Journal of Applied Mathematics and Physics, 2023, 11, 746-759
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

Two Families of Multipoint Root-Solvers Using Inverse Interpolation with Memory

Zhongli Liu[1]*, Quan Zheng[2]

[1]Institute of Mathematics and Physics, Beijing Union University, Beijing, China
[2]College of Sciences, North China University of Technology, Beijing, China
Email: *shtzhongli@buu.edu.cn

How to cite this paper: Liu, Z.L. and Zheng, Q. (2023) Two Families of Multipoint Root-Solvers Using Inverse Interpolation with Memory. *Journal of Applied Mathematics and Physics*, **11**, 746-759.
https://doi.org/10.4236/jamp.2023.113050

Received: February 4, 2023
Accepted: March 25, 2023
Published: March 28, 2023

Abstract

In this paper, a general family of derivative-free $n + 1$-point iterative methods using $n + 1$ evaluations of the function and a general family of n-point iterative methods using n evaluations of the function and only one evaluation of its derivative are constructed by the inverse interpolation with the memory on the previous step for solving the simple root of a nonlinear equation. The order $\left(2^{n+1}-1+\sqrt{2^{2n+2}+1}\right)/2$ and order $\left(3\cdot 2^{n-1}-1+\sqrt{9\cdot 2^{2n-2}-2^n+1}\right)/2$ of convergence of them are proved respectively. Finally, the proposed methods and the basins of attraction are demonstrated by the numerical examples.

Keywords

Nonlinear Equation, General Multipoint Iteration, Inverse Interpolation, Order of Convergence, Basin of Attraction

1. Introduction

Newton's method is a well-known iterative method to solve nonlinear problems in scientific computation. For a nonlinear equation $f(x)=0$, Newton's method is as the following (see [1]):

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)},\ k=0,1,\cdots.$$

Furthermore, Steffensen's method is a derivative-free iterative method, and a self-accelerating Steffensen's method is introduced in Traub's book ([2]) as the following:

$$x_{k+1} = x_k - \frac{f(x_k)}{f\left[x_k, x_k + \beta_k f(x_k)\right]}, k = 0,1,\cdots, \quad (1)$$

where $f[u,v] := \frac{f(u) - f(v)}{u - v}$ and $\beta_k = \frac{1}{f\left[x_{k-1}, x_{k-1} + \beta_{k-1} f(x_{k-1})\right]}$ by using recursively the memory on the previous step without any new functional evaluation.

The efficiency index of an iterative method (IM) is defined as $E = p^{1/d}$ where p is the order of IM and d is the number of function evaluations of IM per step. Kung and Traub conjectured in 1974 that a multipoint iteration based on $n + 1$ evaluations without memory has optimal order 2^n of convergence [3]. Thus, Newton's method and Steffensen's method are methods of optimal second-order, and their efficiency indices are both $\sqrt{2} = 1.4142$. Self-accelerating Steffensen's method achieves super convergence of order $1 + \sqrt{2} = 2.4142$ with memory, and its efficiency index is $\sqrt{1 + \sqrt{2}} = 1.5538$. A one-parameter multipoint iteration of optimal order 16 in [4] [5] consists of successively a Newton substep, a modified Newton substep and two substeps of inverse interpolation, requires four evaluations of f and only one evaluation of f' per step, and reaches the efficiency index $16^{1/5} = 1.7411$. General multipoint iterations of optimal order have been constructed by using inverse interpolation in [3] [6] [7] and direct interpolation in [5] [7]-[12], and reach the efficiency index $2^{\frac{n}{n+1}}$ by $n + 1$ evaluations without memory. Furthermore, self-accelerations of general multipoint iterations with memory from the current and previous iterations can achieve better convergence and efficiency [5] [7] [10] [11] [12].

Recently, the family of $n + 1$-point iterative methods of optimal order 2^n with $n + 1$ self-accelerating parameters was proposed by using Newton's interpolation in [12] as follows:

$$\begin{cases} x_{k,1} = x_{k,0} + \beta_0 f(x_{k,0}), x_{k,0} = x_k, \\ x_{k,2} = x_{k,1} - \dfrac{f(x_{k,1})}{f\left[x_{k,1}, x_{k,0}\right] + \beta_1 f(x_{k,0})}, \\ x_{k,j+1} = x_{k,j} - \dfrac{f(x_{k,j})}{f\left[x_{k,j}, x_{k,j-1}\right] + \cdots + f[x_{k,j}, \cdots, x_{k,0}](x_{k,j} - x_{k,j-1}) \cdots (x_{k,j} - x_{k,1}) + \beta_j (x_{k,j} - x_{k,j-1}) \cdots (x_{k,j} - x_{k,0})}, \\ \qquad j = 2, \cdots, n, \\ x_{k+1} = x_{k,n+1}, k = 0,1,\cdots, \end{cases} \quad (2)$$

where $f[u,v,w] := \frac{f[u,v] - f[v,w]}{u - w}, \cdots$. This family generalized the two-point two-parameter Steffensen's method in [13] and the general parametric families in [7] [9] [10] [11] by using $n + 1$ parameters, and achieved the convergence of order $\left(2^{n+1} - 1 + \sqrt{2^{2(n+1)} + 1}\right)/2$ by using the parameters with the memory on all points in the previous step as the following:

$$\begin{cases} \beta_0 = -1 \big/ N'_{n+1}\left(x_{k,0}\right), \\ \beta_1 = -N''_{n+2}\left(x_{k,1}\right) \big/ \left(2N'_{n+2}\left(x_{k,1}\right)\right), \\ \beta_i = N^{(j+1)}_{n+j+1}\left(x_{k,j}\right) \big/ (j+1)!,\ j = 2,\cdots,n, \end{cases} \tag{3}$$

where $N_{n+j+1}(t)$ is a Newton's interpolating polynomial of order $n+j+1$ for $j = 0,\cdots,n$, such that $N_{n+j+1}\left(x_{k,i}\right) = f\left(x_{k,i}\right)\left(0 \le i \le j\right)$ and $N_{n+j+1}\left(x_{k-1,i}\right) = f\left(x_{k-1,i}\right)\left(0 \le i \le n\right)$. When $n = 4$, the efficiency index of (2) without memory is also $16^{1/5} = 1.7411$ and the efficiency index of (2) and (3) with memory is $\left\{\left(31+\sqrt{1025}\right)\big/2\right\}^{1/5} = 1.9938$. The topic on basins of attraction was addressed in [14]-[20] for derivative-free methods and various other methods.

In this paper, a general family of *n* + 1-point iterative methods derivative-free and another general family of *n*-point iterative methods using a first derivative are constructed by the inverse interpolation with the memory on points in the previous step in Sections 2 and 3 respectively, the proposed families are tested by numerical examples for solving nonlinear equations and the basins of attraction of the methods are illustrated in Section 4, and finally conclusions are made.

2. General *n* + 1-Point Iteration Derivative-Free with Memory

Let $x_{k,0} = x_k$ be an approximation of the simple root of $f(x)$ and $x_{k,1} = x_{k,0} + \gamma_0 f\left(x_{k,0}\right)$ be a further approximation. Let us recognize the mapping *f* previously from the independent variable to the dependent variable inversely as a mapping f^{-1} now from the dependent variable to the independent variable in the obtained discrete information. We can have an inverse Newton's interpolating polynomial of degree one (see, e.g., [2] [3]):

$$Q_1(t) = x_{k,1} + f^{-1}\left[f\left(x_{k,1}\right), f\left(x_{k,0}\right)\right]\left(t - f\left(x_{k,1}\right)\right),$$

such that $Q_1\left(f\left(x_{k,1}\right)\right) = x_{k,1}$ and $Q_1\left(f\left(x_{k,0}\right)\right) = x_{k,0}$, where $f^{-1}\left[f(u), f(v)\right] = \dfrac{u-v}{f(u)-f(v)}$. The next approximation of the root could be obtained by $Q_1(0)$ as the following:

$$\begin{aligned} x_{k+1} = Q_1(0) &= x_{k,0} + \gamma_0 f\left(x_{k,0}\right) - \frac{\gamma_0 f\left(x_{k,0}\right) f\left(x_{k,1}\right)}{f\left(x_{k,1}\right) - f\left(x_{k,0}\right)} \\ &= x_{k,0} - \frac{f\left(x_{k,0}\right)}{f\left[x_{k,0}, x_{k,0} + \gamma_0 f\left(x_{k,0}\right)\right]}, \end{aligned}$$

which can give Steffensen's method and self-accelerating Steffensen's method obviously.

However, by using the most information up to the previous step, *i.e.*, using the known discrete information of f^{-1} in **Table 1** when $n = 1$.

We have the inverse Newton's interpolating polynomial of degree three as the following:

Table 1. The known discrete information of f^{-1} for $j=1,\cdots,n$ and $k>0$.

$f(x_{k,j})$	$\cdots$	$f(x_{k,0})$	$f(x_{k-1,n})$	$\cdots$	$f(x_{k-1,0})$
$x_{k,j}$	$\cdots$	$x_{k,0}$	$x_{k-1,n}$	$\cdots$	$x_{k-1,0}$

$$\begin{aligned}Q_3(t) = {} & x_{k,1} + f^{-1}\left[f(x_{k,1}), f(x_{k,0})\right](t - f(x_{k,1})) \\ & + f^{-1}\left[f(x_{k,1}), f(x_{k,0}), f(x_{k-1,1})\right](t - f(x_{k,1}))(t - f(x_{k,0})) \\ & + f^{-1}\left[f(x_{k,1}), f(x_{k,0}), f(x_{k-1,1}), f(x_{k-1,0})\right](t - f(x_{k,1})) \\ & \times (t - f(x_{k,0}))(t - f(x_{k-1,1})),\end{aligned}$$

such that $Q_3(f(x_{k,1})) = x_{k,1}$, $Q_3(f(x_{k,0})) = x_{k,0}$, $Q_3(f(x_{k-1,1})) = x_{k-1,1}$ and $Q_3(f(x_{k-1,0})) = x_{k-1,0}$, where $f^{-1}[f(u), f(v)] = \dfrac{u-v}{f(u)-f(v)}$ and $f^{-1}[f(u), f(v), f(w)] = \dfrac{f^{-1}[f(u), f(v)] - f^{-1}[f(u), f(w)]}{f(v)-f(w)}$ and so forth. $Q_3(0)$ could be better than $Q_1(0)$ to approximate the root of $f(x)$. We suggest that $x_{k+1} = Q_3(0)$ and propose a derivative-free iteration as the following:

$$\begin{cases} x_{k,0} = x_k, x_{k,1} = x_{k,0} + \gamma_0 f(x_{k,0}), \\ x_{k+1} = x_{k,2} = x_{k,1} + f^{-1}\left[f(x_{k,1}), f(x_{k,0})\right](-f(x_{k,1})) \\ \qquad + f^{-1}\left[f(x_{k,1}), f(x_{k,0}), f(x_{k-1,1})\right](-f(x_{k,1}))(-f(x_{k,0})) \\ \qquad + f^{-1}\left[f(x_{k,1}), f(x_{k,0}), f(x_{k-1,1}), f(x_{k-1,0})\right](-f(x_{k,1})) \\ \qquad \times (-f(x_{k,0}))(-f(x_{k-1,1})). \end{cases}$$

Furthermore, by the inverse Newton's interpolating polynomial of degree $n+j+1$ satisfying Table 1, we construct an optimal family of $n+1$-point iterations with memory as the following:

$$\begin{cases} x_{k,0} = x_k, \quad x_{k,1} = x_{k,0} + \gamma_0 f(x_{k,0}), \\ x_{k,j+1} = x_{k,j} + f^{-1}\left[f(x_{k,j}), f(x_{k,j-1})\right](-f(x_{k,j})) + \cdots \\ \qquad + f^{-1}\left[f(x_{k,j}), \cdots, f(x_{k,0})\right](-f(x_{k,j}))\cdots(-f(x_{k,1})) \\ \qquad + f^{-1}\left[f(x_{k,j}), \cdots, f(x_{k,0}), f(x_{k-1,n})\right](-f(x_{k,j}))\cdots(-f(x_{k,0})) \\ \qquad + \cdots + f^{-1}\left[f(x_{k,j}), \cdots, f(x_{k,0}), f(x_{k-1,n}), \cdots, f(x_{k-1,0})\right] \\ \qquad \times (-f(x_{k,j}))\cdots(-f(x_{k,0}))(-f(x_{k-1,n}))\cdots(-f(x_{k-1,1})), \; j = 1, \cdots, n, \\ x_{k+1} = x_{k,n+1}, \quad k = 0, 1, \cdots, \end{cases} \tag{4}$$

where γ_0 is a constant.

Theorem 1. *Let $f: D \to \Re$ be a sufficiently differentiable function with a simple root $a \in D$, $D \subset \Re$ be an open set, x_k be close enough to a, then the family (4) satisfies the error equation*

$$e_{k+1}=-\frac{\left(f^{-1}\right)^{(2n+2)}(0)}{(2n+2)!}\left(-f'(a)\right)^{2n+2}\prod_{i=2}^{n}\left[\left(-\frac{\left(f^{-1}\right)^{(n+i+1)}(0)}{(n+i+1)!}\right)\left(-f'(a)\right)^{n+i+1}\right]^{2^{n-i}} \tag{5}$$
$$\times\left(1+\gamma_0 f'(a)\right)^{2^{n-1}}\left(e_k^2 e_{k-1,n}\cdots e_{k-1,0}\right)^{2^{n-1}}+o\left(\left(e_k^2 e_{k-1,n}\cdots e_{k-1,0}\right)^{2^{n-1}}\right),$$

where $e_k=x_k-a$ *and* $e_{k,j}=x_{k,j}-a, k=0,1,\cdots,$ *and achieves the convergence of order at least* $\left(3\cdot 2^{n-1}-1+\sqrt{9\cdot 2^{2n-2}+2^n+1}\right)/2$.

Proof. Supposed that $e_{k,j}=C_j e_k^{p_j}+o\left(e_k^{p_j}\right)$ and $e_{k+1}=Ce_k^r+o\left(e_k^r\right)$. Then,

$$e_{k,j}=C_jC^{p_j}e_{k-1}^{rp_j}+o\left(e_{k-1}^{rp_j}\right), e_{k+1}=C^{r+1}e_{k-1}^{r^2}+o\left(e_{k-1}^{r^2}\right),$$

$$e_{k,1}=x_{k,1}-a=e_{k,0}+\gamma_0 f\left[x_{k,0},a\right]e_{k,0}=\left(1+\gamma_0 f\left[x_{k,0},a\right]\right)e_k$$
$$=\left(1+\gamma_0 f'(a)\right)Ce_{k-1}^r+o\left(e_{k-1}^r\right).$$

Noticing the definition of divided difference, for $j=1,\cdots,n$, we have

$$e_{k,j+1}=-f^{-1}\left[f(a),f\left(x_{k,j}\right),\cdots,f\left(x_{k,0}\right),f\left(x_{k-1,n}\right),\cdots,f\left(x_{k-1,0}\right)\right]$$
$$\times(-1)^{n+j+2}f\left[x_{k,j},a\right]\cdots f\left[x_{k,0},a\right]f\left[x_{k-1,n},a\right]\cdots f\left[x_{k-1,0},a\right]e_{k,j}\cdots e_{k,0}e_{k-1,n}\cdots e_{k-1,0}$$
$$-f^{-1}\left[f(a),f\left(x_{k,j}\right),\cdots,f\left(x_{k,0}\right),f\left(x_{k-1,n}\right),\cdots,f\left(x_{k-1,0}\right)\right]$$
$$\times(-1)^{n+j+2}f\left[x_{k,j},a\right]\cdots f\left[x_{k,0},a\right]f\left[x_{k-1,n},a\right]\cdots f\left[x_{k-1,0},a\right]$$
$$\times\left(-f^{-1}\left[f(a),f\left(x_{k,j-1}\right),\cdots,f\left(x_{k,0}\right),f\left(x_{k-1,n}\right),\cdots,f\left(x_{k-1,0}\right)\right]\right)$$
$$\times(-1)^{n+j+1}f\left[x_{k,j-1},a\right]\cdots f\left[x_{k,0},a\right]f\left[x_{k-1,n},a\right]\cdots f\left[x_{k-1,0},a\right]$$
$$\times\cdots\times\left(-f^{-1}\left[f(a),f\left(x_{k,1}\right),f\left(x_{k,0}\right),f\left(x_{k-1,n}\right),\cdots,f\left(x_{k-1,0}\right)\right]\right)^{2^{j-2}}$$
$$\times(-1)^{n+3}\left(f\left[x_{k,1},a\right]f\left[x_{k,0},a\right]f\left[x_{k-1,n},a\right]\cdots f\left[x_{k-1,0},a\right]\right)^{2^{j-2}}\left(e_{k,1}e_{k,0}e_{k-1,n}\cdots e_{k-1,0}\right)^{2^{j-1}}$$
$$=-f^{-1}\left[f(a),f\left(x_{k,j}\right),\cdots,f\left(x_{k,0}\right),f\left(x_{k-1,n}\right),\cdots,f\left(x_{k-1,0}\right)\right]$$
$$\times(-1)^{n+j+2}f\left[x_{k,j},a\right]\cdots f\left[x_{k,0},a\right]f\left[x_{k-1,n},a\right]\cdots f\left[x_{k-1,0},a\right]$$
$$\times\cdots\times\left(-f^{-1}\left[f(a),f\left(x_{k,1}\right),f\left(x_{k,0}\right),f\left(x_{k-1,n}\right),\cdots,f\left(x_{k-1,0}\right)\right]\right)^{2^{j-2}}$$
$$\times\left((-1)^{n+3}f\left[x_{k,1},a\right]f\left[x_{k,0},a\right]f\left[x_{k-1,n},a\right]\cdots f\left[x_{k-1,0},a\right]\right)^{2^{j-2}}$$
$$\times\left(1+\gamma_0 f\left[x_{k,0},a\right]\right)^{2^{j-1}}e_k^{2^j}\left(e_{k-1,n}\cdots e_{k-1,0}\right)^{2^{j-1}}$$
$$=-\frac{\left(f^{-1}\right)^{(n+j+2)}(0)}{(n+j+2)!}\left(-f'(a)\right)^{n+j+2}\prod_{i=2}^{j}\left[\left(-\frac{\left(f^{-1}\right)^{(n+i+1)}(0)}{(n+i+1)!}\right)\left(-f'(a)\right)^{n+i+1}\right]^{2^{j-i}}$$
$$\times\left(1+\gamma_0 f'(a)\right)^{2^{j-1}}e_k^{2^j}\left(e_{k-1,n}\cdots e_{k-1,0}\right)^{2^{j-1}}+o\left(e_k^{2^j}\left(e_{k-1,n}\cdots e_{k-1,0}\right)^{2^{j-1}}\right)$$
$$=-\frac{\left(f^{-1}\right)^{(n+j+2)}(0)}{(n+j+2)!}\left(-f'(a)\right)^{n+j+2}\prod_{i=2}^{j}\left[\left(-\frac{\left(f^{-1}\right)^{(n+i+1)}(0)}{(n+i+1)!}\right)\left(-f'(a)\right)^{n+i+1}\right]^{2^{j-i}}$$
$$\times\left(1+\gamma_0 f'(a)\right)^{2^{j-1}}C^{2^j}e_{k-1}^{2^j r}\left(C_n\cdots C_1 e_{k-1}^{p_n+\cdots+p_1+1}\right)^{2^{j-1}}+o\left(e_{k-1}^{2^j r+2^{j-1}(p_n+\cdots+p_1+1)}\right).$$

So,

$$\begin{cases} rp_1 = r, \\ rp_{j+1} = 2^j r + 2^{j-1}\left(p_n + \cdots + p_1 + 1\right),\ j = 1,\cdots,n, \\ r = p_{n+1}. \end{cases}$$

Thus, $r^2 = \left(3\cdot 2^{n-1} - 1\right)r + 2^n$, and $r = \left(3\cdot 2^{n-1} - 1 + \sqrt{9\cdot 2^{2n-2} + 2^n + 1}\right)\Big/2$. □

The parameter in the multipoint iteration (4) should be expressed by using memory as good as possible. According to the asymptotic convergence constant in (5), besides others such as

$$\gamma_0 = -f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k-1,n}\right)\right],$$

we choose the expression of the parameter here to be the following:

$$\begin{aligned} \gamma_0 = &-f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k-1,n}\right)\right] - f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k-1,n}\right), f\left(x_{k-1,n-1}\right)\right]\left(-f\left(x_{k-1,n}\right)\right) - \cdots \\ &- f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k-1,n}\right),\cdots, f\left(x_{k-1,0}\right)\right]\left(-f\left(x_{k-1,n}\right)\right)\left(-f\left(x_{k-1,n-1}\right)\right)\cdots\left(-f\left(x_{k-1,1}\right)\right). \end{aligned} \tag{6}$$

Theorem 2. *Let* $f: D \to \Re$ *be a sufficiently differentiable function with a simple root* $a \in D$, $D \subset \Re$ *be an open set,* x_0 *be close enough to a, then the family (4) with the self-acceleration (6) satisfies the error equation*:

$$\begin{aligned} e_{k+1} = &-\frac{\left(f^{-1}\right)^{(2n+2)}(0)}{(2n+2)!}\left(f'(a)\right)^{2n+2}\prod_{i=1}^{n}\left[\left(-\frac{\left(f^{-1}\right)^{(n+i+1)}(0)}{(n+i+1)!}\right)\left(-f'(a)\right)^{n+i+1}\right]^{2^{n-i}} \\ &\times\left(e_k e_{k-1,n}\cdots e_{k-1,0}\right)^{2^n} + o\left(\left(e_k e_{k-1,n}\cdots e_{k-1,0}\right)^{2^n}\right), \end{aligned} \tag{7}$$

where $e_k = x_k - a$ *and* $e_{k,j} = x_{k,j} - a, k = 0,1,\cdots$, *and achieves convergence of order at least* $\left(2^{n+1} - 1 + \sqrt{2^{2n+2} + 1}\right)\Big/2$.

Proof. By the proof of Theorem 1, for $j = 0$, we have

$$\begin{aligned} e_{k,1} &= \left(1 + \gamma_0 f\left[x_{k,0}, a\right]\right)e_k = \left(f^{-1}\left[f\left(x_{k,0}\right), f(a)\right] + \gamma_0\right) f\left[x_{k,0}, a\right] e_k \\ &= f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k-1,n}\right),\cdots, f\left(x_{k-1,0}\right), f(a)\right] \\ &\quad\times\left(-f\left[x_{k-1,n}, a\right]e_{k-1,n}\right)\cdots\left(-f\left[x_{k-1,0}, a\right]e_{k-1,0}\right) f\left[x_{k,0}, a\right] e_k \\ &= \left(-\frac{\left(f^{-1}\right)^{(n+2)}(0)}{(n+2)!}\left(-f'(a)\right)^{n+2}\right) C C_n \cdots C_1 e_{k-1}^{r+p_n+\cdots+p_1+1} + o\left(e_{k-1}^{r+p_n+\cdots+p_1+1}\right). \end{aligned}$$

For $j > 0$, we have

$$\begin{aligned} e_{k,j+1} = &-f^{-1}\left[f(a), f\left(x_{k,j}\right),\cdots, f\left(x_{k,0}\right), f\left(x_{k-1,n}\right),\cdots, f\left(x_{k-1,0}\right)\right] \\ &\times(-1)^{n+j+2} f\left[x_{k,j}, a\right]\cdots f\left[x_{k,0}, a\right] f\left[x_{k-1,n}, a\right]\cdots f\left[x_{k-1,0}, a\right] \\ &\times\cdots\times\left(-f^{-1}\left[f(a), f\left(x_{k,1}\right), f\left(x_{k,0}\right), f\left(x_{k-1,n}\right),\cdots, f\left(x_{k-1,0}\right)\right]\right)^{2^{j-2}} \\ &\times\left((-1)^{n+3} f\left[x_{k,1}, a\right] f\left[x_{k,0}, a\right] f\left[x_{k-1,n}, a\right]\cdots f\left[x_{k-1,0}, a\right]\right)^{2^{j-2}} \\ &\times\left(\left(f^{-1}\left[f\left(x_{k,0}\right), f(a)\right] + \gamma_0\right) f\left[x_{k,0}, a\right]\right)^{2^{j-1}} e_k^{2^j}\left(e_{k-1,n}\cdots e_{k-1,0}\right)^{2^{j-1}} \end{aligned}$$

$$= -\frac{\left(f^{-1}\right)^{(n+j+2)}(0)}{(n+j+2)!}\left(-f'(a)\right)^{n+j+2}\prod_{i=1}^{j}\left[\left(-\frac{\left(f^{-1}\right)^{(n+i+1)}(0)}{(n+i+1)!}\right)\left(-f'(a)\right)^{n+i+1}\right]^{2^{j-i}}$$

$$\times e_k^{2^j}\left(e_{k-1,n}\cdots e_{k-1,0}\right)^{2^j} + o\left(\left(e_k e_{k-1,n}\cdots e_{k-1,0}\right)^{2^j}\right)$$

$$= -\frac{\left(f^{-1}\right)^{(n+j+2)}(0)}{(n+j+2)!}\left(-f'(a)\right)^{n+j+2}\prod_{i=1}^{j}\left[\left(-\frac{\left(f^{-1}\right)^{(n+i+1)}(0)}{(n+i+1)!}\right)\left(-f'(a)\right)^{n+i+1}\right]^{2^{j-i}}$$

$$\times C^{2^j} e_{k-1}^{2^j r}\left(C_n\cdots C_1 e_{k-1}^{p_n+\cdots+p_1+1}\right)^{2^j} + o\left(e_{k-1}^{2^j(r+p_n+\cdots+p_1+1)}\right).$$

So,

$$\begin{cases} rp_{j+1} = 2^j\left(r+p_n+\cdots+p_1+1\right),\ j=0,\cdots,n, \\ r = p_{n+1}. \end{cases}$$

Thus, $r^2 = \left(2^{n+1}-1\right)r + 2^n$, and $r = \left(2^{n+1}-1+\sqrt{2^{2n+2}+1}\right)\Big/2$. □

3. General *n*-Point Iteration with Memory and a First Derivative

Let $x_{k,0}$ be an approximation of the simple root of $f(x)=0$. By an inverse interpolation on this one point of degree one, we have

$$\begin{aligned} f^{-1}(t) &\approx Q_1\left(t; f\left(x_{k,0}\right), f\left(x_{k,0}\right)\right) \\ &= x_{k,0} + f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k,0}\right)\right]\left(t - f\left(x_{k,0}\right)\right), \end{aligned}$$

where $f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k,0}\right)\right] = \left(f^{-1}\right)'\left(f\left(x_{k,0}\right)\right) = \dfrac{1}{f'\left(x_{k,0}\right)}$ as usual and so forth. Therefore, by using $Q_1\left(0; f\left(x_{k,0}\right), f\left(x_{k,0}\right)\right)$ to approximate the root, we have

$$x_{k+1} = x_{k,0} + f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k,0}\right)\right]\left(-f\left(x_{k,0}\right)\right),$$

which is Newton's method. However, by using the most information up to the previous step, we have the inverse interpolation of degree two:

$$\begin{aligned} &Q_2\left(t; f\left(x_{k,0}\right), f\left(x_{k,0}\right), f\left(x_{k-1,0}\right)\right) \\ &= x_{k,0} + f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k,0}\right)\right]\left(t - f\left(x_{k,0}\right)\right) \\ &\quad + f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k,0}\right), f\left(x_{k-1,0}\right)\right]\left(t - f\left(x_{k,0}\right)\right)^2. \end{aligned}$$

We suggest a one-point method with memory by $Q_2\left(0; f\left(x_{k,0}\right), f\left(x_{k,0}\right), f\left(x_{k-1,0}\right)\right)$ as follows:

$$\begin{aligned} x_{k+1} &= x_{k,0} + f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k,0}\right)\right]\left(-f\left(x_{k,0}\right)\right) \\ &\quad + f^{-1}\left[f\left(x_{k,0}\right), f\left(x_{k,0}\right), f\left(x_{k-1,0}\right)\right]\left(-f\left(x_{k,0}\right)\right)^2. \end{aligned} \tag{8}$$

Furthermore, we construct a family of *n*-point iterations with the memory on the whole previous step by using the inverse interpolation as follows:

$$\begin{cases} x_{k,0} = x_k, \\ x_{k,1} = x_{k,0} + f^{-1}\left[f(x_{k,0}), f(x_{k,0})\right](-f(x_{k,0})) + f^{-1}\left[f(x_{k,0}), f(x_{k,0}), f(x_{k-1,n-1})\right](-f(x_{k,0}))^2 + \cdots \\ \quad + f^{-1}\left[f(x_{k,0}), f(x_{k,0}), f(x_{k-1,n-1}), \cdots, f(x_{k-1,0})\right](-f(x_{k,0}))^2(-f(x_{k-1,n-1}))\cdots(-f(x_{k-1,1})), \\ x_{k,j+1} = x_{k,j} + f^{-1}\left[f(x_{k,j}), f(x_{k,j-1})\right](-f(x_{k,j})) + \cdots \\ \quad + f^{-1}\left[f(x_{k,j}), \cdots, f(x_{k,1}), f(x_{k,0}), f(x_{k,0})\right](-f(x_{k,j}))\cdots(-f(x_{k,1}))(-f(x_{k,0})) \\ \quad + f^{-1}\left[f(x_{k,j}), \cdots, f(x_{k,0}), f(x_{k,0}), f(x_{k-1,n-1})\right](-f(x_{k,j}))\cdots(-f(x_{k,1}))(-f(x_{k,0}))^2 \\ \quad + \cdots + f^{-1}\left[f(x_{k,j}), \cdots, f(x_{k,0}), f(x_{k,0}), f(x_{k-1,n-1}), \cdots, f(x_{k-1,0})\right] \\ \quad \times(-f(x_{k,j}))\cdots(-f(x_{k,1}))(-f(x_{k,0}))^2(-f(x_{k-1,n-1}))\cdots(-f(x_{k-1,1})), \; j = 1, \cdots, n-1, \\ x_{k+1} = x_{k,n}, \quad k = 0, 1, \cdots. \end{cases} \tag{9}$$

Theorem 3. *Let* $f : D \to \Re$ *be a sufficiently differentiable function with a simple root* $a \in D$, $D \subset \Re$ *be an open set,* x_0 *be close enough to a, then the family of n-point iterations* (*9*) *satisfies the error equation*:

$$e_{k+1} = -\frac{(f^{-1})^{(2n+3)}(0)}{(2n+3)!}(-f'(a))^{2n+3}\prod_{i=1}^{n}\left[\left(-\frac{(f^{-1})^{(n+i+2)}(0)}{(n+i+2)!}\right)(-f'(a))^{n+i+2}\right]^{2^{n-i}} \times\left(e_k^2 e_{k-1,n}\cdots e_{k-1,0}\right)^{2^n} + o\left(\left(e_k^2 e_{k-1,n}\cdots e_{k-1,0}\right)^{2^n}\right), \tag{10}$$

where $e_k = x_k - a$ *and* $e_{k,j} = x_{k,j} - a, k = 0, 1, \cdots$, *and achieves convergence of order at least* $\left(3\cdot 2^{n-1} - 1 + \sqrt{9\cdot 2^{2n-2} - 2^n + 1}\right)/2$.

Proof. Denote $e_k = x_k - a$ and $e_{k,j} = x_{k,j} - a, j = 0, \cdots, n$. Supposed that $e_{k,j} = C_j e_k^{p_j} + o\left(e_k^{p_j}\right)$ and $e_{k+1} = Ce_k^r + o\left(e_k^r\right)$. Then,

$$e_{k,j} = C_j C^{p_j} e_{k-1}^{rp_j} + o\left(e_{k-1}^{rp_j}\right) \text{ and } e_{k+1} = C^{r+1} e_{k-1}^{r^2} + o\left(e_{k-1}^{r^2}\right).$$

When $n = 1$, we have

$$\begin{aligned} e_{k,1} &= x_{k,0} - a + f^{-1}\left[f(x_{k,0}), f(x_{k,0})\right](-f(x_{k,0})) \\ &\quad + f^{-1}\left[f(x_{k,0}), f(x_{k,0}), f(x_{k-1,0})\right](-f(x_{k,0}))^2 \\ &= -f^{-1}\left[f(x_{k,0}), f(x_{k,0}), f(x_{k-1,0}), f(a)\right]\left(-f[x_{k,0}, a]e_{k,0}\right)^2\left(-f[x_{k-1,0}, a]e_{k-1,0}\right) \\ &= -\frac{(f^{-1})^{(3)}(0)}{3!}(-f'(a))^3 e_k^2 e_{k-1,0} + o\left(e_k^2 e_{k-1,0}\right) \\ &= -\frac{(f^{-1})^{(3)}(0)}{3!}(-f'(a))^3 C^2 e_{k-1}^{2r+1} + o\left(e_{k-1}^{2r+1}\right). \end{aligned}$$

So, we have

$$\begin{cases} rp_1 = 2r + 1, \\ r = p_1. \end{cases}$$

Thus, $r^2 = 2r + 1$ and $r = 1 + \sqrt{2}$.

When $n>1$, for $j=0,\cdots,n-1$, we have

$$e_{k,j+1}=-f^{-1}\left[f(a),f(x_{k,j}),\cdots,f(x_{k,1}),f(x_{k,0})f(x_{k,0}),f(x_{k-1,n-1}),\cdots,f(x_{k-1,0})\right]$$
$$\times(-1)^{n+j+2}f[x_{k,j},a]\cdots f[x_{k,1},a]f[x_{k,0},a]^2 f[x_{k-1,n-1},a]\cdots f[x_{k-1,0},a]$$
$$\times e_{k,j}\cdots e_{k,1}e_{k,0}^2 e_{k-1,n-1}\cdots e_{k-1,0}$$
$$=-f^{-1}\left[f(a),f(x_{k,j}),\cdots,f(x_{k,1}),f(x_{k,0})f(x_{k,0}),f(x_{k-1,n-1}),\cdots,f(x_{k-1,0})\right]$$
$$\times(-1)^{n+j+2}f[x_{k,j},a]\cdots f[x_{k,1},a]f[x_{k,0},a]^2 f[x_{k-1,n-1},a]\cdots f[x_{k-1,0},a]$$
$$\times\left(-f^{-1}\left[f(a),f(x_{k,j-1}),\cdots,f(x_{k,1}),f(x_{k,0})f(x_{k,0}),f(x_{k-1,n-1}),\cdots,f(x_{k-1,0})\right]\right)$$
$$\times(-1)^{n+j+1}f[x_{k,j-1},a]\cdots f[x_{k,1},a]f[x_{k,0},a]^2 f[x_{k-1,n-1},a]\cdots f[x_{k-1,0},a]$$
$$\times\left(e_{k,j-1}\cdots e_{k,1}e_{k,0}^2 e_{k-1,n-1}\cdots e_{k-1,0}\right)^2$$
$$=-f^{-1}\left[f(a),f(x_{k,j}),\cdots,f(x_{k,1}),f(x_{k,0})f(x_{k,0}),f(x_{k-1,n-1}),\cdots,f(x_{k-1,0})\right]$$
$$\times(-1)^{n+j+2}f[x_{k,j},a]\cdots f[x_{k,1},a]f[x_{k,0},a]^2 f[x_{k-1,n-1},a]\cdots f[x_{k-1,0},a]$$
$$\times\cdots\times\left(-f^{-1}\left[f(a),f(x_{k,0}),f(x_{k,0}),f(x_{k-1,n-1}),\cdots,f(x_{k-1,0})\right]\right)^{2^{j-1}}$$
$$\times\left((-1)^{n+2}f[x_{k,0},a]^2 f[x_{k-1,n-1},a]\cdots f[x_{k-1,0},a]\right)^{2^{j-1}}\left(e_k^2 e_{k-1,n-1}\cdots e_{k-1,0}\right)^{2^j}$$
$$=-\frac{(f^{-1})^{(n+j+2)}(0)}{(n+j+2)!}(-f'(a))^{n+j+2}\prod_{i=1}^{j}\left[\left(-\frac{(f^{-1})^{(n+i+1)}(0)}{(n+i+1)!}\right)(-f'(a))^{n+i+1}\right]^{2^{j-i}}$$
$$\times\left(e_k^2 e_{k-1,n-1}\cdots e_{k-1,0}\right)^{2^j}+o\left(\left(e_k^2 e_{k-1,n-1}\cdots e_{k-1,0}\right)^{2^j}\right)$$
$$=-\frac{(f^{-1})^{(n+j+2)}(0)}{(n+j+2)!}(-f'(a))^{n+j+2}\prod_{i=1}^{j}\left[\left(-\frac{(f^{-1})^{(n+i+1)}(0)}{(n+i+1)!}\right)(-f'(a))^{n+i+1}\right]^{2^{j-i}}$$
$$\times\left(C^2C_{n-1}\cdots C_1 e_{k-1}^{2r+p_{n-1}+\cdots+p_1+1}\right)^{2^j}+o\left(e_{k-1}^{2^j(2r+p_{n-1}+\cdots+p_1+1)}\right).$$

So,

$$\begin{cases} rp_{j+1}=2^j\left(2r+p_{n-1}+\cdots+p_1+1\right),\ j=0,\cdots,n-1,\\ r=p_n. \end{cases}$$

Thus, $r^2=\left(3\cdot 2^{n-1}-1\right)r+2^{n-1}$, and $r=\left(3\cdot 2^{n-1}-1+\sqrt{9\cdot 2^{2n-2}-2^n+1}\right)/2$. □

4. Numerical Examples

The proposed families (4), (4) with (6), (9), as well as the existing family (2) with and without memory are demonstrated to solve the nonlinear equations in the examples. For general families of biparametric multipoint iterations with and without memory as well as other related discussions, please refer to, e.g., [10] [11]. The computational order of convergence is defined by:

$$\text{COC}=\frac{\log\left(|x_n-a|/|x_{n-1}-a|\right)}{\log\left(|x_{n-1}-a|/|x_{n-2}-a|\right)}.$$

Example 1. The numerical results on $f(x) = e^{x-2} - 1$ in **Table 2** agree with the convergence rates in Theorems 1 - 3.

Example 2. The numerical results in **Table 3** are for nonlinear functions:

$$f_1(x) = x^2 - e^{-x} - 3x + 1, a = 0, x_0 = 0.2,$$

$$f_2(x) = e^{x^2} + \sin x - 1, a = 0, x_0 = 0.25,$$

$$f_3(x) = e^{-x} - \arctan x - 1, a = 0, x_0 = 0.2.$$

Example 3. The basins of attraction of the existing family (2) with (3) by using direct Newton interpolation, the family (4) with (6) and family (9) by using inverse interpolation to solve $f(z) = z^3 - 1 = 0$ with the criterion $\min\{|e_n|, |f(z_n)|\} < 10^{-6}$ in $\mathbb{C}$ are shown in **Figure 1**. The colors "r", "g", "b", "c", "w", "y", "m", "k" are assigned for the number of iteration {0, 1, 2}, 3, 4, {5, 6}, {7, 8}, {9, 10, 11}, {12, 13, 14, 15} and default. The basin of attraction of the method based on inverse interpolation may be a little smaller, but not too much worse than that based on direct interpolation, since the first substep is a Steffensen's method for (4) with (6) and a Newton's method for (9) respectively in the first step when $k = 0$.

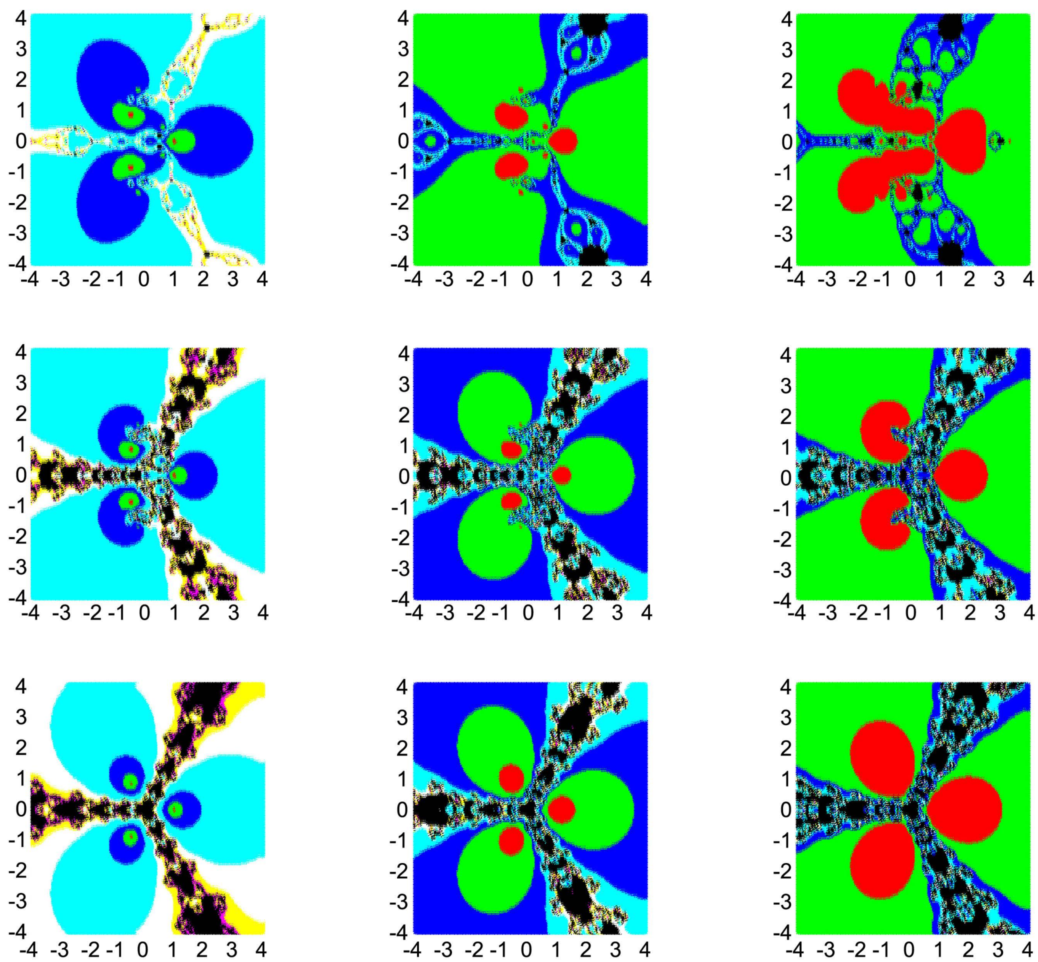

Figure 1. (2) with (3), (4) with (6), (9) in 1st, 2nd, 3rd rows (left n = 1, middle n = 2 and right n = 3).

Table 2. The numerical results of $f(x)=e^{x-2}-1, a=2, x_0=1.8$.

Method	n	1	2	3	4	5
2 evaluations:						
(2), $\beta_j=1$	$\|e_n\|$	0.17729	0.73857e−1	0.14661e−1	0.63024e−3	0.11904e−5
	COC	0.074895	7.26446	1.84655	1.9462	1.99303
(2) and (3)	$\|e_n\|$	0.42075e−1	0.16877e−7	0.25502e−27	0.13172e−99	0.41287e−356
	COC	0.96857	9.44863	3.09857	3.64704	3.54841
(4), $\gamma_0=1$	$\|e_n\|$	0.42075e−1	0.77998e−4	0.10477e−10	0.6678e−30	0.48955e−82
	COC	0.96857	4.0354	2.5154	2.7934	2.7160
(4) and (6)	$\|e_n\|$	0.42075e−1	0.11515e−5	0.28557e−21	3.4382e−77	0.36827e−276
	COC	0.96857	6.7397	3.4202	3.5833	3.5582
(9)	$\|e_n\|$	0.21403e−1	0.31733e−4	0.71458e−11	5.4013e−27	0.6949e−66
	COC	1.3886	2.9148	2.3498	2.4252	2.4123
3 evaluations:						
(2), $\beta_j=1$	$\|e_n\|$	0.35838e−1	0.13272e−4	0.29477e−18	0.71723e−73	0.2514e−291
	COC	1.06827	4.59549	3.97898	4.0	4.0
(2) and (3)	$\|e_n\|$	0.26631e−3	0.45628e−30	0.88977e−232	0.52002e−1750	0.17484e−13184
	COC	4.11412	9.30787	7.53601	7.52681	7.53143
(4), $\gamma_0=1$	$\|e_n\|$	0.12228e−2	0.44394e−17	.24301e−98	0.2107e−562	0.92699e−3208
	COC	3.1671	6.5231	5.6275	5.7107	5.7004
(4) and (6)	$\|e_n\|$	0.12228e−2	0.98529e−24	0.10489e−181	0.74052e−1372	0.83137e−10335
	COC	3.1671	9.5288	7.4891	7.5339	7.5309
(9)	$\|e_n\|$	0.3137e−3	0.99202e−20	0.58718e−108	0.42933e−582	0.31434e−3129
	COC	4.0124	5.8834	5.3471	5.3740	5.3722
4 evaluations:						
(2), $\beta_j=1$	$\|e_n\|$	0.17453e−3	0.81942e−29	0.19310e−231	0.18364e−1852	0.12287e−14820
	COC	4.37665	8.27955	8.00003	8.0	8.0
(2)and(3)	$\|e_n\|$	0.17453e−3	0.65829e−66	0.14269e−1037	0.37634e−16110	0.38186e−249972
	COC	4.37665	20.4055	15.5657	15.5121	15.5157
(4), $\gamma_0=1$	$\|e_n\|$	0.11515e−5	0.12872e−70	0.12264e−827	0.18343e−9674	0.10442e−113045
	COC	7.4964	12.396	11.655	11.686	11.685
(4) and (6)	$\|e_n\|$	0.11515e−5	0.62305e−96	0.22838e−1495	0.57552e−23209	0.19685e−360108
	COC	7.4964	17.227	15.503	15.516	15.516
(9)	$\|e_n\|$	0.7520e−7	0.40585e−83	0.33630e−948	0.36766e−10769	0.46220e−122260
	COC	9.1918	11.871	11.343	11.353	11.352

Table 3. Numerical results for $f_i(x), i = 1,2,3$.

Method	$f_1(x)$		$f_2(x)$		$f_3(x)$	
	$\lvert x_4 - a \rvert$	COC	$\lvert x_4 - a \rvert$	COC	$\lvert x_4 - a \rvert$	COC
2 evaluations:						
(2), $\beta_j = 1$	0.60393e−14	2.00008	0.78672e−3	1.90986	0.59130e−14	2.00008
(2) and (3)	0.10081e−120	3.56363	0.26298e−63	3.59844	0.11450e−124	3.60778
(4), $\gamma_0 = 1$	0.35144e−48	2.7631	0.69234e−16	2.8113	0.25817e−47	2.7602
(4) and (6)	0.3559e−110	3.5726	0.72971e−46	3.5971	0.37112e−109	3.570
(9)	0.2969e−31	2.4190	0.87258e−18	2.4258	0.3511e−31	2.4182
3 evaluations:						
(2), $\beta_j = 1$	0.43659e−224	4.0	0.20878e−80	4.0	0.59394e−224	4.0
(2) and (3)	0.47499e−2253	7.53197	0.30298e−1047	7.52119	0.46843e−2079	7.53541
(4), $\gamma_0 = 1$	0.16921e−823	5.7068	0.36651e−306	5.7064	0.54017e−841	5.7103
(4) and (6)	0.70252e−1930	7.5323	0.14318e−780	7.5306	0.23668e−1942	7.5355
(9)	0.30506e−647	5.3732	0.23495e−360	5.3721	0.12358e−647	5.3746
4 evaluations:						
(2), $\beta_j = 1$	0.17157e−3480	8.0	0.25750e−1636	8.0	0.23985e−3496	8.0
(2) and (3)	0.92232e−27930	15.5143	0.16190e−14127	15.5151	0.55520e−27255	15.5192
(4), $\gamma_0 = 1$	0.22274e−13824	11.685	0.23709e−5329	11.684	0.10333e−13968	11.687
(4) and (6)	0.10116e−32543	15.515	0.11449e−13022	15.514	0.28412e−32929	15.517
(9)	0.49612e−11831	11.352	0.5289e−6530	11.352	0.17565e−11940	11.353
5 evaluations:						
(2), $\beta_j = 1$	0.18825e−55261	16.0	0.88727e−26366	16.0	0.82392e−55118	16.0
(2) and (3)	0.11450e−443164	31.5076	0.38286e−215574	31.5063	0.16232e−437023	31.5092
(4), $\gamma_0 = 1$	0.28256e−226738	23.676	0.42849e−88916	23.676	0.45549e−229614	23.676
(4) and (6)	0.24173e−535738	31.508	0.20924e−213680	31.507	0.67598e−543101	31.508
(9)	0.50946e−202406	23.343	0.10801e−111444	23.343	0.42064e−205790	23.343

5. Conclusion

In this paper, we construct a family (4) of $n+1$-point iterations derivative-free and another family (9) of n-point iterations using a first derivative by the inverse interpolatory polynomial with memory to solve the simple root of a nonlinear equation. The general families (4) and (9) use $n+1$ functional evaluations with the memory in the previous step to achieve the super convergence of order $\left(3\cdot 2^{n-1}-1+\sqrt{9\cdot 2^{2n-2}+2^n+1}\right)/2$ and $\left(3\cdot 2^{n-1}-1+\sqrt{9\cdot 2^{2n-2}-2^n+1}\right)/2$ respectively. The general family (4) with (6) achieves the super convergence of order $\left(2^{n+1}-1+\sqrt{2^{2n+2}+1}\right)/2$, which is the same as that of the existing family (2) with (3). When $n=4$, as special case, both of them achieve the super convergence of order $\left(31+\sqrt{1025}\right)/2=31.5078$ and have the efficiency index $\left\{\left(31+\sqrt{1025}\right)/2\right\}^{1/5}=1.9938$. The application of the memory is more handy in the proposed families than that of (3) in (2). The basins of attraction of the related multipoint iterations with memory are also demonstrated. The advantage of effectiveness and convenience in practice of the proposed families is confirmed by numerical examples.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Ortega, J.M. and Rheinboldt, W.G. (1970) Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York.

[2] Traub, J.F. (1964) Iterative Methods for the Solution of Equations. Prentice-Hall, Englewood Cliffs, New Jersey.

[3] Kung, H.T. and Traub, J.F. (1974) Optimal Order of One-Point and Multipoint Iteration. *Journal of the ACM*, **21**, 634-651. https://doi.org/10.1145/321850.321860

[4] Neta, B. (1981) On a Family of Multipoint Methods for Non-Linear Equations. *International Journal of Computer Mathematics*, **9**, 353-361. https://doi.org/10.1080/00207168108803257

[5] Petković, M.S., Neta, B., Petković, L.D. and Džunić, J. (2013) Chapter 2—Two-Point Methods. *Multipoint Methods for Solving Nonlinear Equations*, Academic Press, Cambridge, 27-83. https://doi.org/10.1016/B978-0-12-397013-8.00002-9

[6] Neta, B. and Petković, M.S. (2010) Construction of Optimal Order Nonlinear Solvers Using Inverse Interpolation. *Applied Mathematics and Computation*, **217**, 2448-2455. https://doi.org/10.1016/j.amc.2010.07.045

[7] Džunić, J. and Petković, M.S. (2012) On Generalized Multipoint Root-Solvers with Memory. *Journal of Computational and Applied Mathematics*, **236**, 2909-2920. https://doi.org/10.1016/j.cam.2012.01.035

[8] Petković, M.S. (2010) On a General Class of Multipoint Root-Finding Methods of High Computational Efficiency. *SIAM Journal on Numerical Analysis*, **47**, 4402-4414. https://doi.org/10.1137/090758763

[9] Zheng, Q., Li, J. and Huang, F. (2011) An Optimal Steffensen-Type Family for Solving Nonlinear Equations. *Applied Mathematics and Computation*, **217**, 9592-9597. https://doi.org/10.1016/j.amc.2011.04.035

[10] Džunić, J. and Petković, M.S. (2014) On Generalized Biparametric Multipoint Root Finding Methods with Memory. *Journal of Computational and Applied Mathematics*, **255**, 362-375. https://doi.org/10.1016/j.cam.2013.05.013

[11] Zheng, Q., Zhao, X. and Liu, Y.F. (2015) An Optimal Biparametric Multipoint Family and Its Self-Acceleration with Memory for Solving. *Algorithms*, **8**, 1111-1120. https://doi.org/10.3390/a8041111

[12] Wang, X.F. and Zhang, T. (2015) Efficient N-Point Iterative Methods with Memory for Solving Nonlinear Equations. *Numerical Algorithms*, **70**, 357-375. https://doi.org/10.1007/s11075-014-9951-8

[13] Zheng, Q., Huang, F.X., Guo, X.H. and Feng, X.L. (2013) Doubly-Accelerated Steffensens Methods with Memory and Their Applications on Solving Nonlinear ODEs. *Journal of Computational Analysis and Applications*, **15**, 886-891.

[14] Zheng, Q., Wang, J., Zhao, P. and Zhang, L. (2009) A Steffensen-Like Method and Its Higher-Order Variants. *Applied Mathematics and Computation*, **214**, 10-16. https://doi.org/10.1016/j.amc.2009.03.053

[15] Chicharro, F., Cordero, A., Gutirrez, J.M. and Torregrosa, J.R. (2013) Complex Dynamics of Derivative-Free Methods for Nonlinear Equations. *Applied Mathematics and Computation*, **219**, 7023-7035. https://doi.org/10.1016/j.amc.2012.12.075

[16] Chicharro, F., Cordero, A. and Torregrosa, J. (2015) Dynamics and Fractal Dimension of Steffensen-Type Methods. *Algorithms*, **8**, 271-279. https://doi.org/10.3390/a8020271

[17] Amat, S., Busquier, S. and Plaza, S. (2004) Dynamics of a Family of Third-Order Iterative Methods That Do Not Require Using Second Derivatives. *Applied Mathematics and Computation*, **154**, 735-746. https://doi.org/10.1016/S0096-3003(03)00747-1

[18] Amat, S., Busquier, S., Bermúdez, C. and Plaza, S. (2012) On Zilies of High Order Newton Type Methods. *Applied Mathematics Letters*, **25**, 2209-2217. https://doi.org/10.1016/j.aml.2012.06.004

[19] Scott, M., Neta, B. and Chun, C. (2011) Basin Attractors for Various Methods. *Applied Mathematics and Computation*, **218**, 2584-2599. https://doi.org/10.1016/j.amc.2011.07.076

[20] Neta, B., Scott, M. and Chun, C. (2012) Basins of Attraction for Several Methods to Find Simple Roots of Nonlinear Equations. *Applied Mathematics and Computation*, **218**, 10548-10556. https://doi.org/10.1016/j.amc.2012.04.017