A GENERAL CLASS OF DERIVATIVE FREE WITH MEMORY ROOT SOLVERS

Moin-ud-Din Junjua[1], Fiza Zafar[1], Nusrat Yasmin[2], Saima Akram[3]

In this paper, we present a general with-memory extension of an existing without memory derivative free family of $n-$point optimal methods to solve nonlinear functions employing a self-accelerating parameter. At each iterative step, we use a suitable variation of the free parameter. This parameter is computed by using the information from current and previous iterations so that the convergence order of the existing family is improved from 2^n to $2^n + 2^{n-1}$ without using any additional function evaluations. An extensive comparison of our with memory method is done with the existing with- and without memory methods using smooth and non-smooth nonlinear functions. The performance of the methods is also analyzed visually using complex plane. Which confirms that the proposed family of with-memory methods is competitive with the previous methods of the same domain.

Keywords: Derivative Free Iterative Method, $R-$order of Convergence, With-Memory Method, Nonlinear Equations.

MSC2010: 65H04, 65H05, 65B99.

1. Introduction

Development of root finding methods is an important task in numerical analysis and applied sciences, which has been focused much attention recently. Traub for the first time classified the root solvers as one point and multipoint methods [11]. He also discussed the limitations of convergence order and computational efficiency of one point methods that for such methods the informational efficiency cannot exceed the upper bound 2. So, the multipoint methods are of great importance because of their high efficiency and accuracy [11]. These methods have notable applications in experimental mathematics, number theory, and research fields including high energy physics, nonlinear process simulation, finite element modeling CAD, 3D real time graphics, statistics, security, cryptography and so on. Newton's method and Steffensen's method are most famous multiprecision methods [5]. Several one point and multipoint iterative methods for finding roots of nonlinear equations have been investigated in the recent past [3, 4, 6, 7, 9, 10, 14]. Steffensen type methods have an advantage that they work for smooth as well as for non-smooth functions because derivative of a function in many practical situations is not easily available. According to the conjecture of Kung and Traub [2], a root finding method without memory with $n + 1$ functional evaluations can have at most convergence order 2^n (optimal order) and efficiency index (EI) $2^{\frac{n}{n+1}}$, where $n \in \mathbb{N}$. The methods which satisfy the conjecture of Kung and Traub are known as optimal methods. An interesting fact about the optimal without memory root solvers is that their convergence order and efficiency can be increased without increasing the number of functional evaluations, that is by extending them to with-memory methods. A root solver which uses current and previous information at each iterative step is known as with-memory root solver. The construction of with-memory optimal root solvers is based on two techniques; first is by the use of self accelerating parameter [10] and the second is by the use of inverse interpolation [8].

[1,2,3]Centre for Advanced Studies in Pure and Applied Mathematics, Bahauddin Zakariya University Multan, Pakistan, E-mail: moin_junjua@yahoo.com (Corresponding Author)

Motivated by this idea, we, in this paper, construct a general derivative free class of with-memory root solvers based on an optimal $n-$point derivative free class of without-memory methods given in [13], using a self-accelerating parameter. This self-accelerating parameter is computed by using the Newton's interpolation polynomial and it is proved that the convergence order of $n-$point family of without memory methods is increased from 2^n to 2^n+2^{n-1}. The contents of this paper are summarized as follows. In Section 2, three methods to calculate the accelerating parameter are presented and a new $n-$point class of derivative free with-memory root solvers is developed. Section 3, includes the analysis of $R-$order convergence [5]. The comparison of the proposed class of methods with the existing methods using some nonlinear functions is given in Section 4. In Section 5, dynamical behavior of the iterative methods is presented and finally the conclusions of this work are provided in Section 6.

2. New derivative free class of with-memory root solvers

In a recent paper [13], Zafar et al. proposed a derivative free family of without memory $n-$point optimal methods of arbitrary order of convergence 2^n $(n \geq 1)$. This family requires $n+1$ functional evaluations and is constructed by using rational interpolation given as follows:

$$\begin{aligned}
w_{k,0} &= x_{k,0} + \gamma f(x_{k,0}), \gamma \in \mathbb{R}, \\
w_{k,1} &= x_{k,0} - \frac{f(x_{k,0})}{f[x_{k,0}, w_{k,0}]}, \\
&\vdots \\
w_{k,n} &= x_{k,0} - \frac{a_0}{a_1}, \; n \geq 2, \qquad (1)
\end{aligned}$$

where, $k \geq 0$ is iteration index, a_0 and a_1 are constants to be determined through a rational polynomial of degree $n-1$ such that:

$$r_{n-1}(t) = \frac{a_0 + a_1(t-x)}{1 + b_1(t-x) + \cdots + b_{n-2}(t-x)^{n-2}}, \; n \geq 2, b_0 = 1, \qquad (2)$$

with the following conditions:

$$\begin{aligned}
r_{n-1}(x_{k,0}) &= f(x_{k,0}), \\
r_{n-1}(w_{k,n}) &= f(w_{k,n}), \; n = 1, 2, \cdots, n-1, \; n \geq 2, \qquad (3)
\end{aligned}$$

where, $b_1, \; b_2, \cdots b_{n-2}$ are constants to be determined through (3).

For $n = 3$ in the above family (1), we obtain the following three-point method [13]:

$$\begin{aligned}
w_{k,0} &= x_{k,0} + \gamma f(x_{k,0}), \gamma \in \mathbb{R}, k \geq 0, \\
w_{k,1} &= x_{k,0} - \frac{f(x_{k,0})}{f[w_{k,0}, x_{k,0}]}, \\
w_{k,2} &= x_{k,0} - \frac{f(x_{k,0})(f(w_{k,1}) - f(w_{k,0}))}{f(w_{k,1})f[w_{k,0}, x_{k,0}] - f(w_{k,0})f[w_{k,1}, x_{k,0}]}, \\
w_{k,3} &= x_{k,0} - \frac{f(x_{k,0})(h_1 + h_2 + h_3)}{h_1 f[w_{k,0}, x_{k,0}] + h_2 f[w_{k,1}, x_{k,0}] + h_3 f[w_{k,2}, x_{k,0}]}, \qquad (4)
\end{aligned}$$

where,

$$\begin{aligned}
h_1 &= f(w_{k,1})f(w_{k,2})(w_{k,2} - w_{k,1}), \qquad (5) \\
h_2 &= f(w_{k,0})f(w_{k,2})(w_{k,0} - w_{k,2}), \qquad (6) \\
h_3 &= f(w_{k,0})f(w_{k,1})(w_{k,1} - w_{k,0}). \qquad (7)
\end{aligned}$$

A four-point and five-point method is also given in the same paper [13]. Throughout this paper, we often use normalized Taylor series coefficients for f as follows:

$$c_k = \frac{f^{(j)}(\alpha)}{k!f'(\alpha)}, j = 2, 3, ...,$$

where, α is a simple root of the real function $f : G \subset \mathbb{R} \to \mathbb{R}$. For the error analysis of (1), we require the following theorem, which was proved in [13].

Theorem 2.1. *Let a function $f : G \subset \mathbb{R} \to \mathbb{R}$ is sufficiently differentiable and let α be a simple zero in an open interval $I_f \subset G$. Let x_0 be a simple root close enough to α then the $n-$point family (1) has convergence order 2^n and the following is the error relation.*

$$\varepsilon_{k+1} = x_{k+1} - \alpha \sim d_n \varepsilon^{2^n}, \; k = 0, 1, \cdots \tag{8}$$

where,

$$d_{-1} = 1 + \gamma f'(\alpha), \; d_0 = 1, \; d_1 = (1 + \gamma f'(\alpha))c_2, \tag{9}$$

$$d_m = d_{m-1}[c_2 d_{m-1} + (-1)^{m-1} c_{m+1} d_{m-2} + \cdots + d_{-1}], \; m = 2, \cdots, n. \tag{10}$$

By using the theorem stated above, the general error for the family (1) is described by:

$$\varepsilon_{k,j} \sim \xi_{k,j}(1 + \gamma f'(\alpha))^{2^{j-1}} \varepsilon_k^{2^j}, \; j = 1, \cdots, n, \tag{11}$$

where, $\varepsilon_k = w_{k,0} - \alpha$, $\varepsilon_{k,j} = w_{k,j} - \alpha$, $j = 1, \cdots, n$, and k is iteration index. The above family (1) is constructed by varying the parameter γ. In this paper, γ is determined by using the information of current and previous iteration, such that the family (1) becomes a with-memory family. To increase the order of convergence of the family (1), we choose $\gamma = \gamma_k \approx -\frac{1}{f'(\alpha)}$, which would exceeds the order of the family (1) from 2^n to $2^n + 2^{n-1}$. However, $f'(\alpha)$ is not easily available at all data points, so we use a suitable approximation of $f'(\alpha)$, calculated by using available information. Our main task is to increase the order of convergence of the new family (1) without increasing the number of function evaluations. To approximate $f'(\alpha)$, three methods are presented here. Method I is the Newton's interpolating polynomial of degree two passing through the points $x_{k,0}, w_{k-1,n-1}$ and $w_{k-1,n-2}$ given as:

$$\begin{aligned} N_2(t) &= N_2(t;\; x_{k,0}, w_{k-1,n-1}, w_{k-1,n-2}) \\ &= f(x_{k,0}) + f[x_{k,0}, w_{k-1,n-1}](t - x_{k,0}) \\ &\quad + f[x_{k,0}, w_{k-1,n-1}, w_{k-1,n-2}](t - x_{k,0})(t - w_{k-1,n-1}). \end{aligned} \tag{12}$$

Method II uses the Newton's interpolant of degree three passing through nodes $x_{k,0}, w_{k-1,n-1}, w_{k-1,n-2}$ and $w_{k-1,n-3}$ such that:

$$\begin{aligned} N_3(t) &= N_3(t; x_{k,0}, w_{k-1,n-1}, w_{k-1,n-2}, w_{k-1,n-3}) \\ &= f(x_{k,0}) + f[x_{k,0}, w_{k-1,n-1}](t - x_{k,0}) \\ &\quad + f[x_{k,0}, w_{k-1,n-1}, w_{k-1,n-2}](t - x_{k,0})(t - w_{k-1,n-1}) \\ &\quad + f[x_{k,0}, w_{k-1,n-1}, w_{k-1,n-2}, w_{k-1,n-3}](t - x_{k,0})(t - w_{k-1,n-1}) \\ &\quad (t - w_{k-1,n-2}), \end{aligned} \tag{13}$$

and Method III uses the Newton's interpolant of degree four passing through $x_{k,0}, w_{k-1,n-1}$, $w_{k-1,n-2}, w_{k-1,n-3}$ and $w_{k-1,n-4}$ such that:

$$\begin{aligned} N_4(t) &= N_4(t; x_{k,0}, w_{k-1,n-1}, w_{k-1,n-2}, w_{k-1,n-3}, w_{k-1,n-4}) \\ &= f(x_{k,0}) + f[x_{k,0}, w_{k-1,n-1}](t - x_{k,0}) \\ &\quad + f[x_{k,0}, w_{k-1,n-1}, w_{k-1,n-2}](t - x_{k,0})(t - w_{k-1,n-1}) \\ &\quad + f[x_{k,0}, w_{k-1,n-1}, w_{k-1,n-2}, w_{k-1,n-3}](t - x_{k,0})(t - w_{k-1,n-1}) \\ &\quad (t - w_{k-1,n-2}) + f[x_{k,0}, w_{k-1,n-1}, w_{k-1,n-2}, w_{k-1,n-3}, w_{k-1,n-4}] \\ &\quad (t - x_{k,0})(t - w_{k-1,n-1})(t - w_{k-1,n-2})(t - w_{k-1,n-3}). \end{aligned} \tag{14}$$

To calculate the varying parameter γ_k, we use (12), (13) and (14) and obtain the following formulae:

$$\gamma_k = -\frac{1}{N_2'(x_{k,0})} \approx -\frac{1}{f'(\alpha)} \text{(Method I)}, \tag{15}$$

$$\gamma_k = -\frac{1}{N_3'(x_{k,0})} \approx -\frac{1}{f'(\alpha)} \text{ (Method II)}, \tag{16}$$

$$\gamma_k = -\frac{1}{N_4'(x_{k,0})} \approx -\frac{1}{f'(\alpha)} \text{ (Method III)}, \tag{17}$$

The derivative of Newton's interpolating polynomial is calculated by the following Lemma given by Petković in [8] as:

Lemma 2.1. *Let N_m be Newton's interpolating polynomial of degree m that interpolates a function f at $m+1$ distinct nodes $t_0, t_1, ..., t_m$ contained in an interval I_f and the derivative is continuous in I_f. Assume that all differences $e_j = tj - \alpha$ are sufficiently close to zero α of f and the condition $e_0 = O(e_1.e_2., ..., e_m)$ holds. Then, $N_m'(t_0) \sim f'(\alpha)(1 + (-1)^{m+1} c_{m+1} \prod_{j=1}^{m} e_j)$.*

Thus, for $m = 2$ we get

$$\begin{aligned} N_2'(x_{k,0}) &\sim f'(\alpha)(1 + (-1)^3 c_3 \prod_{j=1}^{2} e_j) \\ &= f'(\alpha)(1 - c_3 e_1 e_2), \end{aligned} \tag{18}$$

where e_1 and e_2 are errors appeared in the first and second step of the $n-$point family (1). Similarly for $m = 3$ we have,

$$N_3'(w_{k,0}) = f'(\alpha)(1 + c_4 e_1 e_2 e_3), \tag{19}$$

where e_1, e_2 and e_3 are errors appeared in the first, second and third step of the $n-$point family (1).

By replacing the free parameter γ in the iterative family (1) with γ_k, we get the following $n-$point derivative free family with-memory:

$$\begin{aligned} w_{k,0} &= x_{k,0} + \gamma_k f(x_{k,0}), k \geq 0 \\ w_{k,1} &= x_{k,0} - \frac{f(x_{k,0})}{f[x_{k,0}, w_{k,0}]}, \\ &\vdots \\ w_{k,n} &= x_{k,0} - \frac{a_0}{a_1}, \; n \geq 2, \end{aligned} \tag{20}$$

where $a_0, b_1, \cdots b_{n-2}$ and a_1 are constants to be determined through interpolating conditions (3) and γ_k is computed by one of the formulae (15),(16) and (17).

3. $R-$order of convergence of with-memory family

To calculate the order of convergence of the new derivative free with-memory family (20), we use the idea of $R-$order of convergence [5] and give the following theorems.

Theorem 3.1. *Let $f(x)$ be a sufficiently differentiable function in a neighborhood of its simple root α. If an initial approximation x_0 is sufficiently close to α and the varying parameter γ_k is computed by (15), then $R-$order of convergence of family (20) is at least $11\cdot 2^{n-3}$ for $n\geq 3$ and at least $\frac{1}{2}(5+\sqrt{33})\approx 5.372$ for $n=2$.*

Proof. Let us consider that $R-$order of convergence of the iterative sequences $\{w_{k,n}\}, \{w_{k,n-1}\}$ and $\{w_{k,n-2}\}$ be at least r, v and u respectively, that is

$$\varepsilon_{k,n-2}\sim\eta_{k,u}\varepsilon_k^u,\ \varepsilon_{k,n-1}\sim\eta_{k,v}\varepsilon_k^v,\ \varepsilon_{k+1}\sim\eta_{k,r}\varepsilon_k^r. \tag{21}$$

Hence,

$$\varepsilon_{k,n-2}\sim\eta_{k,u}(\eta_{k-1,r}\varepsilon_{k-1}^r)^u=\eta_{k,u}\eta_{k-1,r}^u\varepsilon_{k-1}^{ru}, \tag{22}$$

$$\varepsilon_{k,n-1}\sim\eta_{k,v}(\eta_{k-1,r}\varepsilon_{k-1}^r)^v=\eta_{k,v}\eta_{k-1,r}^v\varepsilon_{k-1}^{rv}, \tag{23}$$

$$\varepsilon_{k+1}\sim\eta_{k,r}(\eta_{k-1,r}\varepsilon_{k-1}^r)^r=\eta_{k,r}\eta_{k-1,r}^r\varepsilon_{k-1}^{r^2}. \tag{24}$$

Now, by using (18) we have

$$N_2'(w_{k,0})\sim f'(\alpha)(1-c_3\varepsilon_{k-1,n-2}\varepsilon_{k-1,n-1}) \tag{25}$$

and from (12)

$$1+\gamma_k f'(\alpha)\sim c_3\varepsilon_{k-1,n-2}\varepsilon_{k-1,n-1}. \tag{26}$$

Now, using (25) and (26) we get

$$\begin{aligned}\varepsilon_{k,n-2} &\sim \xi_{k,n-2}(1+\gamma_k f'(\alpha))^{2^{n-3}}\varepsilon_k^{2^{n-2}}\\ &\sim \xi_{k,n-2}(c_3\varepsilon_{k-1,n-2}\varepsilon_{k-1,n-1})^{2^{n-3}}(\eta_{k-1,r}\varepsilon_{k-1}^r)^{2^{n-2}}\\ &\sim \xi_{k,n-2}c_3^{2^{n-3}}(\varepsilon_{k-1,n-2}\varepsilon_{k-1,n-1})^{2^{n-3}}\eta_{k-1,r}^{2^{n-2}}(\varepsilon_{k-1}^r)^{2^{n-2}}\\ &\sim \xi_{k,n-2}c_3^{2^{n-3}}\eta_{k-1,r}^{2^{n-2}}(\eta_{k-1,u}\eta_{k-1,v})^{2^{n-3}}(\varepsilon_{k-1})^{(u+v)^{2^{n-3}}+r2^{n-2}}\end{aligned} \tag{27}$$

and

$$\begin{aligned}\varepsilon_{k,n-1} &\sim \xi_{k,n-1}(1+\gamma_k f'(\alpha))^{2^{n-2}}\varepsilon_k^{2^{n-1}}\\ &\sim \xi_{k,n-1}(c_3\varepsilon_{k,n-2}\varepsilon_{k,n-1})^{2^{n-2}}(\varepsilon_k)^{2^{n-1}}\\ &\sim \xi_{k,n-1}c_3^{2^{n-2}}(\varepsilon_{k,n-2}\varepsilon_{k,n-1})^{2^{n-2}}(\varepsilon_k)^{2^{n-1}}\\ &\sim \xi_{k,n-1}c_3^{2^{n-2}}\eta_{k-1,r}^{2^{n-1}}(\eta_{k,u}\eta_{k,v})^{2^{n-2}}(\varepsilon_{k-1})^{(u+v)^{2^{n-2}}+r2^{n-1}}.\end{aligned} \tag{28}$$

Similarly,

$$\begin{aligned}\varepsilon_{k,n} &\sim \xi_{k,n}(1+\gamma_k f'(\alpha))^{2^{n-1}}\varepsilon_k^{2^n}\\ &\sim \xi_{k,n}(c_3\varepsilon_{k-1,n-2}\varepsilon_{k-1,n-1})^{2^{n-1}}\varepsilon_k^{2^n}\\ &\sim \xi_{k,n}c_3^{2^{n-1}}\eta_{k-1,r}^{2^n}(\varepsilon_{k-1})^{(u+v)^{2^{n-1}}+r2^n}.\end{aligned} \tag{29}$$

Equating the error exponents of ε_{k-1} in the three pairs (22) and (27), (23) and (28) and (24) and (29), we obtain the following system of equations in unknown order u, v and r.

$$\begin{aligned}ru-(u+v)2^{n-3}-r2^{n-2} &= 0\\ rv-(u+v)2^{n-2}-r2^{n-1} &= 0\\ r^2-(u+v)2^{n-1}-r2^n &= 0.\end{aligned} \tag{30}$$

Solving the above system of equations, we have the solution $u=11\cdot 2^{n-5}$, $v=11\cdot 2^{n-4}$, $r=11\cdot 2^{n-3}$. Thus, the $R-$order of convergence of the family (20) is at least $11\cdot 2^{n-3}$ for $n\geq 3$.

For example, for $n = 3$ the family (20) has at least $R-$order 11 and at least $11 \cdot 2^{4-3} = 22$ for $n = 4$. The case for $n = 2$ is some what different as that of three-point and four-point formulae. We use the nodes $x_{k-1}(= w_{k-1,0})$, $w_{k-1}(= w_{k-1,1})$ and $x_k(= w_{k,0})$ to construct the Newton's interpolating polynomial and set $u = 1$, in (30), to obtain the following system:

$$\begin{aligned} rv - (v+1) - 2r &= 0 \\ r^2 - 2(v+1) - 4r &= 0 \end{aligned} \tag{31}$$

with the solution $v = \frac{1}{4}(5 + \sqrt{33})$ and $r = \frac{1}{2}(5 + \sqrt{33})$. Thus the family (20) is at least of $R-$order $\frac{1}{2}(5 + \sqrt{33}) \approx 5.372$ for $n = 2$. The proof is complete. □

Theorem 3.2. *Let $f(x)$ be a sufficiently differentiable function in a neighborhood of its simple root α. If an initial approximation x_0 is sufficiently close to α and the varying parameter γ_k is computed by (16) then $R-$order of convergence of $n-$point family with-memory (20) is at least $23 \cdot 2^{n-4}$ for $n \geq 4$ and at least $\frac{1}{2}(11 + \sqrt{137}) \simeq 11.352$ and 6 for $n = 3$ and $n = 2$ respectively.*

Proof. The proof is similar to the proof of Theorem 3.1 if we assume that $R-$order of iterative sequences $\{w_{k,n-3}\}$, $\{w_{k,n-2}\}$, $\{w_{k,n-1}\}$ and $\{w_{k,n}\}$ is u, v, y and r respectively. Hence, it is omitted. □

Theorem 3.3. *Let $f(x)$ be a sufficiently differentiable function in a neighborhood of its simple root α. If an initial approximation x_0 is sufficiently close to α and the varying parameter γ_k is computed by (17) then $R-$order of convergence of $n-$point with-memory family (20) is at least $47 \cdot 2^{n-5}$ for $n \geq 5$ and at least 23.34, 11.68 and 6 for $n = 4$, $n = 3$ and $n = 2$ respectively.*

Proof. Assume that $R-$order of iterative sequences $\{w_{k,n-4}\}$, $\{w_{k,n-3}\}$, $\{w_{k,n-2}\}$, $\{w_{k,n-1}\}$ and $\{w_{k,n}\}$ be u, v, y, l and r respectively. Then the proof is similar to the proof of Theorem 3.1. Hence, it is skipped over. □

Remark 3.1. *From the above results it is clear that the convergence order of the family of without memory $n-$point optimal methods (1) is increased from 4 to 6 for $n = 2$, 8 to 11.68 for $n = 3$, 16 to 23.34 for $n = 4$ and 32 to 47 for $n = 5$. We note that the order of convergence of the derivative free without memory family (1) is accelerated up to 50%. Generally, the convergence order of (1) is increased from 2^n to $2^n + 2^{n-1}$ that is the order of convergence of the family of with memory methods (20).*

4. **Numerical examples**

In this section, we test performance of the proposed with-memory $n-$point class of root solvers (20) (FNSM) to solve some smooth and non-smooth nonlinear functions taken from the literature. We compare the results of the without memory $n-$point families of Zafar et al. [13] (1), Kung and Traub [2] (KT) and Zheng et al. [14] (ZLH) (for $n = 3$) with their with-memory extensions, i.e. replacing the free parameter γ with the accelerating parameter γ_k in the without memory families. The parameter, γ_k is computed by (15), (16) and (17) according to availability of the points. All numerical computations are performed using the programming package Maple 16 with multiple-precision arithmetic and 1000 significant digits. We have considered the following test functions:

$$\begin{aligned} f_1(x) &= e^{-x^2}(x-2)(1 + x^3 + x^6), \; x_0 = 1.5, \; \alpha = 2, \\ f_2(x) &= x^2 - (1-x)^{25}, \; x_0 = 0, \alpha = 0.1437..., \\ f_3(x) &= xe^{x^2} - \sin^2 x + 3\cos x + 5, \; x_0 = -1, \; \alpha = -1.2076... \end{aligned}$$

Table 1: Comparison of With- and Without Memory Methods for $f_1, x_0 = 1.5$

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	$\lvert x_4-\alpha\rvert$	COC	EI
ZLH $n=3$						
$\gamma = 0.01$ (Without Memory)	1.01(−3)	4.09(−24)	3.01(−187)	0	8.00	1.68
$\gamma_k(15)$ (With-Memory)	1.01(−3)	4.79(−34)	1.72(−337)	0	10.00	1.77
$\gamma_k(16)$ (With-Memory)	1.01(−3)	2.70(−36)	8.29(−392)	0	10.91	1.81
$\gamma_k(17)$ (With-Memory)	1.01(−3)	1.69(−36)	2.76(−429)	0	11.98	1.86
KT $n=3$						
$\gamma = 0.01$ (Without Memory)	1.09(−3)	3.35(−23)	2.60(−179)	0	8.00	1.68
$\gamma_k(15)$ (With-Memory)	1.09(−3)	4.78(−33)	2.91(−327)	0	10.02	1.77
$\gamma_k(16)$ (With-Memory)	1.09(−3)	1.80(−35)	4.01(−383)	0	10.94	1.81
$\gamma_k(17)$ (With-Memory)	1.09(−3)	2.31(−35)	6.53(−415)	0	11.98	1.86
FNSM $n=3$						
$\gamma = 1$(Without Memory)	7.36(−4)	6.16(−25)	1.49(−193)	0	8.00	1.68
$\gamma_k(15)$ (With-Memory)	7.36(−4)	4.60(−34)	6.17(−348)	0	10.02	1.78
$\gamma_k(16)$ (With-Memory)	7.36(−4)	1.31(−36)	8.60(−396)	0	10.97	1.82
$\gamma_k(17)$ (With-Memory)	7.36(−4)	3.15(−38)	1.12(−449)	0	11.98	1.86

Tables 1-3 display the error of approximations to corresponding root of nonlinear functions ($|x_k - \alpha|$) for first four iterations of the methods, where $A(-E)$ denotes $A \times 10^{-E}$. In that case the root α is not exact, it is replaced by a more accurate value which has more number of significant digits than the assigned precision. The Tables also include computational order of convergence (COC) computed after first three iterations by the formula [12], COC$\approx \frac{\log|(x_{k+1}-\alpha)/(x_k-\alpha)|}{\log|(x_k-\alpha)/(x_{k-1}-\alpha)|}$, and efficiency index (EI) for each of the methods. In all numerical tests, initial value of the accelerating parameter $\gamma_0 = 0.01$ is used. From the Tables 1-3, we can see that, the computational order and efficiency of the existing without memory methods (1), (ZLH) and (KT) has been significantly increased by employing the accelerator γ_k. It is also observed that the computation of the parameter γ_k using (17) provides higher order and accuracy than the use of (15) and (16). For the case of f_2, FNSM with (17) produces the convergence order 12.22 which is higher than 7.98, i.e. the order of the its without memory version (1). For the solution of f_1 and f_3, the proposed with-memory family (FNSM) with (17) possesses the convergence order 11.98 and 12.00 for $n = 3$, respectively. Thus, we can conclude that the convergence order of the without memory family (1) has been increased from 2^n to $2^n + 2^{n-1}$. Numerical tests of f_1, f_2 and f_3 show that the presented with-memory family (FNSM) is also competitive with the with-memory extensions of existing families by Zheng et al. [14] (ZLH) and Kung and Traub [2] (KT). Moreover, the computational efficiency index of the presented with-memory family is 1.86 which is higher than 1.68, i.e. the efficiency index of the without memory family (1) for $n = 3$.

Table 2: Comparison of With- and Without Memory Methods for $f_2, x_0 = 0$

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	$\lvert x_4-\alpha\rvert$	COC	EI
ZLH $n=3$						
$\gamma = 0.01$ (Without Memory)	3.95(−2)	4.93(−6)	1.61(−37)	2.10(−288)	8.00	1.68
$\gamma_k(15)$ (With-Memory)	3.95(−2)	8.83(−11)	3.79(−96)	6.85(−947)	9.87	1.77
$\gamma_k(16)$ (With-Memory)	3.95(−2)	9.22(−8)	3.17(−69)	4.25(−744)	10.98	1.82
$\gamma_k(17)$ (With-Memory)	3.95(−2)	1.24(−8)	1.15(−87)	0	12.13	1.86
KT $n=3$						
$\gamma = 0.01$ (Without Memory)	4.61(−2)	1.40(−4)	5.99(−25)	6.75(−187)	8.09	1.68
$\gamma_k(15)$ (With-Memory)	4.61(−2)	9.86(−9)	2.04(−71)	3.02(−698)	10.00	1.77
$\gamma_k(16)$ (With-Memory)	4.61(−2)	1.67(−5)	7.56(−43)	1.45(−453)	10.85	1.81
$\gamma_k(17)$ (With-Memory)	4.61(−2)	9.54(−6)	1.56(−50)	8.90(−589)	12.15	1.86
FNSM $n=3$						
$\gamma = 0.01$ (Without Memory)	9.64(−3)	1.53(−11)	9.89(−82)	2.89(−643)	7.98	1.68
$\gamma_k(15)$ (With-Memory)	9.64(−3)	1.62(−8)	4.55(−71)	6.18(−696)	10.83	1.81
$\gamma_k(16)$ (With-Memory)	9.64(−3)	5.91(−12)	5.16(−114)	0	11.07	1.82
$\gamma_k(17)$ (With-Memory)	9.64(−3)	2.93(−12)	1.37(−128)	0	12.22	1.87

Table 3: Comparison of With- and Without Memory Methods for $f_3, x_0 = -1$

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	$\lvert x_4-\alpha\rvert$	COC	EI
ZLH $n=3$						
$\gamma = 0.01$ (Without Memory)	4.18(−6)	2.98(−43)	2.00(−339)	0	8.00	1.68
$\gamma_k(15)$ (With-Memory)	4.18(−6)	1.04(−59)	2.02(−595)	0	9.99	1.77
$\gamma_k(16)$ (With-Memory)	4.18(−6)	7.81(−63)	1.16(−686)	0	11.00	1.82
$\gamma_k(17)$ (With-Memory)	4.18(−6)	3.97(−66)	5.58(−786)	0	11.99	1.86
KT $n=3$						
$\gamma = 0.01$ (Without Memory)	2.12(−4)	2.99(−28)	4.63(−219)	0	8.00	1.68
$\gamma_k(15)$ (With-Memory)	2.12(−4)	5.53(−40)	1.41(−395)	0	9.99	1.77
$\gamma_k(16)$ (With-Memory)	2.12(−4)	1.79(−44)	1.99(−483)	0	10.95	1.81
$\gamma_k(17)$ (With-Memory)	2.12(−4)	5.83(−44)	9.42(−518)	0	11.98	1.86
FNSM $n=3$						
$\gamma = 0.01$ (Without Memory)	6.26(−7)	1.06(−50)	7.19(−401)	0	8.00	1.68
$\gamma_k(15)$ (With-Memory)	6.26(−7)	5.21(−67)	1.30(−667)	0	10.00	1.78
$\gamma_k(16)$ (With-Memory)	6.26(−7)	1.34(−69)	2.38(−759)	0	11.00	1.82
$\gamma_k(17)$ (With-Memory)	6.26(−7)	3.09(−72)	4.32(−856)	0	12.00	1.86

5. Dynamical behavior

In this section, we analyze the dynamical planes associated to the rational functions obtained by applying the iterative methods to complex functions in the complex plane using basin of attraction. The dynamical planes are obtained by using two different techniques on Matlab R2013a software as follows. By taking a rectangle $[-2,2]\times[-2,2]$ of the complex plane, we define a mesh of 1000×1000 initial approximations. The starting point is in the basin of attraction of a root to which the sequence of the iterative method converges with an error approximation lower than 10^{-5} and at most 30 iterations. In the first technique this initial point is assigned with a specific color which is already selected for the corresponding root. If the sequence of the iterative method converges in less number of iterations then the color will be more intense and if it is not converging to any of the roots after 30 iterations, then that initial point is marked with dark blue color. For the second technique, maximum number of iterations are 25 with an error estimation lower than 10^{-5} and each initial guess is assigned with a color depending upon to the number of iterations for the iterative method to converge to any of the root of the given function. In this technique we use colormap 'Hot'. The color of the initial point will be more intense if the sequence of the iterative method converges in less number of iterations and if it is not converging to any of the roots after 25 iterations, then initial point is assigned with black color. The proposed with-memory method (FNSM) and the with-memory method of Kung and Traub [2] (KT) are applied to the following complex functions: $p_1(z) = z^3 - 1$, with roots 1.0, $-0.5000 + 0.86605I$, $-0.5000 - 0.86605I$, $p_2(z) = z^5 - 1$, with roots 1.0, $0.3090 + 0.95105I$, $-0.8090 + 0.58778I$, $-0.8090 - 0.58778I$, $0.30902 - 0.95105I$.

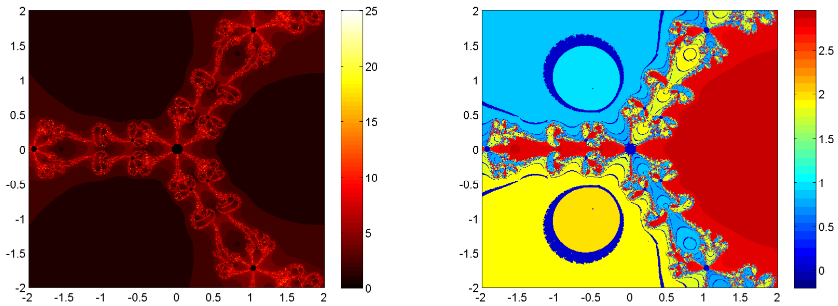

FIGURE 1. Dynamical Planes of KT using (16) for $n=3$ on p_1.

Dynamical planes of the with-memory methods (FNSM) and (KT) for $n=3$ and $\beta_0 = 0.01$ applied to the functions $p_1(z)$ and $p_2(z)$ are depicted in the figures 1-4. Two

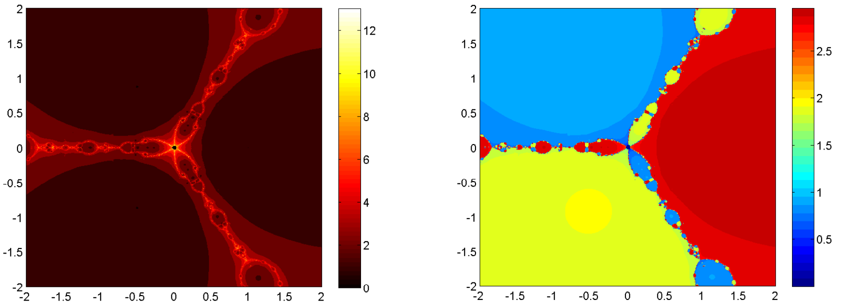

FIGURE 2. Dynamical Planes of FNSM using (16) for $n = 3$ on p_1.

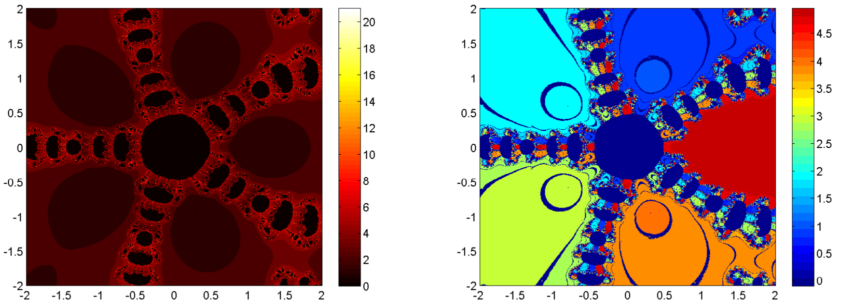

FIGURE 3. Dynamical Planes of KT using (16) for $n = 3$ on p_2.

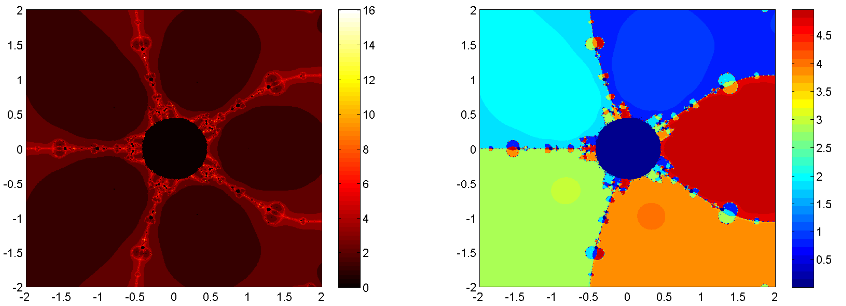

FIGURE 4. Dynamical Planes of FNSM using (16) for $n = 3$ on p_2.

types of attraction basins are given in all the figures. Color maps for both types are provided with each figure which show the root to which an initial guess converges and the number of iterations in which the convergence occurs. It can be observed from the dynamics that the appearance of wide darker region shows that the iterative method (FNSM) consumes less number of iterations and have wider regions of convergence in comparison with (KT). Hence, the proposed with-memory method (FNSM) is more reliable as its dynamical planes has less black and dark blue regions in comparison with the with-memory family of Kung and Traub [2] (KT).

6. Conclusions

In this work, we have proposed a with-memory extension of an existing without memory derivative free family of $n-$point optimal methods to solve nonlinear functions employing a self-accelerating parameter. The $R-$order of convergence of the without memory family (1) has been increased from 2^n to $2^n + 2^{n-1}$ for arbitrary n, without additional functional evaluations. The convergence speed is accelerated by using suitable variation of an accelerating parameter at each iterative step. This parameter is computed using the Newton's interpolating polynomials. Numerical results support the theoretical results and demonstrate the efficiency and robustness of proposed class of methods (20) in comparison with the existing with- and without memory methods as it is clear from the columns EI and $|x_i - \alpha|$ in Table 1-3. It can also be observed that the computational order of convergence is comparable with the existing methods as it is apparent from the COC column of the Tables 1-3. Finally, dynamical behavior of the proposed family is presented to conclude that the proposed family of methods is competitive with the already developed methods of same domain.

Acknowledgements. The authors are thankful to the referee for giving valuable comments and suggestions for the preparation of final version of this paper.

REFERENCES

[1] *J. Džunić and M.S. Petković*, On generalized multipoint root-solvers with-memory, J. Comput. Appl. Math., **236**(2012), 2909-2920.

[2] *H.T. Kung and J.F. Traub*, Optimal order of one-point and multipoint iteration, J. ACM, **21**(1974), 643–651.

[3] *B. Neta*, On a family of multipoint methods for nonlinear equations, Int. J. Comput. Math., **9**(1981), 353–361.

[4] *B. Neta and M.S. Petković*, Construction of optimal order nonlinear solvers using inverse interpolation, Appl. Math. Comput., **217**(2010), 2448–2455.

[5] *J.M. Ortega and W.C. Rheiboldt*, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[6] *M.S. Petković and L.D. Petković*, Families of optimal multipoint methods for solving nonlinear equations: A survey, Appl. Math. Comput., **4**(2010), 1–22.

[7] *M.S. Petković*, Remarks on a general class of multipoint root-finding methods of high computational efficiency, SIAM J. Numer. Anal., **49**(2011), 1317–1319.

[8] *M.S. Petković, J. Džunić and L.D. Petković*, A family of two-point methods with-memory for solving nonlinear equations, Appl. Anal. Discrete Math., **5**(2011), No.2, 298–317.

[9] *M.S. Petković, B. Neta, L.D. Petković and J. Džunić*, Multipoint Methods for Solving Nonlinear Equations, Elsevier, Waltham, Massachusetts, USA, 2013.

[10] *J.R. Sharma, R.K Guha and P. Gupta*, Some efficient derivative free methods with-memory for solving nonlinear equations, Appl. Math. Comput., **219**(2012), 699–707.

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

[12] *S. Weerakoon and T. G. I. Fernando*, A variant of Newton's method with accelerated third-order convergence, *Appl. Math. Lett.* **13**(8)(2000), 87–93 .

[13] *F. Zafar, N. Yasmin, S. Akram and M. D. Junjua*, A general class of root finding methods using rational interpolation, The Scientific World J., **2015**(2015), (DOI:10.1155/2015/934260).

[14] *Q. L. Zheng, J. Li and F. Huang*, Optimal Steffensen-type families for solving nonlinear equations, Appl. Math. Comput., **217**(2011), 9592-9597.