

and

$$f'(t_k) = f'(a)(1 + 2c_2 e_k + 3c_3 e_k^2 + 4c_4 e_k^3 + 5c_5 e_k^4 + 6c_6 e_k^5 + 7c_7 e_k^6 + 8c_8 e_k^7 + O(e_k^8)). \tag{16}$$

According to (12), (15) and (16), we have:

$$ey_k = (c_2 + L_1)e_k^2 + (-2c_2^2 + 2c_3 - 2c_2 L_1 - L_1^2)e_k^3 + (4c_2^3 - 7c_2 c_3 + 3c_4 + 5c_2^2 L_1 - 4c_3 L_1 + 3c_2 L_1^2 + L_1^3)e_k^4 + O(e_k^5). \tag{17}$$

Using the Taylor expansion of $f(y_k)$ at a, we obtain:

$$f(y_k) = f'(a)(ey_k + c_2 ey_k^2 + c_3 ey_k^3 + c_4 ey_k^4 + c_5 ey_k^5 + c_6 ey_k^6 + c_7 ey_k^7 + c_8 ey_k^8 + +O(ey_k^9)), \tag{18}$$

and

$$f[y_k, t_k] = f'(a)[1 + c_2 e_k + (c_2^2 + c3 + c2L_1)e_k^2 + (-2c_2^3 + 3c_2 c_3 + c_4 - 2c_2^2 L_1 + c_3 L_1 - c_2 L_1^2)e_k^3 + (4c_2^4 + 2c_3^2 + c_5 + 5c_2^3 L_1 + c_4 L_1 + c_2^2(-8c_3 + 3L_1^2) + c_2(4c_4 - 4c_3 L_1 + L_1^3))e_k^4] + O(e_k^5). \tag{19}$$

From (16) and (19), we obtain:

$$f[y_k, t_k, t_k] = \frac{f[y_k, t_k] - f'(t_k)}{y_k - t_k}$$

$$= f'(a)(c_2 + 2c_3e_k + (c_2c_3 + 3c_4 + c_3L_1)e_k^2 + (-2c_2^2c_3 + 2c_3^2 + 4c_5 + 2c_4L_1 - c3L_1^2 + 2c_2(c_4 - c_3L_1))e_k^3) + O(e_k^4). \quad (20)$$

Using (8) and (18)–(20), we attain:

$$ez_k = (c_2 + L_1)[c_2(c_2 + L_1) + \frac{L_2}{f'(a)} - c_3]e_k^4 + O(e_k^5). \quad (21)$$

This means that the convergence order of method (8) is four. This concludes the proof. □

For method (12), we can obtain the following convergence theorem.

Theorem 2. *Let $a \in \mathbb{R}$ be a zero of a sufficiently differentiable function $f : D \subseteq \mathbb{R} \to \mathbb{R}$ in an open set D. Assume that initial approximation t_0 is sufficiently close to a. Then, method (12) arrives the optimal order eight and satisfies the following error equation:*

$$e_{k+1} = (c_2 + L_1)^2[c_2(c_2 + L_1) - c_3 + \frac{L_2}{f'(a)}]\{c_2[c_2(c_2 + L_1) - c_3 + \frac{L_2}{f'(a)}] + c_4 - \frac{L_3}{f'(a)}\}e_k^8 + O(e_k^9), \quad (22)$$

where $c_n = (1/n!)f^{(n)}(a)/f'(a), n \geq 2$.

Proof. Let $e_k = t_k - a$, $ey_k = y_k - a$, $ez_k = z_k - a$ and $c_n = (1/n!)f^{(n)}(a)/f'(a), n \geq 2$. Using the Taylor expansion of $f(z_k)$ at a, we obtain:

$$f(z_k) = f'(a)(ez_k + c_2ez_k^2 + c_3ez_k^3 + c_4ez_k^4 + c_5ez_k^5 + O(ez_k^6)), \quad (23)$$

$$f[z_k, y_k] = f'(a) + c_2f'(a)(c_2 + L_1)e_k^2 - c_2f'(a)(2c_2^2 - 2c_3 + 2c_2L_1 + L_1^2)e_k^3 + (5c_2^4f'(a) + 7c_2^3f'(a)L_1 + c_3f'(a)L_1^2 + c_2^2(-7c_3f'(a) + 4f'(a)L_1^2 + L_2) + c_2(3c_4f'(a) + L_1(-3c_3f'(a) + f'(a)L_1^2 + L_2)))e_k^4 + O(e_k^5). \quad (24)$$

From (19) and (24), we obtain:

$$f[z_k, y_k, t_k] = \frac{f[z_k, y_k] - f[y_k, t_k]}{z_k - t_k}$$

$$= f'(a)(c_2 + c_3e_k + (c_2c_3 + c_4 + c_3L_1)e_k^2 + (-2c_2^2c_3 + 2c_3^2 + c_5 + c_4L_1 - c_3L_1^2 + c_2(c_4 - 2c_3L_1))e_k^3) + O(e_k^4). \quad (25)$$

Using (20) and (25), we have

$$f[z_k, y_k, t_k, t_k] = \frac{f[z_k, y_k, t_k] - f[y_k, t_k, t_k]}{z_k - t_k}$$

$$= f'(a)(c_3 + 2c_4e_k + (c_2c_4 + 3c_5 + c_4L_1)e_k^2 + (-2c_2^2c_4 + 2c_3c_4 + 4c_6 + 2c_5L_1 - c_4L_1^2 + 2c_2(c_5 - c_4L_1))e_k^3) + O(e_k^4). \quad (26)$$

Therefore, from (12), (25) and (26), we obtain:

$$e_{k+1} = (c_2 + L_1)^2[c_2(c_2 + L_1) - c_3 + \frac{L_2}{f'(a)}]$$

$$\times \{c_2[c_2(c_2+L_1)-c_3+\frac{L_2}{f'(a)}]+c_4-\frac{L_3}{f'(a)}\}e_k^8+O(e_k^9). \quad (27)$$

This means that method (12) arrives at the optimal order eight. This concludes the proof. □

According to the above study, we can obtain the following convergence theorem.

Theorem 3. *Let $a \in \mathbb{R}$ be a zero of a sufficiently differentiable function $f : D \subseteq \mathbb{R} \to \mathbb{R}$ in an open set D. Assume that initial approximation t_0 is sufficiently close to a. Then, the n-parameter n-point Newton-type iterative scheme (13) without memory reaches the optimal order 2^n and its error relation is*

$$e_{k+1} = e_{k,n} = t_{k,n} - \alpha = q_n e_{k,0} \prod_{i=0}^{n-1} e_{k,i} = q_n q_{n-1} q_{n-2}^2 \cdots q_1^{2^{n-2}} q_0^{2^{n-1}} e_k^{2^n} + O(e_k^{2^n+1}), \quad (28)$$

where $e_{k,j} = t_{k,j} - a$, $(j = 0, 1, 2, \cdots, n), e_k = e_{k,0} = t_{k,0} - a, q_0 = 1, q_1 = c_2 + L_1$ and $q_n = c_n q_{n-1} + (-1)^{n-1} c_{n+1} + (-1)^n \frac{L_n}{f'(a)}, n = 2, 3, \cdots$.

Proof. Induction method is used to prove this Theorem. Form Theorem 1, we know that the Theorem is valid for $n = 2$ and $n = 3$. Suppose that Equation (28) is true for $n = N - 1$, then we have the error relation

$$e_{k,N-1} = t_{k,N-1} - a = q_{N-1} e_{k,0} \prod_{i=0}^{N-2} e_{k,i}$$

$$= q_{N-1} q_{N-2} q_{N-3}^2 \cdots q_1^{2^{N-3}} q_0^{2^{N-2}} e_k^{2^{N-1}} + O(e_k^{2^{N-1}+1}). \quad (29)$$

Noting that $e_{k,0} e_{k,0} e_{k,1} e_{k,2} \cdots e_{k,N-1} = O(e_k^{1+1+2+2^2+\cdots+2^{N-1}}) = O(e_k^{2^N})$ and taking $n = N$, we obtain:

$$\begin{aligned}
e_{k+1} = e_{k,N} = t_{k,N} - a &= e_{k,N-1} - \frac{f[t_{k,N-1}, a] e_{k,N-1}}{H'_N(t_{k,N-1})} = e_{k,N-1}\left(\frac{H'_N(t_{k,N-1}) - f[t_{k,N-1}, a]}{H'_N(t_{k,N-1})}\right) \\
&= e_{k,N-1}(f[t_{k,N-1}, t_{k,N-2}] + f[t_{k,N-1}, t_{k,N-2}, t_{k,N-3}](t_{k,N-1} - t_{k,N-2}) + \cdots \\
&+ f[t_{k,N-1}, t_{k,N-2}, \cdots, t_{k,1}, t_{k,0}, t_{k,0}](t_{k,N-1} - t_{k,N-2})(t_{k,N-1} - t_{k,N-3}) \cdots (t_{k,N-1} - t_{k,0}) \\
&+ (-1)^N L_{N-1} e_{k,N-2} e_{k,N-3} \cdots e_{k,0}^2 - f[t_{k,N-1}, a])[f'(a) + O(e_k)]^{-1} \\
&= e_{k,N-1}\{f[t_{k,N-2}, t_{k,N-1}, a] e_{k,N-2} + f[t_{k,N-1}, t_{k,N-2}, t_{k,N-3}] e_{k,N-1} \\
&- f[t_{k,N-1}, t_{k,N-2}, t_{k,N-3}] e_{k,N-2} + \cdots + (-1)^{N-1} f[t_{k,N-1}, t_{k,N-2}, \cdots, t_{k,1}, t_{k,0}, t_{k,0}] \\
&\times e_{k,N-2} e_{k,N-3} \cdots e_{k,0} + (-1)^N L_{N-1} e_{k,N-2} e_{k,N-3} \cdots e_{k,0}^2\} \left[f'(a) + O(e_k)\right]^{-1} \\
&= e_{k,N-1}\{f[t_{k,N-1}, t_{k,N-2}, t_{k,N-3}] e_{k,N-1} + (-1)^{N-1} f[t_{k,N-1}, t_{k,N-2}, \cdots, t_{k,1}, t_{k,0}, t_{k,0}, a] \\
&\times e_{k,N-2} e_{k,N-3} \cdots e_{k,0}^2 + (-1)^N L_{N-1} e_{k,N-2} e_{k,N-3} \cdots e_{k,0}^2 + O(e_k^{2^{N-1}+1})\} \left[f'(a) + O(e_k)\right]^{-1} \\
&= e_{k,N-1}[c_2 q_{N-1} e_{k,0} \prod_{i=0}^{N-2} e_{k,i} + (-1)^{N-1} c_{N+1} e_{k,0} \prod_{i=0}^{N-2} e_{k,i} + (-1)^N \frac{L_{N-1}}{f'(a)} e_{k,0} \prod_{i=0}^{N-2} e_{k,i} + O(e_k^{2^{N-1}+1})] \\
&= e_{k,0} \prod_{i=0}^{N-1} e_{k,i}[c_2 q_{N-1} + (-1)^{N-1} c_{N+1} + (-1)^N \frac{L_{N-1}}{f'(a)}] + O(e_k^{2^N+1}) \\
&= q_N e_{k,0} \prod_{i=0}^{N-1} e_{k,i} + O(e_k^{2^N+1}). \quad (30)
\end{aligned}$$

Hence, we obtain:

$$e_{k+1} = e_{k,n} = t_{k,n} - a = q_n e_{k,0} \prod_{i=0}^{n-1} e_{k,i} = q_n q_{n-1} q_{n-2}^2 \cdots q_1^{2^{n-2}} q_0^{2^{n-1}} e_k^{2^n} + O(e_k^{2^n+1}). \quad (31)$$

The proof is completed. □

3. A General *n*-Point Newton-Type Multipoint Iterative Method with Memory

Theorem 3 shows that the n-parameter n-point method (13) without memory reaches the optimal order 2^n. Taking $L_1 = -c_2$ and $L_i = \frac{f^{(i+1)}(a)}{(i+1)!}, (i = 2, 3, \cdots, n)$ in (13), the convergence order of method (13) can be improved. In this section, we replace the constant parameters $L_i, (i = 1, 2, \cdots, n)$ of method (13) with the variable parameters $L_{k,i}$ and obtain a new n-parameter n-point method with memory. Variable parameters $L_{k,i}$ are constructed by the iterative sequences from current and previous iterations and satisfy the conditions $lim_{k\to\infty} L_{k,1} = -c_2 = -\frac{f''(a)}{2f'(a)}$ and $lim_{k\to\infty} L_{k,i} = \frac{f^{(i+1)}(a)}{(i+1)!}$. To get the maximal order of convergence of the n-parameter n-point Newton-type method, we design the following variable parameters $L_{k,i}, (i = 1, \cdots, n)$ by using Hermite's interpolation polynomial

$$L_{k,1} = -\frac{H''_{n+2}(t_{k,0})}{2f'(t_{k,0})}, \quad (32)$$

and

$$L_{k,i+1} = \frac{H^{(i+2)}_{n+i+2}(t_{k,i})}{(i+2)!}, (0 \le i \le n), \quad (33)$$

where

$$H_{n+2}(t) = H_{n+2}(t : t_{k,0}, t_{k,0}, t_{k-1,n-1}, \cdots, t_{k-1,0}, t_{k-1,0}),$$

and

$$H_{n+i+2}(t) = H_{n+i+2}(t : t_{k,i}, t_{k,i-1}, \cdots, t_{k,1}, t_{k,0}, t_{k,0}, t_{k-1,n-1}, \cdots, t_{k-1,0}, t_{k-1,0}).$$

Replacing the constant parameter L_i with variable parameter $L_{k,i}$ in (13), we obtain: a general n-parameter n-point method with memory as follows:

$$\begin{cases} t_{k,1} = t_{k,0} - \frac{f(t_{k,0})}{L_{k,1} f(t_{k,0}) + f'(t_{k,0})}, \\ t_{k,2} = t_{k,1} - \frac{f(t_{k,1})}{f[t_{k,1}, t_{k,0}] + f[t_{k,1}, t_{k,0}, t_{k,0}](t_{k,1} - t_{k,0}) + L_{k,2}(t_{k,1} - t_{k,0})^2}, \\ \cdots, \\ t_{k,n} = t_{k,n-1} - \frac{f(t_{k,n-1})}{H'_n(t_{k,n-1})}, \end{cases} \quad (34)$$

where $L_{k,i}, (i = 1, \cdots, n)$ are the variable parameters constructed by (32)–(33) and $H'_n(t_{k,n-1}) = f[t_{k,n-1}, t_{k,n-2}] + \sum_{j=3}^{n} \{f\left[t_{k,n-1}, t_{k,n-2}, \cdots, t_{k,n-j}\right] \prod_{i=2}^{j-1} (t_{k,n-1} - t_{k,n-i})\} + (t_{k,n-1} - t_{k,0}) L_{k,n-1} \prod_{t=2}^{n} (t_{k,n-1} - t_{k,n-t}) + f[t_{k,n-1}, t_{k,n-2}, \cdots, t_{k,1}, t_{k,0}, t_{k,0}] \prod_{j=2}^{n} (t_{k,n-1} - t_{k,n-j}), (n \ge 2)$.

From (14), (22) and (28), the error relations of method (34) can be obtained

$$e_{k,1} = (c_2 + L_{k,1}) e_k^2 + O(e_k^3), \quad (35)$$

$$e_{k,2} = (c_2 + L_{k,1})[c_2(c_2 + L_{k,1}) - c_3 + \frac{L_{k,2}}{f'(a)}] e_k^4 + O(e_k^5), \quad (36)$$

and

$$e_{k,j} = t_{k,j} - a = q_{k,j}e_{k,0}\prod_{i=0}^{j-1} e_{k,i} = q_{k,j}q_{k,j-1}q_{k,j-2}^2 \cdots q_{k,1}^{2^{j-2}} q_{k,0}^{2^{j-1}} e_k^{2^j} + O(e_k^{2^j+1}), 2 \le j \le n, \quad (37)$$

where $e_{k,j} = t_{k,j} - a, (j = 0, 1, 2, \cdots, n)$, $e_k = e_{k,0} = t_{k,0} - a$, $q_{k,0} = 1$, $q_{k,1} = c_2 + L_{k,1}$, $q_{k,n} = (-1)^{n-1}c_{n+1} + (-1)^n \frac{L_{k,n}}{f'(a)} + c_2 q_{k,n-1}, (n \ge 2)$.

The above consideration leads to the following Lemma.

Lemma 1. *Let $H_{n+i+2}(t), (i = 0, 1, 2, \cdots, n)$ be the Hermite interpolation polynomial of degree $n + i + 2$ satisfying $H_{n+i+2}(t_{k,l}) = f(t_{k,l}), (l = 0, 1, \cdots, i)$, $H_{n+i+2}(t_{k-1,n-j}) = f(t_{k-1,n-j})$, $(j = 1, \cdots, n)$, $H'_{n+i+2}(t_{k,0}) = f'(t_{k,0})$ and $H'_{n+i+2}(t_{k-1,0}) = f'(t_{k-1,0})$. Let $f(t)$ and its derivative $f^{(n+i+2)}$ be continuous in interval D. Nodes $t_{k,i}, \cdots, t_{k,1}, t_{k,0}, t_{k,0}, t_{k-1,n-1}, \cdots, t_{k-1,1}, t_{k-1,0}$, $t_{k-1,0}$ are contained in interval D, which are sufficiently close to the simple zero a of $f(t)$. Define the error $e_{k,l} = t_{k,l} - a$ and assume that the condition $e_{k,l} = O(e_{k-1,n} \cdots e_{k-1,0}e_{k-1,0}), (0 \le l \le n)$ holds. Then*

$$H_{n+i+2}^{(i+2)}(t_{k,i}) \sim (i+2)!f'(a)\left[c_{i+2} - (-1)^{n+1}c_{n+i+3}e_{k-1,0}\prod_{j=1}^{n} e_{k-1,n-j}\right], (0 \le i \le n). \quad (38)$$

Proof. The error relation of the Hermite's interpolation polynomial $H_{n+i+2}(t)$ is given by

$$f(t) - H_{n+i+2}(t) = \frac{f^{(n+i+3)}(\eta)}{(n+i+3)!}[(t - t_{k,0})\prod_{l=0}^{i}(t - t_{k,l})][(t - t_{k-1,0})\prod_{j=1}^{n}(t - t_{k-1,n-j})], \quad \eta \in D. \quad (39)$$

Differentiating (39) at t, we obtain:

$$\begin{aligned} f^{(i+2)}(t) - H_{n+i+2}^{(i+2)}(t) &= \frac{1}{(n+i+3)!}\{[(t - t_{k,0})\prod_{l=0}^{i}(t - t_{k,l})][(t - t_{k-1,0})\prod_{j=1}^{n}(t - t_{k-1,n-j})] \\ &\times \frac{d^{i+2}}{dt^{i+2}}[f^{(n+i+3)}(\eta)] + \sum_{m=1}^{i+1} C_{i+2}^m[(t - t_{k,0})\prod_{l=0}^{i}(t - t_{k,l})(t - t_{k-1,0})\prod_{j=1}^{n}(t - t_{k-1,n-j})]^{(m)} \\ &\times \frac{d^{i+2-m}}{dt^{i+2-m}}[f^{(n+i+3)}(\eta)] + [(t - t_{k,0})\prod_{l=0}^{i}(t - t_{k,l})(t - t_{k-1,0})\prod_{j=1}^{n}(t - t_{k-1,n-j})]^{(i+2)}f^{(n+i+3)}(\eta)\}, \end{aligned} \quad (40)$$

and

$$\begin{aligned} f^{(i+2)}(t) - H_{n+i+2}^{(i+2)}(t) &= \frac{1}{(n+i+3)!}\{(t - t_{k,0})\prod_{l=0}^{i}(t - t_{k,l})(t - t_{k-1,0})\prod_{j=1}^{n}(t - t_{k-1,n-j}) \\ &\times \frac{d^{i+2}}{dt^{i+2}}[f^{(n+i+3)}(\eta)] + \sum_{m=1}^{i+1} C_{i+2}^m[(t - t_{k,0})\prod_{l=0}^{i}(t - t_{k,l})(t - t_{k-1,0})\prod_{j=1}^{n}(t - t_{k-1,n-j})]^{(m)} \\ &\times \frac{d^{i+2-m}}{dt^{i+2-m}}[f^{(n+i+3)}(\eta)] + (i+2)![(t - t_{k-1,0})\prod_{j=1}^{n}(t - t_{k-1.n-j}) + P_{n+1}(t)]f^{(n+i+3)}(\eta)\}, \eta \in D, \end{aligned} \quad (41)$$

where $P_{n+1}(t)$ is a polynomial of degree $n + 1$. Taylor's series of $f(t)$ and its derivatives $f^{(i)}(t)$ at the points $t_{k,i}$ about zero a is

$$f'(t_{k,i}) = [1 + 2c_2e_{k,i} + 3c_3e_{k,i}^2 + O(e_{k,i}^3)]f'(a), \quad (42)$$

$$f^{(i+2)}(t_{k,i}) = [(i+2)!c_{i+2} + (i+3)!c_{i+3}e_{k,i} + O(e_{k,i}^2)]f'(a), \quad (43)$$

and

$$f^{(n+i+3)}(\eta) = [(n+i+3)!c_{n+i+3} + (n+i+4)!c_{n+i+4}e_\eta + O(e_\eta^2)]f'(a), \tag{44}$$

where $t_{k,i} \in D, \eta \in D$ and $e_\eta = \eta - a$.

Taking $t = t_{k,i}$ in (41) and using (44), we obtain:

$$H_{n+i+2}^{(i+2)}(t_{k,i}) \sim f^{(i+2)}(t_{k,i}) - (i+2)!\frac{f^{(n+i+3)}(\eta)}{(n+i+3)!}[P_{n+1}(t_{k,i}) + (t_{k,i} - t_{k-1,0})\prod_{j=1}^{n}(t_{k,i} - t_{k-1,n-j})]$$

$$\sim (i+2)!f'(a)[c_{i+2} - (-1)^{n+1}c_{n+i+3}e_{k-1,0}\prod_{j=1}^{n}e_{k-1,n-j}]. \tag{45}$$

The proof is completed. □

From (33) and (45), we obtain:

$$L_{k,i+1} = \frac{H_{n+i+2}^{(i+2)}(t_{k,i})}{(i+2)!} \sim f'(a)[c_{i+2} - (-1)^{n+1}c_{n+i+3}e_{k-1,0}\prod_{j=1}^{n}e_{k-1,n-j}], \tag{46}$$

and

$$c_{i+2} - \frac{L_{k,i+1}}{f'(a)} \sim (-1)^{n+1}c_{n+i+3}e_{k-1,0}\prod_{j=1}^{n}e_{k-1,n-j}. \tag{47}$$

If sequence $\{t_n\}$ converges to the zero a with order r, we can get

$$e_{k+1} = e_{k+1,0} = e_{k,n} \sim Q_{k,n}e_{k,0}^r, \tag{48}$$

where $e_{k,n} = t_{k,n} - a$, and $Q_{k,n}$ is the asymptotic error constant. Suppose sequence $\{t_{k,j}\}$ has the convergence order r_j, we obtain:

$$e_{k,j} \sim Q_{k,j}e_{k,0}^{r_j}, 1 \le j \le n-1, \tag{49}$$

and

$$c_{i+2} - \frac{L_{k,i+1}}{f'(a)} \sim (-1)^{n+1}c_{n+i+3}e_{k-1,0}\prod_{j=1}^{n}e_{k-1,n-j}$$

$$\sim (-1)^{n+1}c_{n+i+3}e_{k-1,0}^{r_0+r_0+r_1+\cdots+r_{n-1}}\prod_{j=1}^{n}Q_{k-1,n-j}$$

$$\sim T_{k,i+1}e_{k-1,0}^R, \tag{50}$$

where $r_0 = 1, R = r_0 + \sum_{i=0}^{n-1} r_i$ and $T_{k,i+1} = (-1)^{n+1}c_{n+i+3}\prod_{j=1}^{n}Q_{k-1,n-j}$.

From (32) and (35), we obtain:

$$q_{k,1} = c_2 + L_{k,1} \sim T_{k,1}e_{k-1,0}^R, \tag{51}$$

where $T_{k,1} = (-1)^{n+1}c_{n+3}\prod_{j=1}^{n}Q_{k-1,n-j}$.

$$e_{k,1} \sim q_{k,1}e_{k,0}^2 \sim T_{k,1}e_{k,0}^2e_{k-1,0}^R. \tag{52}$$

Lemma 2. *Let $a \in I$ be a simple zero of a sufficiently differentiable function f, then the $q_{k,j}$ in (37) such that*

$$q_{k,j} \sim M_{k,j}e_{k-1,0}^R, (2 \le j \le n), \tag{53}$$

where $r_0 = 1, R = r_0 + \sum_{i=0}^{n-1} r_i, T_{k,i} = (-1)^{n+1} c_{n+i+2} \prod_{j=1}^{n} Q_{k-1,n-j}$ *and* $M_{k,j} = \sum_{i=1}^{j} (-1)^{i-1} c_2^{j-i} T_{k,i}$.

Proof. We proof the lemma by induction. Using (50)–(51) and taking $j = 2$ in (37), we obtain:

$$\begin{aligned} e_{k,2} &\sim q_{k,2} e_{k,1} e_{k,0}^2 \\ &\sim \{c_2 q_{k,1} - c_3 + \frac{L_{k,2}}{f'(a)}\} e_{k,1} e_{k,0}^2 \\ &\sim (c_2 T_{k,1} - T_{k,2}) e_{k,1} e_{k,0}^2 e_{k-1,0}^R. \end{aligned} \tag{54}$$

Let $M_{k,2} = c_2 T_{k,1} - T_{k,2}$ in (54), we obtain:

$$e_{k,2} \sim q_{k,2} e_{k,1} e_{k,0}^2 \sim M_{k,2} e_{k,1} e_{k,0}^2 e_{k-1,0}^R, \tag{55}$$

and

$$q_{k,2} \sim M_{k,2} e_{k-1,0}^R. \tag{56}$$

Using (50) and (51) and taking $j = 3$ in (37), we get

$$\begin{aligned} e_{k,3} &\sim q_{k,3} e_{k,2} e_{k,1} e_{k,0}^2 \\ &\sim \{c_2 q_{k,2} + c_4 - \frac{L_{k,2}}{f'(a)}\} e_{k,2} e_{k,1} e_{k,0}^2 \\ &\sim (c_2(c_2 T_{k,1} - T_{k,2}) + T_{k,3}) e_{k,2} e_{k,1} e_{k,0}^2 e_{k-1,0}^R. \end{aligned} \tag{57}$$

Let $M_{k,3} = c_2(c_2 T_{k,1} - T_{k,2}) + T_{k,3}$, we obtain:

$$\begin{aligned} e_{k,3} &\sim q_{k,3} e_{k,2} e_{k,1} e_{k,0}^2 \\ &\sim M_{K,3} e_{k,2} e_{k,1} e_{k,0}^2 e_{k-1,0}^R, \end{aligned} \tag{58}$$

and

$$q_{k,3} \sim M_{k,3} e_{k-1,0}^R. \tag{59}$$

Assume that (53) holds for $j = n - 1$, we obtain

$$q_{k,n-1} \sim M_{k,n-1} e_{k-1,0}^R. \tag{60}$$

Using (50) and (51) and taking $j = n$ in (37), we get

$$\begin{aligned} e_{k,n} &\sim q_{k,n} e_{k,0} \prod_{j=0}^{n-1} e_{k,i} \\ &\sim \{c_2 q_{k,n-1} + (-1)^{n-1} c_{n+1} + (-1)^n \frac{L_{k,n-1}}{f'(a)}\} e_{k,0} \prod_{j=0}^{n-1} e_{k,i} \\ &\sim (T_{k,n} + c_2 M_{k,n-1}) e_{k,0} \prod_{j=0}^{n-1} e_{k,i} e_{k-1,0}^R \\ &\sim M_{k,n} e_{k,0} \prod_{j=0}^{n-1} e_{k,i} e_{k-1,0}^R, \end{aligned} \tag{61}$$

and

$$q_{k,n} \sim M_{k,n} e_{k-1,0}^R. \tag{62}$$

The proof is completed. □

According to the Lemma 1 and Lemma 2, we obtain: the convergence theorem as follows:

Theorem 4. *If t_0 is sufficiently close to a simple zero a of the sufficiently differentiable function $f(t)$ and the variable parameters $L_{k,1}$ and $L_{k,i}, (2 \le i \le n)$ of method (34) are calculated by (32) and (33), respectively. Then, the n-parameter n-point Newton-type iterative method (34) with memory reaches order $(2^n + 2^{n-1} - 1 + \sqrt{2^{2n+1} + 2^{2n-2} + 2^n + 1})/2$.*

Proof. From (48) and (49), we obtain:

$$e_{k+1} = e_{k,n} \sim Q_{k,n}(Q_{k-1,n}e_{k-1,0}^r)^r \sim Q_{k,n}Q_{k-1,n}^r e_{k-1,0}^{r^2}, \quad e_{k,n} = x_{k,n} - a, \tag{63}$$

$$e_{k,j} \sim Q_{k,j}(Q_{k-1,n}e_{k-1,0}^r)^{r_j} \sim Q_{k,j}Q_{k-1,n}^{r_j} e_{k-1,0}^{rr_j}, \quad 1 \le j \le n-1. \tag{64}$$

Using (37) and (49), we arrive at:

$$\begin{aligned} e_{k,j} = t_{k,j} - a &\sim q_{k,j}e_{k,0}\prod_{i=0}^{j-1} e_{k,i} \\ &\sim q_{k,j}e_{k,0}^2\prod_{i=1}^{j-1}(Q_{k,i}e_{k,0}^{r_i}) \\ &\sim q_{k,j}S_{k,j}e_{k,0}^{2+r_1+r_2+\cdots r_{j-1}} \\ &\sim M_{k,j}e_{k-1,0}^R S_{k,j}e_{k,0}^{2+r_1+r_2+\cdots r_{j-1}} \\ &\sim M_{k,j}e_{k-1,0}^R S_{k,j}Q_{k-1,n}^{2+r_1+r_2+\cdots r_{j-1}} e_{k-1,0}^{r(2+r_1+r_2+\cdots r_{j-1})} \\ &\sim M_{k,j}S_{k,j}Q_{k-1,n}^{2+r_1+r_2+\cdots r_{j-1}} e_{k-1,0}^{R+r(2+r_1+r_2+\cdots r_{j-1})}, \end{aligned} \tag{65}$$

where $S_{k,j} = \prod_{i=1}^{j-1} Q_{k,i}, 2 \le j \le n$.

According to (48) and (53), we have:

$$\begin{aligned} e_{k,1} \sim q_{k,1}e_{k,0}^2 &\sim T_{k,1}e_{k-1,0}^R(Q_{k-1,n}e_{k-1,0}^r)^2 \\ &\sim T_{k,1}Q_{k-1,n}^2 e_{k-1,0}^{2r+R}. \end{aligned} \tag{66}$$

Comparing the exponents of error $e_{k-1,0}$ in pairs of ((64), (66)) for $j = 1$, ((63), (65)) for $j = n$ and ((64), (65)) for $2 \le j \le .n - 1$, we obtain:

$$\begin{cases} r^2 = r(2 + r_1 + r_2 + \cdots + r_{n-1}) + R = rR + R, \\ rr_{n-1} = r(2 + r_1 + r_2 + \cdots + r_{n-2}) + R, \\ rr_j = r(2 + r_1 + r_2 + \cdots + r_{j-1}) + R, (j = 2, \cdots, n-2), \\ rr_1 = R + 2r. \end{cases} \tag{67}$$

According to (67), we obtain:

$$\frac{r^2}{1+r} = R = 2 + r_1 + r_2 + \cdots r_{n-1}, \tag{68}$$

$$r_j = 2^{j-1}r_1, (j = 2, \cdots, n-1), \tag{69}$$

$$r = 2r_{n-1}, \tag{70}$$

$$r_1 = 2 + \frac{r}{1+r}. \tag{71}$$

From (68) and (71), we obtain:

$$r^2 - (2^n + 2^{n-1} - 1)r - 2^n = 0. \tag{72}$$

The solution of the equation (72) is $(2^n + 2^{n-1} - 1 + \sqrt{2^{2n+1} + 2^{2n-2} + 2^n + 1})/2$. Thus, the n-parameter n-point Newton-type method with memory has the convergence order $(2^n + 2^{n-1} - 1 + \sqrt{2^{2n+1} + 2^{2n-2} + 2^n + 1})/2$. □

Remark 1. *According to Theorem 4, we conclude that the maximal order of n-parameter n-point Newton-type iterative method (34) with memory is* $(2^n + 2^{n-1} - 1 + \sqrt{2^{2n+1} + 2^{2n-2} + 2^n + 1})/2$. *The variable parameters* $L_{k,i}$ *of method (34) can be designed by simple interpolation polynomial, but this will decrease the order of convergence of method (34). Therefore, we do not discuss the low-order interpolation polynomial in this paper. For* $n = 2$, *the order of method (34) with memory is* $r = (5 + \sqrt{41})/2 \approx 5.701$. *For* $n = 3$, *the order of (34) with memory is* $r = (11 + \sqrt{153})/2 \approx 11.684$.

4. Basins of Attraction

The basins of attraction can be applied to analyze the stability of the multipoint iterative method [23–32], which will help us to select the iterative schemes whose behaviors are better qualitatively. Figures 1–5 show the basins of attraction of different methods. Our methods (13) (n = 3) and (34) (n = 2, 3) are compared with methods (2) (n = 3), (3) (n = 3) , (4) (n = 3), (73) and (76) for solving complex equations $z^n - 1 = 0$, $(n = 2, 3, 4, 5, 6)$. The field $D = [-5.0, 5.0] \times [-5.0, 5.0] \in C$ is divided into a grid of 500×500 in Figures 1–5. The initial point z_0 will be painted with black after 25 iterations. If the sequence generated by the iterative method reaches a zero of the polynomial , the initial point z_0 will be painted in a color previously selected for this zero. In the same basin of attraction, the number of iterations needed to achieve the solution is shown in darker or brighter colors (the less iterations, the brighter the color). The tolerance is $|z - z^*| < 10^{-3}$ in programs. Tables 1–5 show the average number of iterations(ANI) and the percentage of points (POP) which guarantee the convergence to the roots of complex equations $z^n - 1 = 0$,$(n = 2, 3, 4, 5, 6)$.

Table 1. Numerical results of different methods for $z^2 - 1 = 0$.

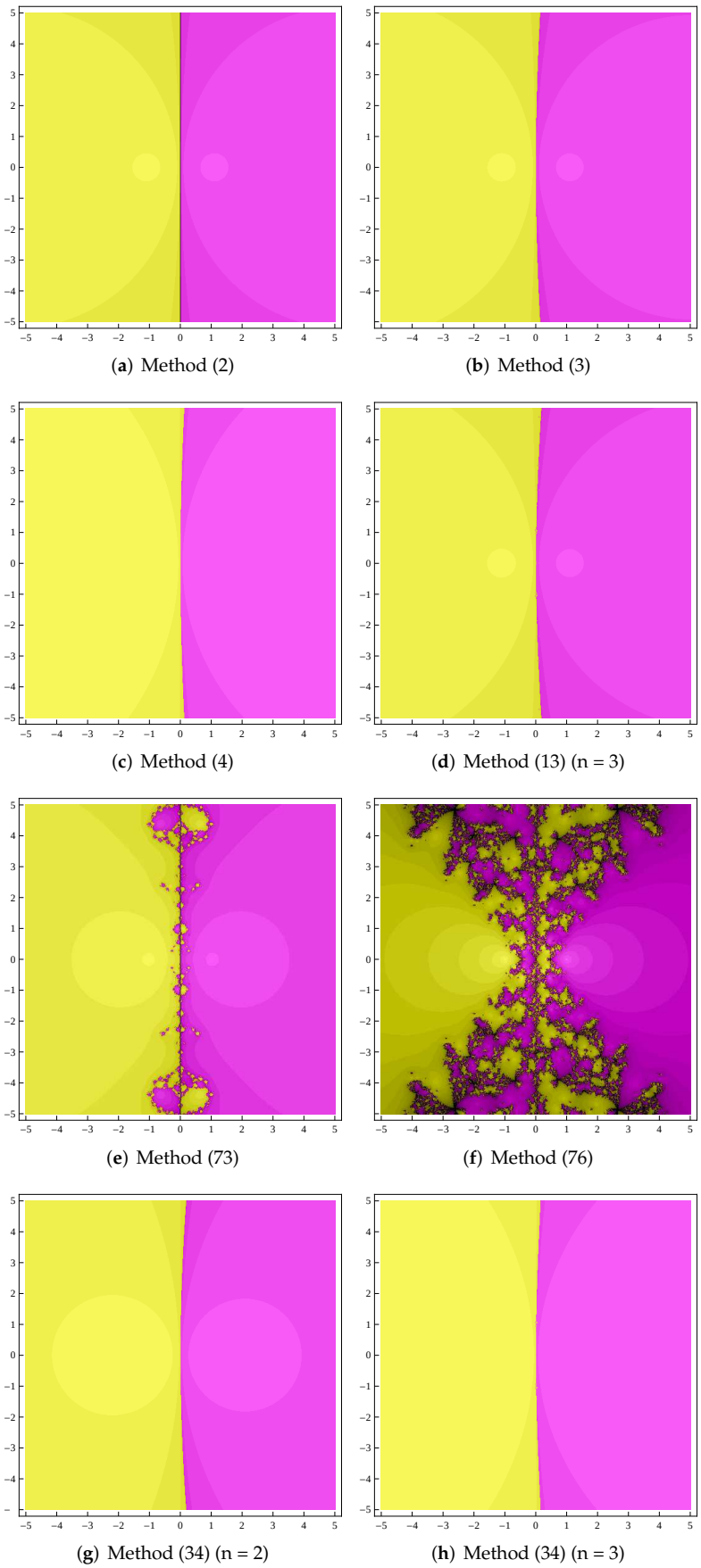
Methods	**(2)**	**(3)**	**(4)**	**(13) (n = 3)**	**(73)**	**(76)**	**(34) (n = 2)**	**(34) (n = 3)**
POP	99.80 %	100%	100%	100%	99.80%	96.02%	100%	100%
ANI	2.2781	2.2415	1.1430	2.2233	3.4107	11.401	1.8806	1.1371

Table 2. Numerical results of different methods for $z^3 - 1 = 0$.

Methods	**(2)**	**(3)**	**(4)**	**(13) (n = 3)**	**(73)**	**(76)**	**(34) (n = 2)**	**(34) (n = 3)**
POP	100 %	99.99%	100%	100%	99.62%	77.53%	100%	100%
ANI	2.6354	2.6416	1.3591	2.6386	4.4523	15.669	2.1994	1.3430

Table 3. Numerical results of different methods for $z^4 - 1 = 0$.

Methods	**(2)**	**(3)**	**(4)**	**(13) (n = 3)**	**(73)**	**(76)**	**(34) (n = 2)**	**(34) (n = 3)**
POP	99.60 %	100%	100%	100%	97.10%	63.42%	100%	100%
ANI	3.5018	3.4208	2.1952	3.4200	6.3275	18.667	2.5931	1.9959



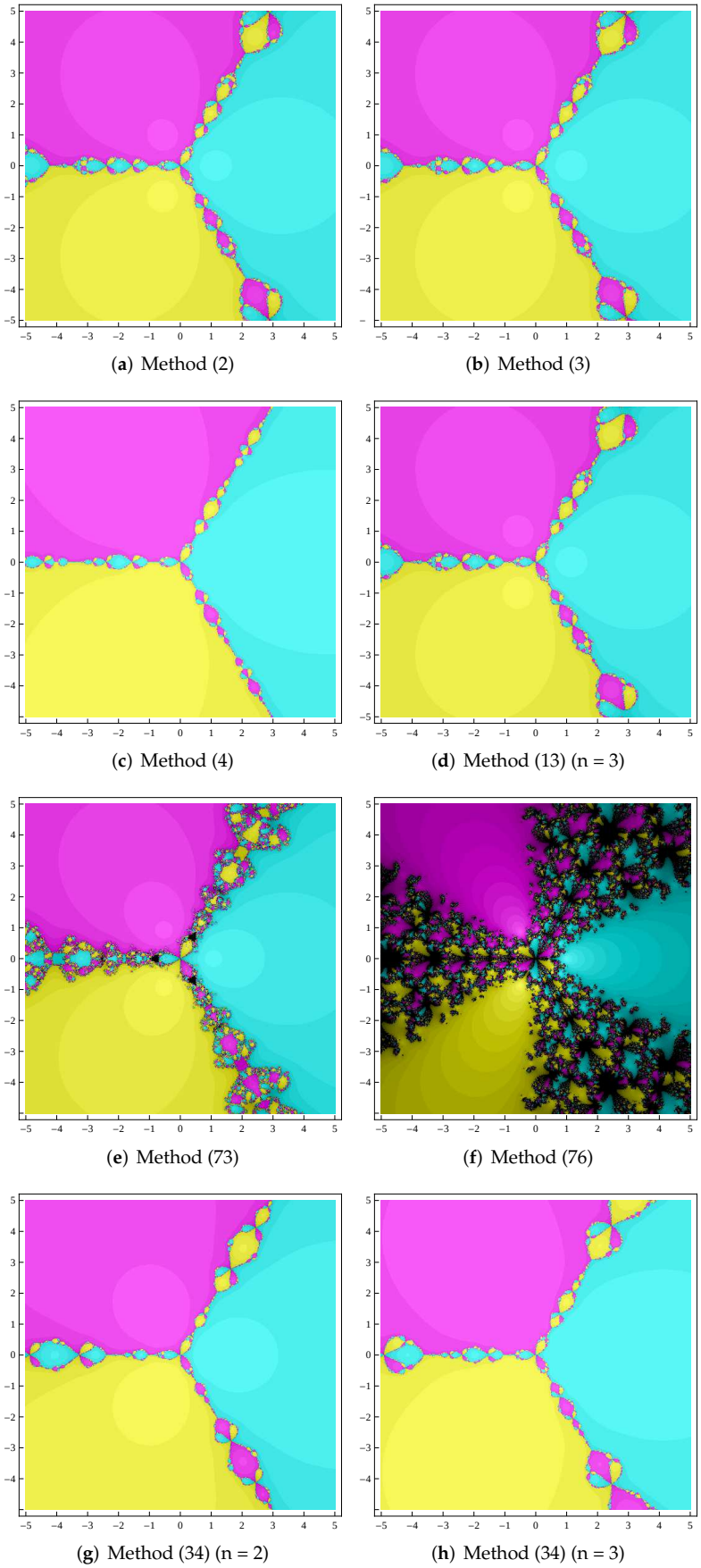
(**a**) Method (2) (**b**) Method (3)

(**c**) Method (4) (**d**) Method (13) (n = 3)

(**e**) Method (73) (**f**) Method (76)

(**g**) Method (34) (n = 2) (**h**) Method (34) (n = 3)

Figure 1. Dynamical planes for $z^2 - 1 = 0$.



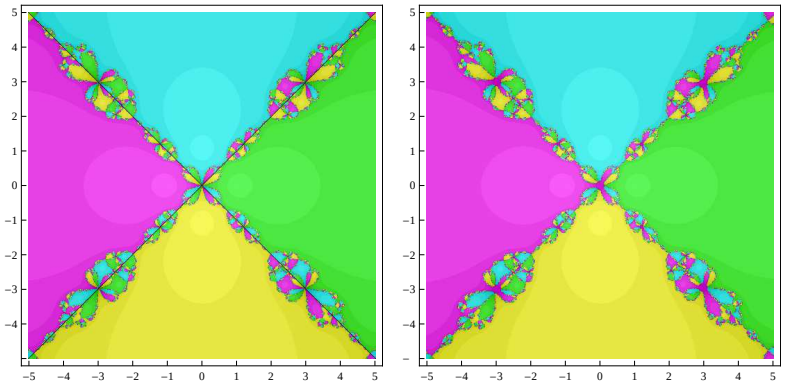
(**a**) Method (2) (**b**) Method (3)

(**c**) Method (4) (**d**) Method (13) (n = 3)

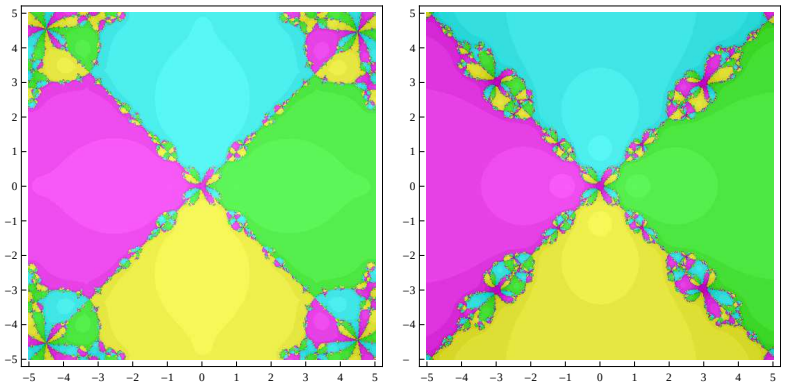
(**e**) Method (73) (**f**) Method (76)

(**g**) Method (34) (n = 2) (**h**) Method (34) (n = 3)

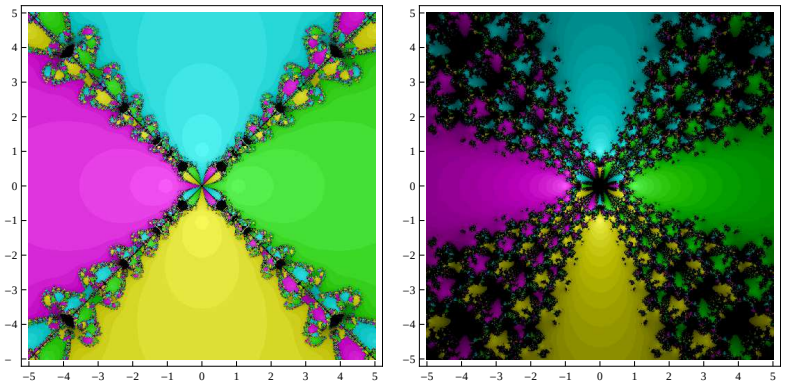
Figure 2. Dynamical planes for $z^3 - 1 = 0$.



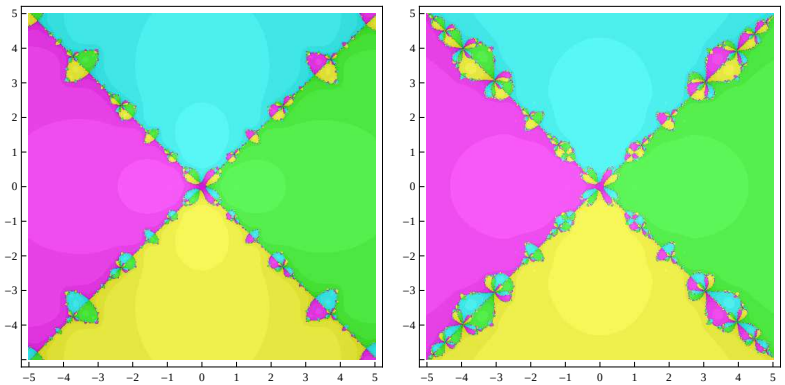
(**a**) Method (2) (**b**) Method (3)



(**c**) Method (4) (**d**) Method (13) (n = 3)

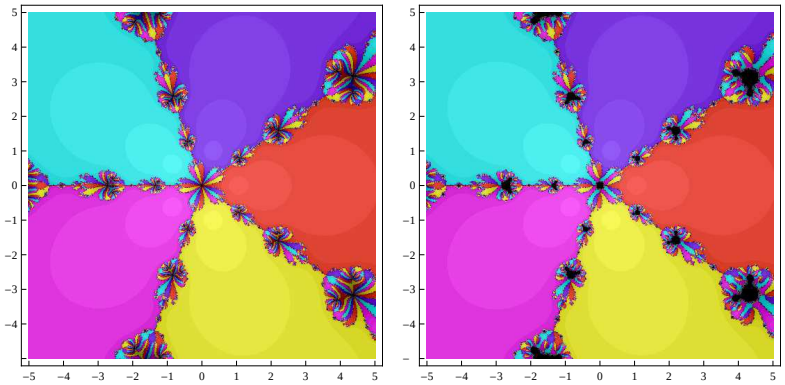


(**e**) Method (73) (**f**) Method (76)

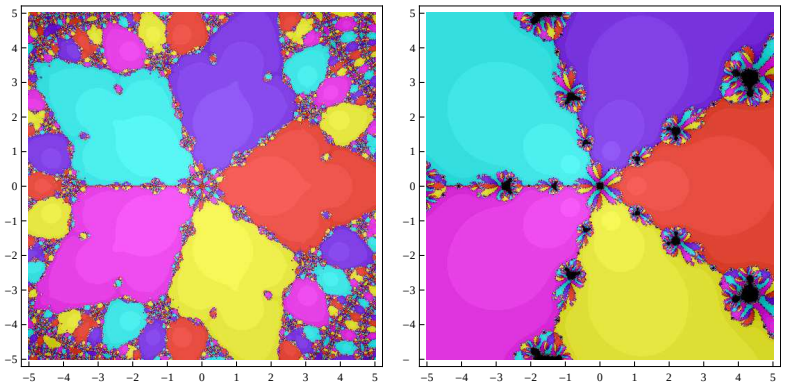


(**g**) Method (34) (n = 2) (**h**) Method (34) (n = 3)

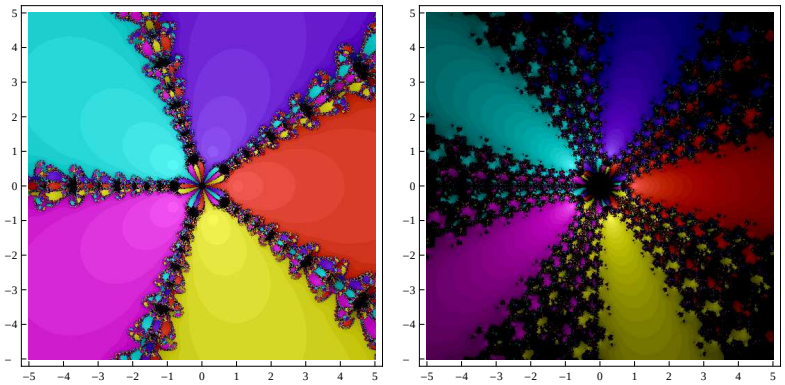
Figure 3. Dynamical planes for $z^4 - 1 = 0$.



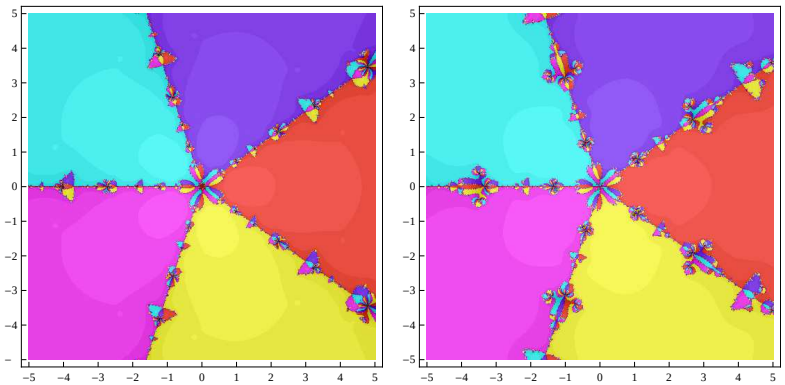
(**a**) Method (2) (**b**) Method (3)



(**c**) Method (4) (**d**) Method (13) (n = 3)

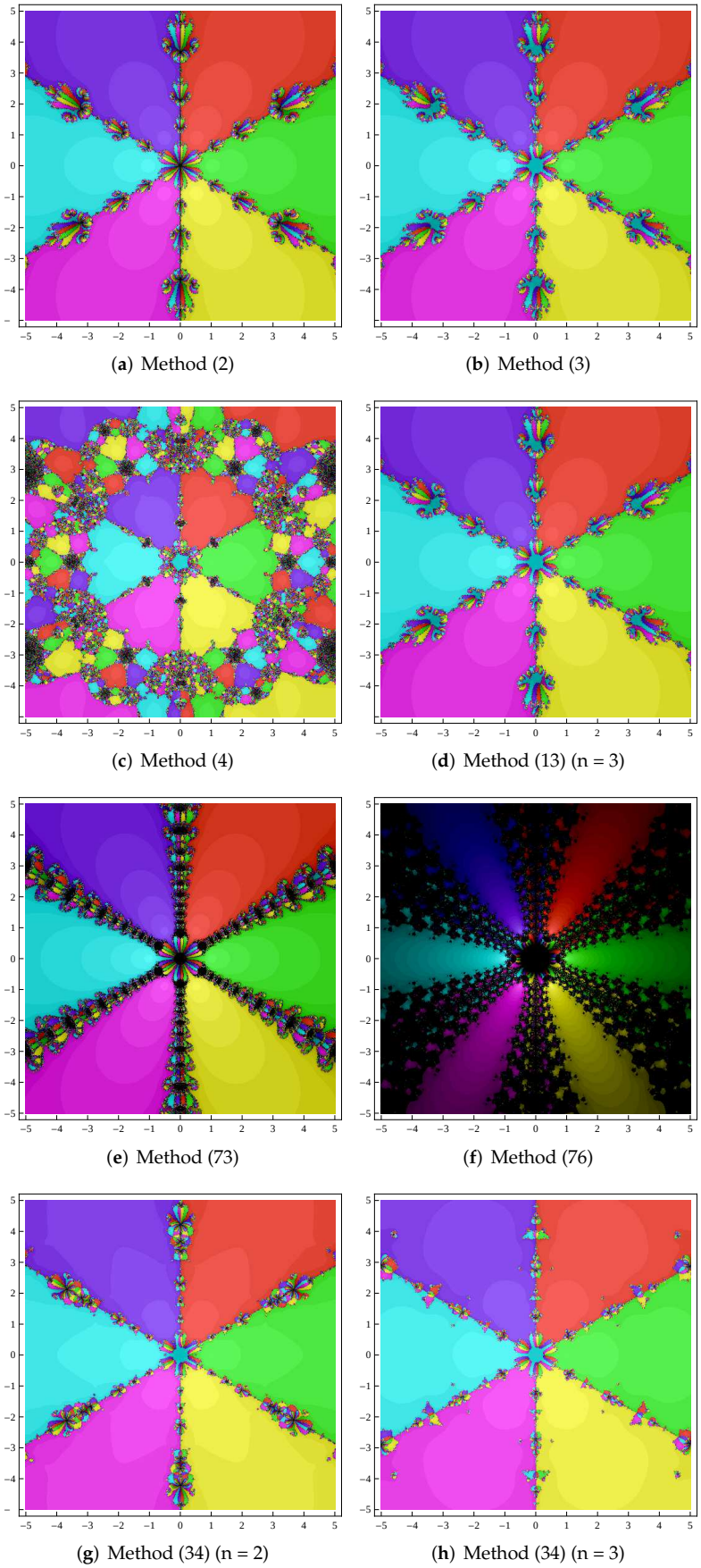


(**e**) Method (73) (**f**) Method (76)



(**g**) Method (34) (n = 2) (**h**) Method (34) (n = 3)

Figure 4. Dynamical planes for $z^5 - 1 = 0$.



(**a**) Method (2) (**b**) Method (3)

(**c**) Method (4) (**d**) Method (13) (n = 3)

(**e**) Method (73) (**f**) Method (76)

(**g**) Method (34) (n = 2) (**h**) Method (34) (n = 3)

Figure 5. Dynamical planes for $z^6 - 1 = 0$.

Table 4. Numerical results of different methods for $z^5 - 1 = 0$.

Methods	**(2)**	**(3)**	**(4)**	**(13) (n = 3)**	**(73)**	**(76)**	**(34) (n = 2)**	**(34) (n = 3)**
POP	99.83 %	98.34%	99.98%	98.35%	95.75%	61.57%	99.99%	100%
ANI	4.2132	4.3115	3.5273	4.3119	6.9659	19.920	2.7746	2.2169

Table 5. Numerical results of different methods for $z^6 - 1 = 0$.

Methods	**(2)**	**(3)**	**(4)**	**(13) (n = 3)**	**(73)**	**(76)**	**(34) (n = 2)**	**(34) (n = 3)**
POP	99.60 %	99.98%	98.09%	99.98%	94.27%	53.35%	99.92%	99.99%
ANI	4.6530	4.5337	5.1056	4.5347	7.8804	21.428	3.3563	2.4707

Figures 1–5 show that iterative methods without memory (2), (3) and (13) have similar convergence behavior. The black areas of basins of attraction for methods (73) and (76) are larger than in other methods. This implies that the convergence behavior of iterative methods with memory (73) and (76) are poor. The basins of attraction for method (34) with memory are brighter than the other methods with and without memory. This means that iterative method (34) requires a lesser number of iterations than other methods. Tables 1–5 show that, compared with other methods, our method (34) (n = 3) has the highest percentage of points which guarantee the convergence to the roots of complex equations and requires less number of iterations than various well-known methods. Thus, our method (34) (n = 3) has good stability for solving simple nonlinear equations.

5. Numerical Examples

The general n-parameter n-point Newton-type methods (13) and (34) are compared with Petković's two-step Newton-type method with memory [12] (73), Wang's Newton-type method with memory [33] (76), Petković's n-point Newton-type method (2) without memory and Wang's n-point Newton-type methods (3) and (4) for solving nonlinear functions.

Petković's two-step Newton-type iterative method with memory is calculated as follows [12]:

$$\begin{cases} y_k = \phi(y_{k-1}) f(t_k)^2 + M(t_k), \\ t_{k+1} = \phi(y_k) f(t_k)^2 + M(t_k), \end{cases} \tag{73}$$

where

$$M(t_k) = t_k - \frac{f(t_k)}{f'(t_k)}, \tag{74}$$

$$\phi(t) = \left(\frac{t - t_k}{f(t) - f(t_k)} - \frac{1}{f'(t_k)} \right) \frac{1}{f(t) - f(t_k)}. \tag{75}$$

Wang's Newton-type iterative method with memory is calculated as follows [33]:

$$\begin{cases} z_k = t_k - \dfrac{f(t_k)}{f'(t_k)}, \\ y_k = t_k - \dfrac{(z_k - t_k)}{1 - L_k(z_k - t_k)}, \\ t_{k+1} = y_k - \dfrac{f(y_k) f'(t_k)}{f[t_k, y_k]^2}, \end{cases} \tag{76}$$

where $L_k = \frac{t_k - z_{k-1}}{(z_k - t_{k-1})(t_k - t_{k-1})}$.

Iterative methods are applied to solve the following nonlinear equations:

$$\begin{array}{lll} f_1(t) = \cos(t) + t^2 - te^t = 0, & a \approx 0.63915409633200758, & t_0 = 0.5, \\ f_2(t) = t^4 + log(t) - 5 = 0, & a \approx 1.4658939193282127, & t_0 = 1.6, \\ f_3(t) = -\sin(t) + 9t^2 - 1 = 0, & a \approx 0.3918469070026, & t_0 = 0.3. \\ f_4(t) = t - \pi - t^3 sin(t) = 0, & a \approx 3.1415926535898, & t_0 = 2.9. \end{array}$$

The initial parameters $L = L_1 = 0.01$ and $L_{k,i} = 0.01, (i = 0, 1, 2)$ are used in the the iterative methods (3), (4), (13), (34) (73) and (76). Tables 6–9 show the absolute error $|t_k - a|$ for the first four steps and the approximate computational order of convergence(ACOC)[34]:

$$R = \frac{ln(|t_{n+1} - t_n|/|t_n - t_{n-1}|)}{ln(|t_n - t_{n-1}|/|t_{n-1} - t_{n-2}|)}. \tag{77}$$

Table 6. Numerical results for $f_1(t)$.

Method	$\|t_1 - a\|$	$\|t_2 - a\|$	$\|t_3 - a\|$	$\|t_4 - a\|$	ACOC
(2) n = 2	0.51126×10^{-4}	0.97721×10^{-18}	0.13043×10^{-72}	0.41401×10^{-292}	4.0000003
(3) n = 2	0.54528×10^{-4}	0.13358×10^{-17}	0.48098×10^{-72}	0.80864×10^{-290}	4.0000000
(13) n = 2	0.52925×10^{-4}	0.11597×10^{-17}	0.26735×10^{-72}	0.75510×10^{-291}	4.0000000
(73)	0.44088×10^{-5}	0.64006×10^{-25}	0.44087×10^{-115}	0.32433×10^{-526}	4.5599449
(76)	0.27347×10^{-3}	0.29224×10^{-16}	0.13268×10^{-70}	0.60206×10^{-301}	4.2386945
(34) n = 2	0.52925×10^{-4}	0.21668×10^{-27}	0.48274×10^{-159}	0.86928×10^{-910}	5.7024880
(2) n = 3	0.40835×10^{-8}	0.24321×10^{-68}	0.38502×10^{-550}		8.0008692
(3) n = 3	0.45332×10^{-8}	0.61330×10^{-68}	0.68830×10^{-547}		8.0000000
(13) n = 3	0.44026×10^{-8}	0.47630×10^{-68}	0.89387×10^{-548}		8.0000000
(4) n = 3	0.45332×10^{-8}	0.14236×10^{-85}	0.62560×10^{-861}		10.004219
(34) n = 3	0.44026×10^{-8}	0.29405×10^{-104}	$0.86439 \times 10^{-1222}$		11.619740

Table 7. Numerical results for $f_2(t)$.

Method	$\|t_1 - a\|$	$\|t_2 - a\|$	$\|t_3 - a\|$	$\|t_4 - a\|$	ACOC
(2) n = 2	0.11793×10^{-3}	0.84626×10^{-16}	0.22447×10^{-64}	0.11110×10^{-258}	4.0000000
(3) n = 2	0.12130×10^{-3}	0.97704×10^{-16}	0.41138×10^{-64}	0.12928×10^{-257}	4.0000000
(13) n = 2	0.12145×10^{-3}	0.98370×10^{-16}	0.42339×10^{-64}	0.14529×10^{-257}	4.0000000
(73)	0.62678×10^{-5}	0.88468×10^{-24}	0.18867×10^{-109}	0.41780×10^{-500}	4.5599373
(76)	0.31113×10^{-3}	0.69098×10^{-15}	0.29916×10^{-64}	0.23336×10^{-273}	4.2360780
(34) n = 2	0.12145×10^{-3}	0.42142×10^{-24}	0.33957×10^{-142}	0.70177×10^{-815}	5.6961912
(2) n = 3	0.15535×10^{-7}	0.72017×10^{-63}	0.15361×10^{-505}		8.0000000
(3) n = 3	0.16390×10^{-7}	0.11729×10^{-62}	0.80666×10^{-504}		8.0000000
(13) n = 3	0.16405×10^{-7}	0.11831×10^{-62}	0.86592×10^{-504}		8.0000000
(4) n = 3	0.16390×10^{-7}	0.46896×10^{-79}	0.17682×10^{-794}		9.9998487
(34) n = 3	0.16405×10^{-7}	0.65259×10^{-95}	$0.93155 \times 10^{-1120}$		11.725876

Table 8. Numerical results for $f_3(t)$.

Method	$\|t_1 - a\|$	$\|t_2 - a\|$	$\|t_3 - a\|$	$\|t_4 - a\|$	ACOC
(2) n = 2	0.42633×10^{-3}	0.10988×10^{-12}	0.48611×10^{-51}	0.18620×10^{-204}	4.0000000
(3) n = 2	0.43209×10^{-3}	0.11749×10^{-12}	0.64401×10^{-51}	0.58130×10^{-204}	4.0000000
(13) n = 2	0.43241×10^{-3}	0.11792×10^{-12}	0.65398×10^{-51}	0.61858×10^{-204}	4.0000000
(73)	0.59903×10^{-4}	0.25674×10^{-18}	$0 .13907 \times 10^{-83}$	0.35294×10^{-381}	4.5597146
(76)	0.10103×10^{-2}	0.69855×10^{-12}	0.23785×10^{-50}	0.22107×10^{-213}	4.2381255
(34) n = 2	0.43241×10^{-3}	0.48374×10^{-19}	0.18668×10^{-114}	0.20222×10^{-656}	5.6801734
(2) n = 3	0.27207×10^{-6}	0.50023×10^{-51}	0.65317×10^{-409}		8.0000000
(3) n = 3	0.27947×10^{-6}	0.63668×10^{-51}	0.46203×10^{-408}		8.0000000
(13) n = 3	0.27977×10^{-6}	0.64293×10^{-51}	0.50004×10^{-408}		8.0000000
(4)n=3	0.27947×10^{-6}	0.14545×10^{-67}	0.21179×10^{-680}		10.000013
(34) n = 3	0.27977×10^{-6}	0.19008×10^{-76}	0.75443×10^{-913}		11.920006

Table 9. Numerical results for $f_4(t)$.

Method	$\|t_1 - a\|$	$\|t_2 - a\|$	$\|t_3 - a\|$	$\|t_4 - a\|$	ACOC
(2) n = 2	0.45791×10^{-2}	0.29033×10^{-9}	0.47507×10^{-38}	0.34058×10^{-153}	4.0000000
(3) n = 2	0.46989×10^{-2}	0.32946×10^{-9}	0.80651×10^{-38}	0.28961×10^{-152}	4.0000000
(13) n = 2	0.47013×10^{-2}	0.33026×10^{-9}	0.81472×10^{-38}	0.30172×10^{-152}	4.0000000
(73)	0.21843×10^{-2}	0.71702×10^{-12}	0.63014×10^{-55}	0.30506×10^{-251}	4.5595182
(76)	0.13208×10^{-1}	0.35938×10^{-8}	0.23138×10^{-35}	0.10762×10^{-150}	4.2415301
(34) n = 2	0.47013×10^{-2}	0.78035×10^{-14}	0.22231×10^{-82}	0.33152×10^{-472}	5.6871333
(2) n = 3	0.16926×10^{-4}	0.22742×10^{-38}	0.24157×10^{-309}		8.0000000
(3) n = 3	0.17852×10^{-4}	0.36590×10^{-38}	0.11400×10^{-307}		8.0000000
(13) n = 3	0.17864×10^{-4}	0.36802×10^{-38}	0.11943×10^{-307}		7.9999986
(4) n = 3	0.17852×10^{-4}	0.56501×10^{-49}	0.55091×10^{-494}		10.115357
(34) n = 3	0.17864×10^{-4}	0.38690×10^{-57}	0.18244×10^{-682}		11.873804

Tables 6–9 show that the numerical results coincide with the theory developed in this paper. Our n-parameter n-point Newton-type method (34) has higher convergence order and computational accuracy than the existing Newton-type methods. Method (34) greatly improves the convergence order of method (13) by using n variable parameters. The variable parameters $L_{k,i}$ in method (34) are designed by using iteration sequences from current and previous iterations, which do not increase the computational cost of the iterative method. This implies that our method (34) with memory posses a very high computational efficiency.

6. Conclusions

In order to improve the convergence order of Newton-type multipoint iterative method, a general n-parameter n-point Newton-type iterative method with memory is designed in this paper. Firstly, an n-parameter n-point Newton-type multipoint method (13) with optimal order 2^n is proposed. Based on method (13), a general n-parameter n-point Newton-type multipoint iterative method (34) with n variable parameters is proposed for

solving nonlinear equations. The maximal order of method (34) is superior to the existing Newton-type iterative methods with and without memory. The basins of attraction show that the proposed method (34) has the highest percentage of points, which guarantees the convergence to the roots of complex equations. The ANI of the proposed method (34) is less than that of other methods in Tables 6 to 9. This implies that the proposed method (34) has good stability. The numerical results shows that the proposed method (34) greatly improves the computational efficiency and convergence order of the Newton-type iterative method.

Funding: This research was supported by the National Natural Science Foundation of China (No. 61976027), the Liaoning Revitalization Talents Program (XLYC2008002), the Educational Commission Foundation of Liaoning Province of China (No. LJ2020015) and the University-Industry Collaborative Education Program (No. 202102030037).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: Not applicable.

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

References

1. Traub, J.F. *Iterative Methods for the Solution of Equations*; Prentice-Hall: Hoboken, NJ, USA, 1964.
2. Petković, M.S.; Neta, B.; Petković, L.; Džunić, J. *Multipoint Methods for Solving Nonlinear Equtions*; Academic Press: Boston, MA, USA, 2013.
3. Chun, C. Some fourth-order iterative methods for solving nonlinear equations. *Appl. Math. Comput.* **2008**, *195*, 454–459. [CrossRef]
4. Soleymani, F.; Sharifi, M.; Mousavi, B.S. An improvement of Ostrowski's and King's techniques with optimal convergence order eight. *J. Optim. Theo. Appl.* **2012**, *153*, 225–236. [CrossRef]
5. Geum, Y.H.; Kim, Y.I. A uniparametric family of three-step eighth-order multipoint iterative methods for simple roots. *Appl. Math. Lett.* **2011**, *24*, 929–935. [CrossRef]
6. Petković, M.S. On a general class of multipoint root-finding methods of high computational efficiency. *Siam. J. Numer. Anal.* **2010**, *47*, 4402–4414. [CrossRef]
7. Ren, H.; Wu, Q.; Bi, W. A class of two-step Steffensen type methods with fourth-order convergence. *Appl. Math. Comput.* **2009**, *209*, 206–210. [CrossRef]
8. Wang, X.; Zhu, M. Two Iterative Methods with Memory Constructed by the Method of Inverse Interpolation and Their Dynamics. *Mathematics* **2020**, *8*, 1080. [CrossRef]
9. 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]
10. Chun, C.; Neta, B. Certain improvements of Newton's method with fourth-order convergence. *Appl. Math. Comput.* **2009**, *215*, 821–828. [CrossRef]
11. Neta, B.; Petković, M.S. Construction of optimal order nonlinear solvers usnig inverse interplation. *Appl. Math. Comput.* **2010**, *217*, 2448–2455.
12. Petković, M.S.; Džunić, J.; Neta, B. Interpolatory multipoint methods with memory for solving nonlinear equations. *Appl. Math. Comput.* **2011**, *218*, 2533–2541. [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. Kung, H.T.; Traub, J.F. Optimal order of one-point and multipoint iteration. *J. Assoc. Comput. Math.* **1974**, *21*, 634–651. [CrossRef]
15. Zheng, Q.; Li, J.; Huang, F. An optimal Steffensen-type family for solving nonlinear equations. *Appl. Math. Comput.* **2011**, *217*, 9592–9597. [CrossRef]
16. Džunić, J.; Petković, M.S. On generalized biparametric multipoint root finding methods with memory. *J. Comput. Appl. Math.* **2014**, *255*, 362–375. [CrossRef]
17. Wang, X.; Zhang, T. Efficient *n*-point iterative methods with memory for solving nonlinear equations. *Numer. Algor.* **2015**, *70*, 357–375. [CrossRef]
18. Petković, M.S. Remarks on "On a general class of multipoint root-finding methods of high computational efficiency". *Siam. J. Numer. Anal.* **2011**, *49*, 1317–1319
19. Cordero, A.; Torregrosa, J. R.; Triguero-Navarro, P.; A general optimal iterative scheme with arbitrary order of convergence. *Symmetry* **2021**, *13*, 884. [CrossRef]
20. Wang, X.; Qin, Y.; Qian, W.; Zhang, S.; Fan, X.; A family of newton type iterative methods for solving nonlinear equations. *Algorithms* **2015**, *8*, 786–798. [CrossRef]
21. Jelley, C.T. *Solving Nonlinear Equations with Newton's Method*; SIAM: Philadelphia, PA, USA, 2003.

22. Ortega, J.M.; Rheinbolt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*; Academic Press: New York, NY, USA, 1970.
23. Wang, X.; Chen, X. The Dynamical Analysis of a Biparametric Family of Six-Order Ostrowski-Type Method under the Möbius Conjugacy Map. *Fractal Fract.* **2022**, *6*, 174. [CrossRef]
24. Wang, X.; Chen, X. Derivative-Free Kurchatov-Type Accelerating Iterative Method for Solving Nonlinear Systems: Dynamics and Applications. *Fractal Fract.* **2022**, *6*, 59. [CrossRef]
25. Sharma, D.; Argyros, I.K.; Parhi, S.K.; Sunanda, S.K. Local Convergence and Dynamical Analysis of a Third and Fourth Order Class of Equation Solvers. *Fractal Fract.* **2021**, *5*, 27. [CrossRef]
26. Behl, R.; Cordero, A.; Torregrosa, J. R. High order family of multivariate iterative methods: Convergence and stability. *J. Comput. Appl. Math.* **2020**, *405*, 113053. [CrossRef]
27. Neta, B.; Scott, M.; Chun, C. Basin attrators for various methods for multiple roots. *Appl. Math. Comput.* **2012**, *218*, 5043–5066.
28. Argyros, I.K.; Magreñán, Á.A. *Iterative Methods and their Dynamics with Applications: A Contemporary Study*; CRC Press: Boca Raton, FL, USA, 2017.
29. Galilea, V.; Gutiéreez, J.M. A Characterization of the Dynamics of Schröder's Method for Polynomials with Two Roots. *Fractal Fract.* **2021**, *5*, 25. [CrossRef]
30. Susanto, H., Karjanto, N. Newton's method's basins of attraction revisited. *Appl. Math. Comput.* **2009**, *215*, 1084–1090. [CrossRef]
31. Mallawi, F.O.; Behl, R.; Maroju, P. On Global Convergence of Third-Order Chebyshev-Type Method under General Continuity Conditions. *Fractal Fract.* **2022**, *6*, 46. [CrossRef]
32. Ardelean, G. A comparison between iterative methods by using the basins of attraction. *Appl. Math. Comput.* **2011**, *218*, 88–95. [CrossRef]
33. Wang, X. A new accelerating technique applied to a variant of Cordero-Torregrosa method. *J. Comput. Appl. Math.* **2018**, *330*, 695–709. [CrossRef]
34. Cordero, A.; Torregrosa, J.R. Variants of Newton's Method using fifth-order quadrature foumulas. *Appl. Math. Comput.* **2007**, *190*, 686–698.