Journal of Prime Research in Mathematics Vol. **16(1)***(2020), 64 - 83*

A GENERAL FAMILY OF DERIVATIVE FREE WITH AND WITHOUT MEMORY ROOT FINDING METHODS

SAIMA AKRAM[1], FIZA ZAFAR[1], MOIN-UD-DIN JUNJA[2], NUSRAT YASMIN[1]

ABSTRACT. In this manuscript, we construct a general family of optimal derivative free iterative methods by using rational interpolation. This family is further extended to a family of with-memory methods with increased order of convergence by employing two free parameters. At each iterative step, we use a suitable variation of the free parameters. These parameters are computed by using the information from current and previous iterations so that the convergence order of the existing family is increased from 2^n to $2^n + 2^{n-1} + 2^{n-2}$ without using any additional function evaluations. To check the performance of newly developed iterative schemes with and without memory, an extensive comparison with the existing with- and without memory methods is done by taking some real world problems and standard nonlinear functions. Numerical experiments illustrate that the proposed family of methods with-memory retain better computational efficiency and fast convergence speed as compared to existing with- and without memory methods. The performance of the methods is also analyzed visually by using complex plane. Numerical and dynamical comparisons confirm that the proposed families of with and without memory methods have better efficiency, convergence regions and speed in contrast with the existing methods of the same kind.

Key words: Nonlinear equation, iterative methods, polynomiograph.
MSC: 65H04, 65H05.

1. Introduction

In this paper, the problem of finding numerical solution of nonlinear equations $f(x) = 0$ is addressed. Iterative procedures are widely used to solve this

[1] Centre for advanced studies in Pure and Applied Mathematics, Bahauddin Zakariya University Multan, Pakistan.
[2] Department of Mathematics and Statistics, Institute of Southern Punjab, Multan, Pakistan.
[*] Corresponding Author: saimaakram@bzu.edu.pk .

problem, (see, for instance, [1–6]). Traub [6], classified the iterative methods as one-step and multi-step schemes. Multi-step iterative methods are of much importance than one-step methods because they produce approximations of great accuracy. Multi-step iterative methods can be further categorized as without memory and with-memory iterative methods. Without memory root finding methods use information from the recent iteration only whereas the multi-step with-memory root finding methods use information from the recent as well as from previous iterations. One-step Steffensen's iterative method is a known improvement of Newton's method as it avoids to use the derivative unlike in the case of Newton's method. Newton's method is arguably worth important root finding method due to its quadratic convergence but its convergence depends badly on the choice of initial guess and behavior of the function in the vicinity of root. If initial guess is far from root or the function has improper behavior in the vicinity of root the Newton's method diverges. Therefore, the derivative free methods have vital importance. The concept of optimal iterative method was given by Kung and Traub [7] that is a multi-step iterative scheme, without memory, based on $n + 1$ functional evaluations could attain an optimal order of convergence 2^n. Ostrowski [2] defined that if O is the convergence order of an iterative method and n is the total number of functional evaluations per iteration, then the index $EI = O^{1/n}$ is known as efficiency index of an iterative method. Since multi-step iterative methods overcome theoretical limits of one-step methods concerning the order of convergence and the efficiency index, therefore several multi-step iterative schemes have been developed for solving nonlinear equations (see, for example, [8–10] and the overview [11]). Some optimal eighth order methods without memory can be found in [12–17]; these methods, among others, have been designed by using different techniques as composition of known schemes and elimination of function evaluations using interpolation, rational approximations, etc. or by freezing the derivatives and using weight function procedure[18].

Multi-step iterative methods with memory, that use information from the current and previous iterations, can increase the convergence order and the efficiency index of the multi-step iterative methods without memory with no additional functional evaluations. The increase in order of convergence is based on one or more accelerator parameters which appear in the error equations of methods without memory. For this reason, several multi-step with-memory iterative methods have been developed in recent years. For a background study regarding the acceleration of convergence order with memorization, one should see e.g. [11, 19, 20].

Steffensen [21] was the first who provided a without memory derivative free method by modifying the well known Newton's method given by:

$$x_{n+1} = x_n - \frac{f^2(x_n)}{f(x_n + f(x_n)) - f(x_n)}, n \geq 0. \tag{1}$$

Traub [6] provided the first with-memory iterative method by involving a free parameter in the Steffensen's iterative method (1) such that:

$$\begin{aligned} w_n &= x_n + p_n f(x_n), \\ x_{n+1} &= x_n - \frac{f(x_n)}{f[x_n, w_n]}, \ n \geq 0, \end{aligned} \tag{2}$$

where, x_0,p_0 are given, $p_{n+1} = \frac{-1}{M_1'(x_n)}$, $M_1(x_n) = f(x_n) + (x - x_n) f[x_n, w_n]$ and p_n is a self-accelerator. The with-memory iterative scheme (1) has convergence order 2.41421.

Recently, many researchers have developed iterative methods with memory based on existing optimal methods of different orders, mainly four (see for example [20, 22]), eight ([23–26], among others), sixteen (as [8]), or even general n-point schemes [27–29].

In this paper, we present a general procedure to obtain a general class of optimal derivative free without memory iterative methods by using rational interpolation along with its special cases which satisfy Kung and Tarub's Hypothesis . The proposed without memory schemes require $n+1$ function evaluations to achieve the convergence order 2^n and efficiency index $2^{\frac{n}{n+1}}$. Furthermore, the proposed class is extended to a general family of with-memory root finding methods. The contents of the paper are summarized as: Section 2, consists of construction of the general class of optimal derivative free without memory iterative methods and their error analysis. Section 3 presents the extension of the proposed without memory general class discussed in section 2 to with memory class of iterative methods and their analysis of convergence. Section 4 includes numerical comparisons of the proposed methods with existing methods of the same domain. Dynamical behavior is given in Section 5 for the better visualization of convergence regions and stability of proposed root finding methods. Concluding remarks are given in Section 6.

2. A General Class of Optimal Derivative Free Without Memory Iterative Methods

In this section, we present a general class of $n-$step without memory iterative methods using $n+1$ functional evaluations with order of convergence 2^n which satisfy the conjecture of Kung and Traub [7]. Consider a rational polynomial of degree n as follows:

$$r_n(t) = \frac{p_1(t)}{q_{n-1}(t)}, \tag{3}$$

where

$$p_1(t) = a_0 + a_1(t-x), \tag{4}$$

$$q_{n-1}(t) = 1 + d_1(t-x) + \cdots + d_n(t-x)^{n-1}, n \geq 1 \tag{5}$$

and

$$q_0 \equiv 1.$$

Thus, the proposed family of $n-$step iterative methods is given by:

$$\begin{aligned} w_0 &= x + \beta f(x), \\ w_1 &= x - \frac{f(x)}{f[w_0, x] + \eta f(w_0)}, \\ &\vdots \\ w_n &= w_{n-1} - \frac{f(w_{n-1})}{r'_{n-1}(w_{n-1})}, \ n \geq 2, \end{aligned} \tag{6}$$

where, $\beta, \eta \in \mathbb{R}\backslash\{-1\}$ and the unknowns $a_0, a_1, d_1, \cdots, d_{n-1}$ are determined by following interpolating conditions:

$$\begin{aligned} r_{n-1}(x) &= f(x), r_{n-1}(w_0) = f(w_0), r_{n-1}(w_1) \\ &= f(w_1), \cdots, r_{n-1}(w_{n-1}) = f(w_{n-1}). \end{aligned} \tag{7}$$

Thus, the root of non-linear equation is the root of the numerator of the rational interpolant of degree $n-1$ for the $n-$step method. For instance, for $n = 2$, we obtain the three-step iterative method given by:

$$\begin{aligned} w_0 &= x + \beta f(x), \\ w_1 &= x - \frac{f(x)}{f[w_0, x] + \eta f(w_0)}, \\ w_2 &= w_1 - \frac{f(w_1)}{r'_1(w_1)}, \end{aligned} \tag{8}$$

where,

$$r_1(t) = \frac{a_0 + a_1(t-x)}{1 + b_1(t-x)}, \tag{9}$$

such that

$$r_1(x) = f(x), r_1(w_0) = f(w_0), \ r_1(w_1) = f(w_1). \tag{10}$$

The unknowns a_0, a_1 and b_1 are determined by using conditions (10) as follows:

$$\begin{aligned} a_0 &= f(x), \\ a_1 &= f[w_1, x] + b_1 f(w_1), \\ b_1 &= \frac{f[w_0, x] - f[w_1, x]}{f(w_1) - f(w_0)}. \end{aligned} \tag{11}$$

Now using (9), we have the following two-step root finder:

$$\begin{aligned} w_0 &= x + \beta f(x), \\ w_1 &= x - \frac{f(x)}{f[w_0, x] + \eta f(w_0)}, \\ w_2 &= w_1 - \frac{f(w_1) q_1^2}{q_1 a_1 - (a_0 + a_1(w_1 - x)) b_1}, \end{aligned} \tag{12}$$

where, $q_1 = 1 + b_1(w_1 - x)$ and a_0, a_1, b_1 are given as in (11). Similarly, for $n = 3$, we obtain the following four-step iterative scheme:

$$\begin{aligned} w_0 &= x + \beta f(x), \\ w_1 &= x - \frac{f(x)}{f[w_0, x] + \eta f(w_0)}, \\ w_2 &= w_1 - \frac{f(w_1)}{r_1'(w_1)}, \\ w_3 &= w_2 - \frac{f(w_2)}{r_2'(w_2)}, \end{aligned} \tag{13}$$

where the rational interpolant $r_2(t)$ is given by:

$$r_2(t) = \frac{a_0 + a_1(t - x)}{1 + b_1(t - x) + b_2(t - x)^2}, \tag{14}$$

such that

$$r_2(x) = f(x),\ r_2(w_0) = f(w_0),\ r_2(w_1) = f(w_1),\ r_2(w_2) = f(w_2). \tag{15}$$

By using the conditions (15), we have:

$$\begin{aligned} a_0 &= f(x), \\ a_1 &= \frac{A f[w_0, x] + B f[w_1, x] + C f[w_2, x]}{P + Q + R}, \\ b_1 &= \frac{D f[w_1, x] + E f[w_2, x] + F f[w_0, x]}{P + Q + R}, \\ b_2 &= \frac{L f[w_2, x] + M f[w_1, x] + N f[w_0, x]}{P + Q + R}, \end{aligned} \tag{16}$$

where,

$$\begin{aligned}
A &= f(w_1)f(w_2)(w_2-w_1),\ B=f(w_0)f(w_2)(w_0-w_2),\\
C &= f(w_0)f(w_1)(w_1-w_0),\\
D &= (f(w_0)(w_0-x)-f(w_2)(w_2-x)),\\
E &= (f(w_1)(w_1-x)-f(w_0)(w_0-x)),\\
F &= (f(w_2)(w_2-x)-f(w_1)(w_1-x)),\\
L &= (f(w_0)-f(w_1)),\ M=(f(w_2)-f(w_0)),\\
N &= (f(w_1)-f(w_2)),\ P=f(w_1)f(w_2)(w_2-w_1),\\
Q &= f(w_0)f(w_2)(w_0-w_2), R=f(w_0)f(w_1)(w_1-w_0).
\end{aligned}$$

Hence by using (14), we have the following three-step iterative scheme:

$$\begin{aligned}
w_0 &= x+\beta f(x),\\
w_1 &= x-\frac{f(x)}{f[w_0,x]+\eta f(w_0)},\\
w_2 &= w_1-\frac{f(w_1)}{r_1'(w_1)},\\
w_3 &= w_2-\frac{f(w_2)q_2^2}{q_2a_1-(a_0+a_1(w_2-x))(b_1+2b_2(w_2-x))},
\end{aligned} \tag{17}$$

where, $q_2=1+b_1(w_2-x))+b_2(w_2-x)^2$ and a_0,a_1,b_1,b_2 are given as in (16). For convergence analysis of the proposed iterative methods (12) and (17) we, now, state the following theorem.

Theorem 1. *Let us consider $\alpha\in D$ as the simple root of the function $f: D\subseteq\mathbb{R}\to\mathbb{R}$, where f is adequately differentiable in the vicinity of its zero for interval D. If x is sufficiently close to α then for $\beta,\eta\in\mathbb{R}\backslash\{-1\}$, the iterative methods defined by (12) and (17) are of fourth order and eighth order convergence respectively with the error equations given by:*

$$e_{n+1}=(1+\beta c_1)^2(c_2+\eta)(\eta c_2c_1-c_3c_1+2c_2^2)e_n^4+O(e_n^5) \tag{18}$$

and

$$\begin{aligned}
e_{n+1} &= (1+\beta c_1)^4(\eta+c_2)^2(\eta c_2c_1-c_3c_1+2c_2^2)(\eta c_1c_2^2+c_4c_1^2-3c_1c_2c_3\\
&\quad +3c_2^3)\ e_n^8+O(e_n^9)
\end{aligned} \tag{19}$$

respectively, where,

$$c_k=\frac{1}{k!}\frac{f^{(k)}(\alpha)}{f'(\alpha)}\quad ,k=2,3,... \tag{20}$$

Proof. The proof is similar to those previously considered in [22, 27], Using Taylor expansions of the function, it can be easily proved that the iterative methods defined by (12) and (17) are of fourth order and eighth order convergence respectively. f. □

Remark 1. *From the above convergence results it can be seen that the iterative schemes (12) and (17) are fourth and eighth order convergent requiring three and four functional evaluations respectively. Hence, the proposed class of iterative schemes is optimal in the sense of hypothesis of Kung and Traub [7]. Also by using the general rational interpolating polynomial (3) with the scheme (7), we attain with the* $n-$*step, optimal* 2^{n-1} *order convergent iterative method involving* n *function evaluations with efficiency index* $2^{\frac{n}{n+1}}$.

3. Extension of Without Memory Root Finders to With-Memory

In this section, we present a general family of with-memory iterative methods based on our newly suggested family of without memory methods (6). For this, we approximate the involved parameters in such a way that the local order of convergence is increased. For example, the coefficient of e_n^8 in (19) disappears if we set $\beta = \frac{-1}{c_1}$ and $\eta = -c_2$, where $c_1 = f'(\alpha)$ and $c_k = \frac{f^{(k)}(\alpha)}{k!f'(\alpha)}, k \geq 2$. Hence, by replacing the free parameters β and η in (6) with self-accelerators β_k and η_k, we obtain the following $n-$step class of with-memory root finders:

$$\begin{aligned} w_{k,0} &= x_k + \beta_k f(x_k), \\ w_{k,1} &= x_k - \frac{f(x_k)}{f\left[w_{k,0}, x_k\right] + \eta_k f(w_{k,0})}, \\ &\vdots \\ w_{k,n} &= w_{k,n-1} - \frac{f(w_{k,n-1})}{r'_{n-1}(w_{k,n-1})}, \ n > 2, \end{aligned} \tag{21}$$

where,

$$\beta_k = \frac{-1}{N'_m(w_{k,0})}, m \geq 1, \eta_k = -\frac{N''_{m+1}(w_{k,1})}{2N'_{m+1}(w_{k,1})}, m \geq 1 \tag{22}$$

where, $N_m(w_{k,0})$ and $N_{m+1}(w_{k,1})$ are the Newton's interpolating polynomials of degree m and $m+1$ respectively passing through best available points, given by:

$$\begin{aligned} N_m(\Gamma) &= N_m(\Gamma; w_{k,0}, w_{k-1,n} w_{k-2,n}, ..., w_{k-1,n-m+1}), \\ N_{m+1}(\Gamma) &= N_m(\Gamma; w_{k,1}, w_{k,0}, w_{k-1,n} w_{k-2,n}, ..., w_{k-1,n-m+1}). \end{aligned} \tag{23}$$

The Newton's interpolating polynomial of degree m is defined as follows:

$$\begin{aligned} N_m(\Gamma) = \ & f(w_{k,0}) + f[w_{k,0}, w_{k-1,n}](\Gamma - w_{k,0})(\Gamma - w_{k-1,n}) + \cdots \\ & + f[w_{k,0}, w_{k-1,n} w_{k-2,n}, ..., w_{k-1,n-m+1}](\Gamma - w_{k,0})(\Gamma - w_{k-1,n}) \\ & \cdots (\Gamma - w_{k-1,n-m}). \end{aligned}$$

It is discussed and proved in [27] that the following error relation holds for n- point with-memory iterative methods, where α represents the real root of the function.

$$\begin{aligned} \epsilon_{k,j} &= w_{k,j} - \alpha \sim \Omega_{k,j} \prod_{i=0}^{j-1} \epsilon_{k,i}, \ j = 1, ..., n+1 \\ \text{where } \Omega_{k,1} &= 1 + \beta_k f'(\alpha)\ ,\ \Omega_{k,2} = c_2 + \eta_k \end{aligned} \tag{24}$$

and $\Omega_{k,j}(j \geq 3)$ is dependent on derivative of f at α, β_k, η_k and the type of interpolation used at $j-th$ step. For example when we use Newton's interpolating polynomial at $j-th$ step we get $\Omega_{k,j} = (-1)^j c_j + c_2 \Omega_{k,j-1}$ By using induction Equation (24) can take the following form:

$$\epsilon_{k,j} \sim (\Omega_{k,j} \prod_{i=0}^{j-1} \Omega_{k,i}^{2^{j-i-1}}) \epsilon_{k,0}^{2^{j-1}}. \tag{25}$$

Equating (24) and (25), we obtained the following error relations:

$$\begin{aligned} \varepsilon_{k,1} &\simeq w_{k,1} - \alpha \sim ((1 + \beta_k f'(\alpha))\epsilon_{k,0}, \qquad (26) \\ \varepsilon_{k,2} &\simeq w_{k,2} - \alpha \sim (c_2 + \eta_k)\epsilon_{k,0}\varepsilon_{k,1} \\ \epsilon_{k,j} &= w_{k,j} - \alpha \sim \Omega_{k,j} \prod_{i=0}^{j-1} \epsilon_{k,i}\ \ ,\ j = 3, ..., n+1 \end{aligned}$$

where $\Omega_{k,1} = (1 + \beta_k) f'(\alpha)$, $\Omega_{k,2} = c_2 + \eta_k$ and $\Omega_{k,j}(\ j \geq 3)$ depend on the type of interpolation used at $j-th$ step.It is important to mention that we have used Hermite interpolation to develop (21). To find the R -order of convergence of newly developed family (21) the knowledge of two entities are required. First entity is the error estimates of accelerating parameters η_k and β_k which are approximated by Newtonian polynomial given in (23) and thus factors appeared in first and second row of (26) can be written in [27] as:

$$1 + \beta_k f'(\alpha) \sim U_m \epsilon_{k-1,0}^{s_{n-m+1}+...+s_n}, 1 \leq m \leq n+1 \tag{27}$$

$$c_2 + p_k \sim V_m \epsilon_{k-1,0}^{s_{n-m+1}+...+s_n}, 1 \leq m \leq n+1, \tag{28}$$

where

$$U_m = (-1)^{m+1} c_{m+1} \prod_{j=1}^{m-1} G_{k-1,n-j}, 1 \leq m \leq n$$

$$U_{m+1} = ((-1)^n c_{n+2} + c_2 \Omega_{k-1,n+1} \prod_{j=1}^{n} G_{k-1,n-j},$$

$$V_m = (-1)^m \frac{c_{m+2}}{2} \prod_{j=0}^{m-1} G_{k-1,n-j}, 1 \leq m \leq n+1$$

$$G_{k-1,0} = 1, r_0 = 1$$

and $G_{k-1,n-j}$ is asymptotic error constant. From (27) and (28) it is evident that the accelerating parameters β_k and η_k have same order of accuracy. Second entity is to utilize these error estimates to find the $R-order$ of convergence of newly developed family (21) based on Hermite interpolation. For this let we have a sequence $\{y_{k,j}\}$ converges to zero α with $R-order$ of convergence "s". The error relation for this sequence can be represented mathematically as:

$$\epsilon_{k+1,0} = \epsilon_{k,n+1} \simeq G_{k,n+1}\epsilon_{k,0}^{s}, \tag{29}$$

where $G_{k,n+1}$ is asymptotic error constant which tends to G_{n+1} as $k \to \infty$. We now, study the estimation of $R-order$ of convergence of with-memory family (21). So, in accordance with (29) we have the following relation for the iterative sequence$\{y_{k,j}\}$:

$$\epsilon_{k,j} \simeq G_{k,j}\epsilon_{k,0}^{s_j}, 1 \leq j \leq n. \tag{30}$$

Where s_j is the number of information on the function $f(f, f', f'', ..., f^{(\lambda_i - 1)})$ taken at the point $y_{k,j}$.. By using these of relations (29) and (30) in the third relation of (26) we have:

$$\begin{aligned}
\epsilon_{k,j} &\simeq \Omega_{k,j}\prod_{i=0}^{j-1}\epsilon_{k,i} \simeq \Omega_{k,j}\epsilon_{k,0}\prod_{i=1}^{j-1}\epsilon_{k,i} \simeq \Omega_{k,j}\epsilon_{k,0}\prod_{i=1}^{j-1}G_{k,i}\epsilon_{k,0}^{s_j}, \\
&\simeq \Omega_{k,j}\prod_{i=0}^{j-1}G_{k,i}\ \epsilon_{k,0}\epsilon_{k,0}^{s_1}\epsilon_{k,0}^{s_2}\epsilon_{k,0}^{s_3}...\epsilon_{k,0}^{s_{j-1}}, \\
&\simeq H_{k,j}\epsilon_{k,0}^{1+s_1+s_2+...+s_{j-1}}, 3 \leq j \leq n+1, \\
&= H_{k,j}\epsilon_{k,0}^{s_j},
\end{aligned} \tag{31}$$

where $H_{k,j} \simeq \Omega_{k,j}\prod_{i=0}^{j-1}G_{k,i}$ and $s_j = 1 + s_1 + s_2 + ... + s_{j-1}, 3 \leq j \leq n$. Considering $j = n+1$ in (31) and then equating it with (29) we obtain:

$$\epsilon_{k,0}^{s} = \epsilon_{k,0}^{1+s_1+s_2+...+s_n},$$

So,

$$s = 1 + s_1 + s_2 + ... + s_n, \tag{32}$$

For $3 \leq j \leq n$ we already have:

$$s_j = 1 + s_1 + s_2 + ... + s_{j-1}. \tag{33}$$

The relations (32) and (33) yields:

$$\begin{aligned}
&.s = 2s_n = 2^{n-2}s_3 = 2^{n-2}(1+s_1+s_2).s_j = 2s_{j-1} = 2^{j-3}s_3 \\
&= 2^{j-3}(1+s_1+s_2), 3 \leq j \leq n.
\end{aligned} \tag{34}$$

For the sake of simplicity the $R-order$ of convergence is commutatively defined by another symbol τ_j and hence we obtain the following relation:

$$\tau_j = s_{n-j+1} + ... + s_n = \begin{cases} s - s_{n-j+1} & 1 \leq j < n \\ s-1 & j = n \\ s & j = n+1. \end{cases} \tag{35}$$

According to relation (27) and (28) the factors $(1+\beta_k f'(\alpha))$ and $(c_2+\eta_k)$ have same order of accuracy $\epsilon_{k-1,0}^{\tau_j}$,where τ_j is defined by (35). Hence,

$$\epsilon_{k,1} = O(\epsilon_{k-1,0}^{\tau_j}\epsilon_{k,0}) \text{ and } \epsilon_{k,2} = O(\epsilon_{k-1,0}^{\tau_j}\epsilon_{k,1}\epsilon_{k,0}) = O(\epsilon_{k,1}^2),$$

gives $s_2 = 2s_1$ and $s_3 = 1+3s_1$.Therefore, for $j \geq 2$, relation (34) can take the following form:

$$s = 2^{n-2}(1+3s_1) \ , \ s_j = 2^{j-3}(1+3s_1), 3 \leq j \leq n. \tag{36}$$

If we take β_k only as self accelerating parameter in (21) and keeping η_k as a constant, we have the following error relation,

$$s_2 = s_1 + 1. \tag{37}$$

By using relations (72) and (33) in (34) we get

$$s = 2^{n-1}(1+s_1) \text{ and } s_j = 2^{j-2}(1+s_1), 2 \leq j \leq n, n \geq 1. \tag{38}$$

Combining (26) and (30) for $j=1$, we have

$$\epsilon_{k,1} \simeq G_{k,1}G_{k-1,n+1}^{s_1}\epsilon_{k-1,0.}^{ss_1} \tag{39}$$

From the second relation of (26) by using (27) and (29), we have

$$\epsilon_{k,1} \simeq U_m\epsilon_{k-1,0}^{\tau_j}G_{k-1,n+1}\epsilon_{k-1,0}^{s} \simeq U_m\epsilon_{k-1,0}^{\tau_j}G_{k-1,n+1}\epsilon_{k-1,0}^{\tau_j+s}. \tag{40}$$

Equating the error relation (39) with (40) and then using (35), we have the following:

$$ss_1 = s + \tau_j = \begin{cases} 2s - s_{n-j+1}, & 1 \leq j < n \\ 2s-1, & j = n \\ 2s, & j = n+1. \end{cases} \tag{41}$$

By using the value of s and s_j in (38)we are able to get the relation for $s_1$as:

$$s_1 = \begin{cases} 2 - 2^{-j} \ , & 1 \leq j < n \\ \frac{1}{2}(1 + 2^{\frac{-n}{2}}\sqrt{9.2^{2n}-8}), & j = n \\ 2, & j = n+1. \end{cases} \tag{42}$$

Now using relation (42) in (36), we obtain the $R-order$ of convergence of $n-point$ with-memory family (21) as follows .

$$s = \begin{cases} 2^n - 2^{n-1} + 2^{n-2} - 3.2^{n-j-2} = 2^{n-j-2}(7.2^j - 3) \ , & 1 \leq j < n \\ 7.2^{n-3} + 2^{\frac{n}{2}-3}\sqrt{49.2^n - 48}, & j = n \\ 2^n + 2^{n-1} + 2^{n-2} = 1.75.2^n, & j = n+1, n \geq 2 \end{cases} \tag{43}$$

Now, we are able to establish the following conclusion of the above discussion.

Theorem 2. *Let x_0 be an initial approximation sufficiently close to a simple root α of a function f then the order of convergence of the family of n-point methods with self accelerating parameters β_k and η_k calculated by (22) for $1 \leq j \leq n+1$ is given by (43).*

Remark 2. *It can easily be seen from (43) that order of family of without memory methods (6) is increased from 2^n to $2^n + 2^{n-1} + 2^{n-2}$ i.e. from 8 to 14 for the three-point iterative scheme. It is worth mentioning and very attractive that the convergence order of newly developed family of without memory methods (6) is accelerated by 75% without increasing the number of functional evaluations.*

4. Numerical Results

In this section we give the comparison of the proposed families of without memory methods (NF-1) (17) and with-memory methods (NF-2) (21) with the existing iterative methods of Zheng et al. (ZM) (45) and Kung and Traub (KT) (44) for $n = 3$. Kung and Traub [7] presented the following iterative method:

$$\begin{aligned} w_{k,0} &= \psi_0(f)(x_k) = x_k,\ w_{k,-1} = \psi_{-1}(f)(x_k) = x_k + \eta_k f(x_k), k \geq 0, \\ y_{k,r} &= \psi_r(f)(x) = Q_n(0),\ n = 1, \ldots, m,\ \text{for } m > 0, \\ z_{k+1} &= y_{k,m} = \psi_m(f)(x_k), \end{aligned} \tag{44}$$

where, $m \in \mathbb{N}$, x_0 is given initial guess and $Q_n(s) = Q_n(f(w_{j,m})) = w_{j,m}$, where, $m = -1, 0, \ldots, r-1$ is the nth degree inverse interpolation polynomial. To compute the free parameter η_k, Newton's interpolation polynomial of appropriate degree is used.

Zheng et al. [10] (ZLH) presented the following $n-$step class of iterative methods given by:

$$
\begin{aligned}
z_{k,0} &= x_k,\ z_{k,-1} = z_{k,0} + \eta_k f(z_{k,0}), k \geq 0, \\
z_{k,1} &= z_{k,0} - \frac{f(z_{k,0})}{f[z_{k,0}, z_{k,-1}]}, \\
z_{k,2} &= z_{k,1} - \frac{f(z_{k,1})}{f[z_{k,1}, z_{k,0}] + f[z_{k,1}, z_{k,0}, z_{k,-1}](z_{k,1} - z_{k,0})}, \\
&\vdots \\
z_{k,n} &= z_{k,n-1} \\
&\quad - \frac{f(z_{k,n-1})}{f[z_{k,n-1}, z_{k,n-2}] + \sum\limits_{j=1}^{n-1} f[z_{k,n-1}, \ldots, z_{k,n-2-j}] \prod\limits_{i=1}^{j} (z_{k,n-1} - z_{k,n-i-j})},
\end{aligned}
\tag{45}
$$

where x_0 is initial approximation and the Newton's interpolation polynomial is used to compute the parameter η_k.

Example 1. *Consider the isothermal continuous stirred tank reactor (CSTR). Components A and R are fed to the reactor at rates of Q and q-Q respectively. The following reaction scheme develops in the reactor (see [30]):*

$$
\begin{aligned}
A + R &\rightarrow B \\
B + R &\rightarrow C \\
C + R &\rightarrow D \\
D + R &\rightarrow E
\end{aligned}
$$

The problem was analyzed by Douglas [31] in order to design simple feedback control systems. In the analysis, he gave the following equation for the transfer function of the reactor:

$$
K_C = \frac{2.98(y + 2.25)}{y^4 + 11.50y^3 + 47.49y^2 + 83.06325y + 51.23266875} = -1,
$$

where K_C is the gain of the proportional controller. The control system is stable for values of K_c that yields roots of the transfer function having negative real part. If we choose $K_C = 0$, we get the poles of the open-loop transfer function as roots of the nonlinear equation:

$$
f_1(y) = y^4 + 11.50y^3 + 47.49y^2 + 83.06325y + 51.23266875 = 0
$$

given as:

$$
x = -1.45, -2.85, -2.85, -4.35.
$$

So, we see that there are two simple roots. We take $\alpha = -1.45$.

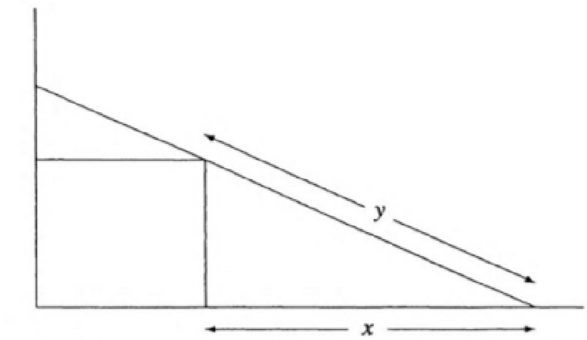

Figure 1. Beam Positioning Problem

Example 2. *We consider a beam positioning problem (see [32]) where a 4 meters long beam is leaning against the edge of the cubical box with sides of length 1 meter each such that one of its end touches the wall and the other touches the floor as shown in Figure 1. What should be the distance along the floor from the base of the wall to the bottom of the beam. Let y be the distance in meters along the beam from the floor to the edge of the box and let x be the distance in meters from the bottom of the box to the bottom of the beam. Then, we have the following equation:*

$$f_2(y) = y^4 + 2y^3 - 14y^2 + 2y + 1 = 0.$$

The positive solution of the equation 0.3621999926 and 2.7609056329 are the solutions to the beam positioning problem.

In addition, we take the following test functions from the literature for the comparison.

$$\begin{aligned} f_3(y) &= e^{-y^2}(y-2)(1+y^3+y^6),\ y_0 = 1.8,\ \omega = 2, \\ f_4(y) &= \sin y - \frac{y}{100},\ y_0 = 0.5,\ \omega = 0, \\ f_5(y) &= y^5 + y^4 + 4y^2 - 15,\ y_0 = 1.6,\ \omega = 1.3474..., \\ f_6(y) &= (y-1)(y^6 + y^{-6} + 4)\sin y^2,\ y_0 = 1.3,\ \omega = 1. \end{aligned}$$

We used the programming software Maple 16 for all numerical calculations with multiple-precision arithmetic. The error of the approximation to the corresponding root of nonlinear functions are shown in Table 1, where $E(-h)$ denotes $E \times 10^{-h}$. In Table 1, it is demonstrated that the proposed family of methods (NF-1) (17) and (NF-2) (21) has a consistent convergence behavior.

TABLE 1. Comparison of With and Without Memory Root Finders

Error	Without Memory ($\beta = -0.01, \eta = 0.1$)			With Memory ($\beta_0 = -0.01, \eta_0 = 0.1$)		
	ZM	KT	NF-1	ZM	KT	NF-2
$f_1,\ y_0 = -1.2$						
$\lvert y_1 - \omega \rvert$	2.84(−2)	4.54(−2)	6.79(−4)	3.11(-4)	1.03(-5)	6.79(-5)
$\lvert y_2 - \omega \rvert$	5.27(−16)	5.91(−16)	6.58(−32)	1.62(−44)	2.74(−55)	1.10(−65)
$\lvert y_3 - \omega \rvert$	6.54(−128)	2.20(−128)	3.65(−256)	4.95(−528)	3.37(−660)	2.22(−910)
$f_2,\ y_0 = 1.0$						
$\lvert y_1 - \omega \rvert$	3.11(-4)	1.03(-5)	6.79(-5)	3.11(-4)	1.03(-5)	6.79(-5)
$\lvert y_2 - \omega \rvert$	5.27(−32)	5.91(−38)	6.58(−40)	1.62(−40)	2.74(−40)	1.10(−60)
$\lvert y_3 - \omega \rvert$	6.54(−256)	2.20(−304)	3.65(−320)	4.95(−320)	3.37(−480)	2.22(−840)
$f_2,\ y_0 = 1.8$						
$\lvert y_1 - \omega \rvert$	7.88(−7)	1.19(−6)	2.98(−6)	7.88(−7)	1.19(−6)	3.00(−6)
$\lvert y_2 - \omega \rvert$	5.27(−49)	5.91(−44)	6.58(−45)	1.62(−38)	2.74(−46)	1.10(−80)
$\lvert y_3 - \omega \rvert$	6.54(−389)	2.20(−358)	3.65(−360)	4.95(−244)	3.37(−390)	2.22(−1120)
$f_3,\ y_0 = 0.5$						
$\lvert y_1 - \omega \rvert$	2.04(−7)	2.13(−6)	7.64(−8)	1.83(−7)	2.13(−6)	7.64(−8)
$\lvert y_2 - \omega \rvert$	2.57(−55)	1.24(−53)	5.74(−69)	4.08(−67)	4.36(−69)	3.47(−104)
$\lvert y_3 - \omega \rvert$	5.45(−613)	1.91(−500)	4.35(−619)	1.25(−642)	1.37(−709)	1.25(−1418)
$f_4,\ y_0 = 1.6$						
$\lvert y_1 - \omega \rvert$	3.03(−7)	1.73(−6)	1.99(−6)	3.03(−7)	1.73(−6)	1.99(−6)
$\lvert y_2 - \omega \rvert$	4.56(−44)	4.47(−45)	1.74(−46)	2.95(−41)	7.36(−48)	1.48(−78)
$\lvert y_3 - \omega \rvert$	1.21(−365)	9.00(−363)	5.89(−367)	1.05(−262)	5.60(−399)	3.33(−1092)
$f_5,\ y_0 = -2$						
$\lvert y_1 - \omega \rvert$	9.96(−4)	2.96(−4)	2.78(−4)	9.96(−5)	2.96(−4)	2.78(−4)
$\lvert y_2 - \omega \rvert$	1.94(−25)	3.06(−27)	8.45(−28)	1.77(−25)	7.12(−29)	3.25(−52)
$\lvert y_3 - \omega \rvert$	4.18(−215)	4.07(−201)	6.21(−216)	3.32(−153)	1.43(−227)	2.17(−728)

Comparing the results of proposed families of with and without memory methods, it can be seen that the new families (17) and (21) have better performance than the existing families (KT) (44) and (ZM) (45).

5. Dynamical Analysis

For visualizing the dynamical properties of associated rational functions of iterative methods, we use basins of attraction technique. This technique gives an important information about the stability and reliability of the iterative methods. In this section, we give the dynamical comparison of the proposed with-memory iterative method (21) (NF-2) and with-memory method of Kung and Traub (44) (KT) by showing their convergence planes for $n = 3$, $\beta_0 = -0.01$, $\eta_0 = 0.1$. We use the technique of basins of attraction and apply iterative methods to complex functions to get the planes of associated rational function. Two different methods are considered to obtain the dynamical planes on Matlab R2018 programming software. In the complex plane we use a rectangle $[-2, 2] \times [-2, 2]$ and initial approximations by a mesh of 1000×1000 points. The initial guess for a zero lies in the basins of attraction if the root finder converges in 30 iterations or error estimation less than 10^{-5}. In the first method points are coloured according to the number of iterations needed to find a root, so those method visualize speed of convergence. A particular color is allotted to each initial guess. The color will be more intense if the root finder has faster convergence. The initial guess is assigned with dark blue color if after 30 iterations it is not converging to any of the zeros. In the second method, each root gets the distinct color and the points are colored based on the distance to the nearest root, i.e., after finding a root we find a closest root and assign its color to the starting point. Thus the second method shows only to which of the roots the method converges. we use error approximation less than 10^{-5} and 25 iterations. In this method, depending upon the number of iterations for the convergence of the root finder to any of the root of given nonlinear polynomial each initial approximation is allotted a different color. Again, the color will be more intense if the iterative method has faster convergence. The initial guess is assigned with black color if, after 25 iterations it is not converging to any of the roots. The complex polynomials used for the dynamical comparison of the proposed with-memory method (21) (NF-2) and Kung-Traub family of with-memory methods [7] (44) are given below with their roots:

$$
\begin{aligned}
p_1(z) &= z^3-1,\ \omega = 1.0, -0.5000+0.86605I, -0.5000-0.86605I \\
p_2(z) &= z^5-1,\ \omega = 1.0,\ 0.3090+0.95105I,\ -0.8090+0.58778I, \\
&\quad -0.8090-0.58778I, 0.30902-0.95105I \\
p_3(z) &= z^6-\frac{1}{2}z^5+\frac{11(i+1)}{4}z^4-\frac{3i+19}{4}z^3+\frac{5i+11}{4}z^2 \\
&\quad +\frac{i-11}{4}z+\frac{3}{2}-3i, \\
\omega &= -1.0068+2.0047i, 0.0281+0.9963i, 0.0279-1.5225i, \\
&\quad 1.0235-0.9556i, 0.9557-0.0105i, -0.5284-0.5125i.
\end{aligned}
$$

Figures $2-7$ consists of the dynamical planes of the presented family of methods with-memory (21) (NF-2) and the family of methods with-memory by Kung and Traub [7] (44). In all figures two kinds of attraction basins are shown. It can be seen from the color maps of both types of figures that to which root an initial guess converges and in how much number of iterations the iterative sequence convergence. In all figures appearance of wider darker regions confirms that the proposed family of root finding methods (21) (NF-2) use less number of iterations in contrast with (44). Since dynamical planes of the presented family of root finders has less dark blue and black regions in contrast with (44) (KT), it means that the presented family of methods (21) (NF-2) is the better alternate to existing families of the same domain.

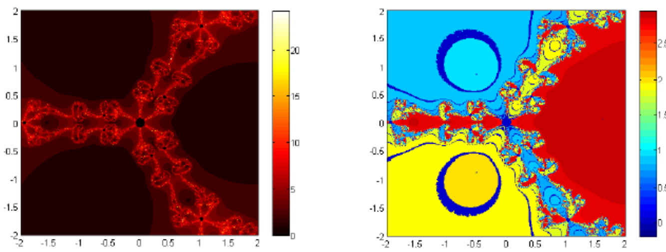

FIGURE 2. Convergence Regions of $p_1(z)$ for (KT)

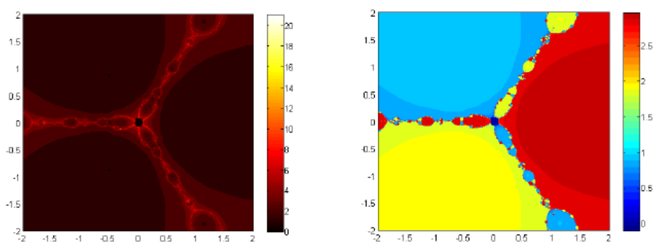

FIGURE 3. Convergence Regions of $p_1(z)$ for (NF-2)

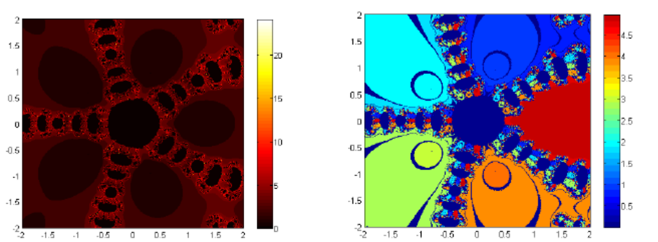

FIGURE 4. Convergence Regions of $p_2(z)$ for (KT)

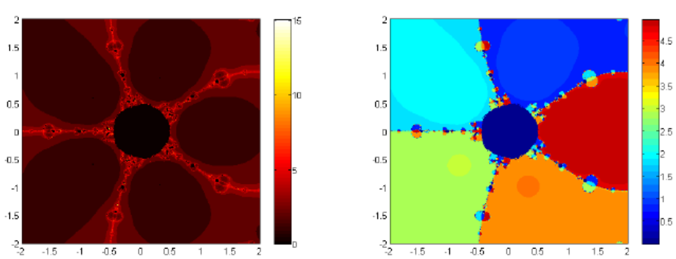

FIGURE 5. Convergence Regions of $p_2(z)$ for (NF-2)

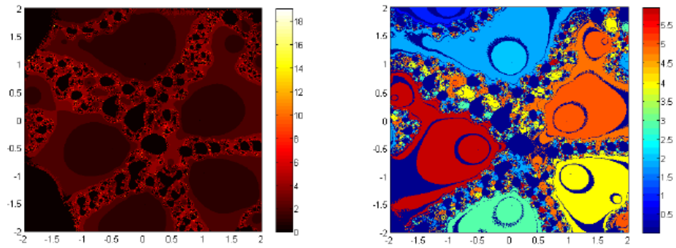

FIGURE 6. Convergence Regions of $p_3(z)$ for (KT)

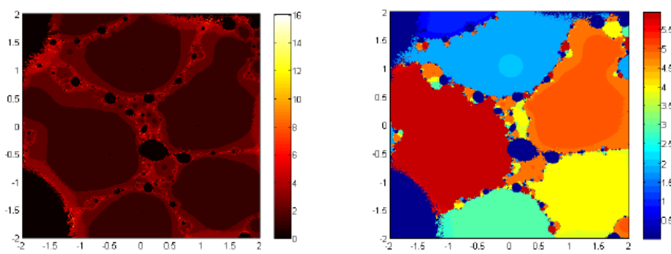

FIGURE 7. Convergence Regions of $p_3(z)$ for (NF-2)

6. Concluding Remarks

We have presented a new $n-$step general derivative free family of without memory root finding methods as well as its extension to the general family of with-memory methods. It is shown that the $R-$order of convergence of the new family of without memory methods (6) has been boosted from 2^n to $2^n + 2^{n-1} + 2^{n-2}$. The speed of convergence is accelerated by using suitable variation of two free parameters in each iterative step. For the computation of the self-accelerating parameters, Newton's interpolation polynomials of third and fourth degree are used. Numerical and dynamical comparisons confirm that the proposed families of with and without memory methods have better efficiency, convergence regions and speed in contrast with the existing methods of the same kind. Dynamical behavior of the proposed family of with-memory methods demonstrates that the proposed family is more efficient and reliable as compare to the previous families of the same kind.

References

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

[2] Ostrowski, A. M., (1960). *Solution of Equations and Systems of Equations.* Academic Press, New York.

[3] Nawaz, M., Naseem, A., & Nazeer, W., (2018). New iterative method using variational iteration technique and their dynamical behavior. *Open Journal of Mathematical Abalysis., 2*(2), 01-09.

[4] Saqib, M., & Iqbal, M., (2017). Some multi-step iterative methods for solving nonlinear equations-1. *Open Journal of Mathematical Sciences, 1*(1), 34-43.

[5] Saqib, M., Majeed, Z., Quraish, M., & Nazeer, W., (2018). A new third-order method for solving nonlinear equations. *Open Journal of Mathematical Analysis., 2*(1),. 01-07.

[6] Traub, J., F., (1964). *Iterative methods for the solution of equations.* Prentice-Hall, Englewood Cliffs, N.J.

[7] Kung, H.T., & Traub, J.F. (1974). Optimal oder of one point and multipoint iteratio. *Journal of association for computing machinery., 21*, 643-651.

[8] Sharma, J.R., & Gupta, P., (2014). On some highly efficient derivative free methods with and without memory for solving nonlinear equations. *Int. J. Comput. Methods., 12*(1) . 1350093, 28 pages.

[9] Zafar, F., Yasmin, N., Akram, S., & Junjua, M., (2015). A general class of derivative free optimal root finding methods based on rational interpolation. *Sci. World J., 2015*, Article ID 934260, 12 pages.

[10] Zheng, Q., Li, j., & Huang, F., (2011). An optimal Steffensen-type family for solving nonlinear equations. *Appl. Math. Comput., 217*, 9592–9597.

[11] Petković, M. S., Neta, B., Petković, L. D., & Džunić, J. (2013). *Multipoint Methods for Solving Nonlinear Equations.* Elsevier, Amsterdam.

[12] Cordero, A., & Torregrosa, J.R., (2015). Low-complexity root-finding iteration functions with no derivatives of any order of convergence, Comput. *Appl. Math. 275,* 502–515.

[13] Petković, L.D., Petković, M.S., & Džunić, J., (2010). A class of three-point root-solvers of optimal order of convergence. *Appl. Math. Comput. 216*, 671–676.

[14] Sharifi, M., Karimi, S., Haghani, K. , Arab, M., & Shateyi, S., (2014). On a new iterative scheme without memory with optimal eighth order. *The Scientific World J., 2014*, 6 pages. Article ID 727490.

[15] Sharma, J.R., & Sharma, R.,(2010). A new family of modified Ostrowski's methods with accelerated eighth order convergence. *Numer. Algor. 54*, 445–458.

[16] Soleymani, F., Vanani, K., S., & Paghaleh, M., J., (2012). A class of three-step derivative-free root solvers with optimal convergence order. *Appl. Math., 2012,* Article ID 568740, 15 pages.

[17] Wang, X., & Liu, L., (2010). New eighth-order iterative methods for solving nonlinear equations. *Comput. Appl. Math., 234*, 1611–1620.

[18] Junjua, M., Zafar, F., & Yasmin, N., (2019). Optimal derivative- free root finding methods based on inverse interpolation. *Mathematics,7*(2), (2019).

[19] Petković, M. S., Neta, B., Petković, L. D., & Džunić, J., (2014) Multipoint methods for solving nonlinear equations: a survey. *Appl. Math. Comput., 226*, 635–660.

[20] Petković, M.S., Džunić, J., & Petković, L.D., (2011). A family of two-point methods with-memory for solving nonlinear equations. *Appl. Anal. Discrete Math., 5*, 298–317.

[21] Steffensen, I., F., (1933). Remarks on iteration. *Skand. Aktuarietidskr., 16*, 64-72.

[22] Cordero, A., Lotfi, T., Bakhtiari, P., & Torregrosa, J. R., (2015). An efficient two-parametric family with-memory for nonlinear equations. *Numer Algor, 68*, 323-335.

[23] Cordero, A., Junjua, M., Torregrosa, J. R., Yasmin, N. & Zafar, F., (2018). Efficient four-parametric with-and-without-memory iterative methods possessing high efficiency indices. *Math. Probl. Eng.. 2018*, Article ID 8093673, 12 pages.

[24] Džunić, J., Petković, M.S., & Petković, L.D., (2012). Three-point methods with and without memory for solving nonlinear equations. *Appl. Math. Comput., 218*, 4917–4927.

[25] Lotfi, T., Soleymani, F., & Assari, P., (2015). On the construction of some tri-parametric iterative methods with-memory. *Numer. Algor., 70*(4), 835–845.

[26] Soleymani, F., Lotfi, T.,Tavakoli, E., & Haghani, K., F., (2015). Several iterative methods with memory using self-accelerators. *Appl. Math. Comput., 254*, 452–458.

[27] Džunić, J., & Petkovic, M. S., (2014). On generalized biparametric multipoint root finding methods with-memory. *Journal of Computational and Applied Mathematics., 255*, 362-375.

[28] Junjua, M., Zafar, F., Yasmin, N.& Akram, S., (2017). A general class of derivative-free with-memory root solvers. *U.P.B. Sci. Bull., Series A, 79*(4), 19-28.

[29] Wang, X., & Zhang, T., (2015). Efficient n-point iterative methods with memory for solving nonlinear equations. *Numerical Algorithms 70*(2), 357–375.

[30] Constantinides, A., & Mostoufi, M. (1999). *Numerical Methods for Chemical Engineers with MATLAB Applications.* Prentice Hall PTR, New Jersey.

[31] Douglas, J. M., (1972). *Process Dynamics and Control.* Vol. 2, Prentice Hall, Englewood Cliffs, NJ.

[32] Zachary, J. L., (2012). *Introduction to scientific programming: Computational problem solving using Maple and C.* Springer, New York.