.... manuscript No.
(will be inserted by the editor)

Construction of iterative adaptive methods with memory with 100% improvement of convergence order

Vali Torkashvand, Mohammad Ali Fariborzi Araghi

the date of receipt and acceptance should be inserted later

Abstract An efficient family of the recursive methods of adaptive is proposed for solving nonlinear equations, is developed such that all previous information are applied. These methods have reached the maximum degree of convergence improvement of 100%, and also have an efficiency index of 2. Three families have been examined from Steffensen-Like single, two, and three-step methods that have used $2, 3$ and 4 parameters respectively. Numerical comparisons are made with other existing methods one-, two-, three-, and four-point to show the performance of the convergence speed of the proposed method and confirm theoretical results.

Keywords Adaptive method with memory; Accelerator parameter; Nonlinear equations; Newton's interpolatory polynomial; Order convergence

Mathematics Subject Classification (2000) 65B99;41A25;65H05;34G20.

1 Introduction

It is often necessary in scientific and engineering practices to find a root of a polynomial or a nonlinear equation. Undoubtedly, Traub is the pioneer in classifying iterative methods for solving such equations as one or multi point [43]. It is well-known that Newton's method is one of the most common iterative methods to approximate the solution α of $f(x) = 0$ is of great importance [30]. However, the condition of derivative existence for function f in a neighborhood of the required root is mandatory indeed for convergence of Newton's method, which restricts its applications in practice. To overcome on this problem, Steffensen replaced the first derivative of the function in the Newton's iterate by forward finite difference approximation. Steffensen-type methods without using derivatives, only compute divided differences and can be used

Corresponding author:V.Torkashvand , torkashvand1978@gmail.com
Young Researchers and Elite Club, Shahr-e-Qods Branch, Islamic Azad University, Tehran, Iran

M.A.Fariborzi Araghi
Department of Mathematics, Central Tehran Branch, Islamic Azad University, Tehran, Iran

for nondifferentiable problems. Traub in his book proved the best one point iterative method should achieve order of convergence n using n function evaluations. In trying to furnish multi-point methods of various orders, the Kung-Traub conjecture [23] is a crucial part of the development. On the basis of this hypothesis, a multi-point iteration without memory using n evaluations per full cycle possesses the maximal order of convergence 2^{n-1}, which is called the optimal order. Following the Kung and Traub conjecture, many authors tried to construct optimal multipoint methods without memory [1–9, 11, 13, 17, 19–23, 29, 31, 33, 34, 36–38, 40, 41, 44, 47]. Traub developed the first method with memory by applying Steffensen's method [40], and increased the order of convergence of this method from 2 to 2.41 (improvement 20.5%)without using any new information, and only by reusing the information of the previous step. Traub presented his memory method by entering a free parameter to the Steffensen's method as follows:

$$\begin{cases} \gamma_k = -\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}, \; k = 1, 2, 3, \cdots, \\ x_{k+1} = x_k - \frac{\gamma_k f(x_k)^2}{f(x_k + \gamma_k f(x_k)) - f(x_k)} \; k = 0, 1, 2, \cdots. \end{cases} \tag{1}$$

To see more related papers in the with memory methods of study, the readers might refer to [10,12,14–16,24–28,32,39,43,45,46]. We develop an adaptive method with memory; i.e., that uses the information not only from the last two steps, but also from the all previous iterations. This technique enables us to achieve the highest efficiency both theoretically and practically. Adaptive methods with memory have efficiency index 2, hence competes all the existing methods without and with memory in the literature. It should be noted that improvement of the degree of convergence up to 100% is mentioned in references [42], but these families are different from the mentioned methods. The computational efficiency in the sense of Ostrowski-Traub [31, 43], of an iterative method of the order p, requiring n function evaluations per iteration, is frequently calculated using the Ostrowski-Traub's efficiency index $E(p, n) = p^{1/n}$. We later compare both numerical performances and efficiency index of the proposed method with some significant methods to show our claims. To achieve and remodify the optimal one-, two-, three-steps methods, we approximate and update the introduced accelerator parameters in each iteration by suitable kind and optimal of Newton's interpolation.
The main objective of this paper is to achieve the highest efficiency index, 2, without imposing an evaluation of the function. Contents of the paper are summarized in what follows. In the next section, deals with modifying the optimal one-, two-, three-points methods without memory introduced by Zheng et al. [47], Soleymani et al. [39], and Lotfi-Assari. [27]. In Section 3, with memory methods with maximum self-referential parameters (one, two, and three) are presented for one, two, and three-step methods, respectively. The new class of recursive with methods of adaptive is supported with detailed proof in this section to verify the construction theoretically. Numerical examples are given in Section 4 to illustrate convergence behavior of our methods for simple roots. Finally, a short conclusion is given in the last section.

2 Modified Steffensen-Like Methods

2.1 One step method by Zheng et al.

In this section, we deal with modifying one-point method without memory by Zheng et al. [47], such that the error equation has two accelerators. Zheng et al.'s method has the iterative expression as follows:

$$\begin{cases} w_k = x_k + \gamma f(x_k),\ k = 0, 1, 2, \ldots, \\ x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \end{cases} \tag{2}$$

where $\gamma \in \mathbb{R}$ is nonzero arbitrary parameter. To transform Eq. (2) into a method with memory, with two accelerators :

$$\begin{cases} w_k = x_k + \gamma_k f(x_k),\ k = 0, 1, 2, \cdots, \\ x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k] + q_k f[w_k]}, \end{cases} \tag{3}$$

where γ and q are arbitrary nonzero parameters. In what follows, we present the error equation of Eq. (3).

Theorem 1 *Let $I \subseteq \mathbb{R}$ be an open interval, $f : I \to \mathbb{R}$ be a scalar function which has a simple root α in the open interval I, and also the initail approximation x_0 is sufficiently close the simple zero, then, the one-step iteration method (3) has two-order satisfies the following error equation:*

$$e_{k+1} = (1 + f'(\alpha)\gamma)(q + c_2)e_k^2 + O(e_k^3). \tag{4}$$

Proof: Using symbolic computation the following code written in the computational software pakage Mathematica is given. We emphasize that this proof is different from the given by [14, 47].
$In[1] : f[e\_] = fla(e + \sum_{i=2}^{3} c_i e^i);$
$In[2] : ew = e + \gamma Series[f[e], \{e, 0, 3\}]//Fullsimplify$
$Out[2] : (1 + \gamma fla)e + O[e]^2$
$In[3] : f[x\_, y\_] = \frac{f[x]-f[y]}{x-y};$
$In[4] : e_{k+1} = e - Series[\frac{f[e]}{f[e,ew]+qf[ew]}, \{e, 0, 3\}]//Fullsimplify$
$Out[4] : (1 + f'(\alpha)\gamma)(q + c_2)e^2 + O(e^3).$ □

2.2 Acceleration of the modified Soleymani et al.'s method

In this section, concerns with modifying Soleymani et al.'s method (SLTKM) [39], so that it could be considered for the proposed scheme in the next section. Let recall the mentioned method:

$$\begin{cases} w_k = x_k + \gamma f(x_k),\ y_k = x_k - \frac{f(x_k)}{f[x_k,w_k]+qf(w_k)},\ k = 0, 1, 2, \cdots, \\ x_{k+1} = y_k - \frac{f(y_k)}{f[w_k,y_k]+qf(w_k)+\lambda(y_k-x_k)(y_k-w_k)}(1 + \frac{f(y_k)}{f(x_k)}), \end{cases} \tag{5}$$

where $w_k = x_k + \gamma f(x_k)$,$0 \neq \lambda, q$ and $\gamma \in \mathbb{R}$, and $f[x, y] = \frac{f(x)-f(y)}{x-y}$ stands for the divided difference of the first order. This is an optimal method without memory. In order words, it uses three function evaluations per iteration, and has optimal convergence order 4. It is possible to adapt the method (5) in some ways that it remains optimal in the sence of Kung and Traub conjecture [23] as follows:

$$\begin{cases} w_k = x_k + \gamma_k f(x_k),\ y_k = x_k - \frac{f(x_k)}{f[x_k,w_k]+q_k f(w_k)},\ k = 0, 1, 2, \cdots, \\ x_{k+1} = y_k - \frac{f(y_k)}{f[w_k,y_k]+q_k f(w_k)+\lambda_k(y_k-x_k)(y_k-w_k)}(1 + \frac{f(y_k)}{f(x_k)}), \end{cases} \tag{6}$$

where γ, λ and q are arbitrary nonzero real parameters. The next theorem states the error equation of the method (6).

Theorem 2 *Let $I \subseteq \mathbb{R}$ be an open interval, $f : I \to \mathbb{R}$ be a differentiable function, and has a simple zero, say α. If x_0 is an initial guess to α, then the error equation of the method (6) is given by*

$$e_{k+1} = \frac{1}{f'(\alpha)}((1+\gamma f'(\alpha))^2(q+c_2)(\lambda + f'(\alpha)q^2(1+\gamma f'(\alpha)) + f'(\alpha)c_2$$
$$(2q(2+\gamma f'(\alpha)) + (3+\gamma f'(\alpha))c_2) - f'(\alpha)c_3)e_k^4 + O(e_k^5),$$

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

Proof : We use the self-explained mathematical approach to avoid the tedious and humdrum algebraic manipulation. First, we define the Taylor's series of $f(x)$ as follows:

$$In[1] : f[e\_] = fla(e + c_2e^2 + c_3e^3 + c_4e^4);$$

where $e = x - \alpha$, $fla = f'(\alpha)$. Note that since α is a simple zero of $f(x)$, then $f'(\alpha) \neq 0$, $f(\alpha) = 0$. We define

$$In[2] : f[x\_, y\_] = \frac{f[x] - f[y]}{x - y};$$

$$In[3] : ew = e + \gamma f[e];$$

$$In[4] : ey = e - Series[\frac{f[e]}{f[e, ew] + qf[ew]}, \{e, 0, 4\}];$$

$$In[5] : e_{k+1} = ey - Series[\frac{f[ey]}{f[ey, ew] + qf[ew] + \lambda(ey - e)(ey - ew)}$$
$$(1 + \frac{f[ey]}{f[e]}),\ \{e, 0, 4\}]//FullSimplify$$

$$Out[5] : e_{k+1} = \frac{1}{f'(\alpha)}((1+\gamma f'(\alpha))^2(q+c_2)(\lambda + f'(\alpha)q^2(1+\gamma f'(\alpha)) + f'(\alpha)c_2$$
$$(2q(2+\gamma f'(\alpha)) + (3+\gamma f'(\alpha))c_2) - f'(\alpha)c_3)e_k^4 + O(e_k^5).$$

This completes the proof. □

2.3 Acceleration of the modified Lotfi and Assari's method

This section concerns with modifying Lotfi and Assari's method (LAM) [27], so that it could be considered for the proposed scheme in the next section. Let recall the mentioned method:

$$\begin{cases} w_k = x_k + \gamma f(x_k),\ y_k = x_k - \frac{f(x_k)}{f[x_k, w_k] + qf(w_k)},\ k = 0, 1, 2, \cdots, \\ z_k = y_k - \frac{f(y_k)}{f[y_k, x_k] + f[w_k, x_k, y_k](y_k - x_k) + \lambda(y_k - x_k)(y_k - w_k)}, \\ x_{k+1} = z_k - \frac{f(z_k)}{f[x_k, z_k] + (f[w_k, x_k, y_k] - f[w_k, x_k, z_k] - f[y_k, x_k, z_k])(x_k - z_k) + \beta(z_k - y_k)(z_k - x_k)(z_k - w_k)}. \end{cases} \tag{7}$$

This optimal method without memory use four function evaluations per iteration, and has convergence order 8. To transform Eq. (7) in a method with memory, with four accelerators, we consider the following modification of (7) [27]:

$$\begin{cases} w_k = x_k + \gamma_k f(x_k),\ k = 1,2,3,\cdots, \\ y_k = x_k - \frac{f(x_k)}{f[x_k,w_k]+q_k f(w_k)},\ k = 0,1,2,\cdots, \\ z_k = y_k - \frac{f(y_k)}{f[y_k,x_k]+f[w_k,x_k,y_k](y_k-x_k)+\lambda_k(y_k-x_k)(y_k-w_k)}, \\ x_{k+1} = z_k - \frac{f(z_k)}{f[x_k,z_k]+(f[w_k,x_k,y_k]-f[w_k,x_k,z_k]-f[y_k,x_k,z_k])(x_k-z_k)+\beta_k(z_k-y_k)(z_k-x_k)(z_k-w_k)}, \end{cases} \tag{8}$$

where γ_k, β_k, λ_k and q_k are nonzero arbitrary parameters. We give the following convergence theorem for the proposed method (8) as follows:

Theorem 3 *Let $I \subseteq \mathbb{R}$ be an open interval, $f : I \to \mathbb{R}$ be a differentiable function, and has a simple zero, say α. If x_0 is an initial guess to α, then the error equation of the method (8) is given by*

$$\begin{aligned} e_{k+1} =&(1+\gamma f'(\alpha))^4(q+c_2)^2(\lambda+f'(\alpha)c_2(q+c_2)-f'(\alpha)c_3)(-\beta+c_2 \\ &(\lambda+f'(\alpha)c_2(q+c_2)-f'(\alpha)c_3)+f'(\alpha)c_4)f'(\alpha)^{-2}e_k^8+O(e_k^9). \end{aligned} \tag{9}$$

Proof : First, we define the Taylor series of $f(x)$ as follows:

$$In[1] : f[e\_] = fla(e + c_2e^2 + \cdots + c_8e^8),$$

where $e = x - \alpha$, $fla = f'(\alpha)$. Note that since α is a simple zero of $f(x)$, the $f'(\alpha) \neq 0, f(\alpha) = 0$. We define

$$In[2] : f[x\_, y\_] = \frac{f[x] - f[y]}{x - y};$$

$$In[3] : f[x\_, y\_, z\_] = \frac{f[x, y] - f[y, z]}{x - z};$$

$$In[4] : f[x\_, y\_, z\_, t\_] = \frac{f[x, y, z] - f[y, z, t]}{x - t};$$

$$In[5] : ew = e + \gamma f[e];$$

$$In[6] : ey = e - Series[\frac{f[e]}{f[e, ew] + qf[ew]}, \{e, 0, 8\}];$$

$$In[7] : ez = ey - Series[\frac{f[ey]}{f[ey, e] + f[ey, e, ew](ey - e) + \lambda(ey - e)(ey - ew)}, \{e, 0, 8\}];$$

$$\begin{aligned} In[8] : e_{k+1} =&ez - Series[f[ez]/(f[ez, ey] + f[ez, ey, e](ez - ey) + f[ez, ey, e, ew](ez - ey) \\ &(ez - e) + \beta(ez - ey)(ez - e)(ez - ew)), \{e, 0, 8\}]//FullSimplify \end{aligned}$$

$$\begin{aligned} Out[8] : e_{k+1} = &((1 + \gamma f'(\alpha))^4(q + c_2)^2(\lambda + f'(\alpha)c_2(q + c_2) - f'(\alpha)c_3)(-\beta + c_2(\lambda + f'(\alpha)c_2(q + c_2) \\ &- f'(\alpha)c_3) + f'(\alpha)c_4)f'(\alpha)^{-2})e_k^8 + O(e_k^9) \end{aligned}$$

And thus proof is completed. □

3 Development the recursive adaptive method with memory

This section deals with the main contribution of this work. In other words, it is attempted to introduce a recursive adaptive method with memory so that it has the highest possible efficiency index as proposed to methods with memory in the literature. It is worth mentioning that some special cases of this new method covers the existing methods.

3.1 One step adaptive method

This section concerns with extracting the novel method with memory from (3) by using two self-accelerating parameters. Theorem (1) states that modified method (3) has order of convergence 2 if $\gamma \neq \frac{-1}{f'(\alpha)}$ and $q \neq -c_2$. Now, we pose some questions: Is it possible to increase the order of convergence of this method? If so, how can it be done, and what is the new convergence order? For answering these questions, we look at the error equation (4). As can be seen that if we set $\gamma = \frac{-1}{f'(\alpha)}$ and $q = -c_2 = \frac{f''(\alpha)}{-2f'(\alpha)}$, then at least the coefficient of e_k^2 disappears. However, since α is not determined and consequently, $f'(\alpha)$ and $f''(\alpha)$ cannot be computed. On the other hand, we can approximate α using available data and therefore improve order of convergence. Following the same idea in the methods with memory, this issue can be resaved. However, we are going to do it in a more efficient way, say recursive adaptively. Let us describe it a little more. If we use information from the current and only the last iteration, we come up with the method with memory introduced in [27, 28]. Also, note that we have considered the best approximations. Hence, to this end, the following approximates are applied

$$\gamma_k = \frac{-1}{N_2'(x_k)} \approx \frac{-1}{f'(\alpha)}, q_k = \frac{N_3''(w_k)}{-2N_3'(w_k)} \approx -\frac{f''(\alpha)}{2f'(\alpha)}, \tag{10}$$

where $k = 1, 2, \cdots$, the $N_2'(x_k), N_3'(w_k)$ are Newton's interpolating polynomials of two and third degree, set through three and four best available approximations (nodes) (x_k, x_{k-1}, w_{k-1}) and $(w_k, x_k, x_{k-1}, w_{k-1})$, respectively. It should be noted that if one uses lower Newton's interpolation, lower accelerators are obtained.
Replacing the fixed parameters q and γ in the iterative formula (4) by the varying γ_k and q_k calculated by (4), we propose the following new methods with memory, x_0, q_0, γ_0 are given, and $w_0 = x_0 + \gamma_0 f(x_0)$

$$\begin{cases} \gamma_k = \frac{-1}{N_{2k}'(x_k)},\ q_k = \frac{N_{2k+1}''(w_k)}{-2N_{2k+1}'(w_k)},\ k = 1, 2, \cdots, \\ w_k = x_k + \gamma_k f(x_k),\ x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k] + q_k f[w_k]},\ k = 0, 1, 2, \cdots. \end{cases} \tag{11}$$

Here, we answer the second question regarding order of convergence of the method with memory (11). In what follows, we discuss the general convergence analysis of the recursive adaptive method with memory (11). It should be noted that the convergence order varies as the iteration go ahead. First, we need the following lemma.

Lemma 1 *If* $\gamma_k = \frac{-1}{N'_{2k}(x_k)}$ *and* $q_k = \frac{N''_{2k+1}(w_k)}{-2N'_{2k+1}(w_k)}$, *then*

$$(1 + \gamma_k f'(\alpha)) \sim \prod_{s=0}^{k-1} e_s e_{s,w}, \tag{12}$$

$$(q_k + c_2) \sim \prod_{s=0}^{k-1} e_s e_{s,w}, \tag{13}$$

where $e_s = x_s - \alpha, e_{s,w} = w_s - \alpha$.

Proof : The proof is similar to the Lemma 4 mentioned in [46].
The following result determines the order of convergence of the one-point iterative method with memory (11).

Theorem 4 *If an initial estimation* x_0 *is close enough to a simple root* α *of* $f(x) = 0$, *and* γ_0 *and* q_0 *are uniformly bounded above, being* f *a real sufficiently differentiable function, then the R-order of convergence of the one-point method adaptive with memory (11) is obtained from the following system of nonlinear equations.*

$$\begin{cases} r^k p - (1+p)(1 + r + r^2 + r^3 + \ldots + r^{k-1}) - r^k = 0, \\ r^{k+1} - 2(1+p)(1 + r + r^2 + r^3 + \ldots + r^{k-1}) - 2r^k = 0, \end{cases} \tag{14}$$

where r *and* p *are the convergence order of the sequences* $\{x_k\}$ *and* $\{w_k\}$, *respectively. Also,* k *indicates the number of iterations.*
Proof : Let $\{x_k\}$ *and* $\{w_k\}$ *be convergent with orders* r *and* p, *respectively. Then:*

$$\begin{cases} e_{k+1} \sim e_k^r \sim e_{k-1}^{r^2} \sim \ldots \sim e_0^{r^{k+1}}, \\ e_{k,w} \sim e_k^p \sim e_{k-1}^{rp} \sim \ldots \sim e_0^{pr^k}, \end{cases} \tag{15}$$

where $e_k = x_k - \alpha$ *and* $e_{k,w} = w_k - \alpha$. *Now, by Lemma (1) and Eq (15), we obtain*

$$\begin{aligned} (1 + \gamma_k f'(\alpha)) \sim \prod_{s=0}^{k-1} e_s e_{s,w} &= (e_0 e_{0,w}) \ldots (e_{k-1} e_{k-1,w}) \\ &= (e_0 e_0^p)(e_0^r e_0^{rp}) \ldots (e_0^{r^{k-1}} e_0^{r^{k-1}p}) \\ &= e_0^{(1+p)+(1+p)r+\ldots+(1+p)r^{k-1}} \\ &= e_0^{(1+p)(1+r+\ldots+r^{k-1})}. \end{aligned} \tag{16}$$

Similarly, we get

$$(q_k + c_2) \sim e_0^{(1+p)(1+r+\ldots+r^{k-1})}. \tag{17}$$

By considering the errors of w_k *and* x_{k+1} *in Eq. (15), and Eqs. (16)-(17). We conclude:*

$$e_{k,w} \sim (1 + \gamma_k f'(\alpha)) e_k \sim e_0^{(1+p)(1+r+\ldots+r^{k-1})} e_0^{r^k}, \tag{18}$$

$$e_{k+1} \sim (1 + \gamma_k f'(\alpha))(q_k + c_2) e_k^2 \sim e_0^{((1+p)(1+r+\ldots+r^{k-1}))^2} e_0^{2r^k}. \tag{19}$$

To obtain the desire result, it is enough to match the right-hand-side of the Eqs(15),(18), and (19) :

$$\begin{cases} r^k p-(1+p)(1+r+r^2+r^3+\ldots+r^{k-1})-r^k=0,\ k=1,2,\ldots, \\ r^{k+1}-2(1+p)(1+r+r^2+r^3+\ldots+r^{k-1})-2r^k=0. \end{cases}$$

This completes the proof of the Theorem. □

Remark 1 *For $k=1$, the order of convergence of the method with memory (11) can be computed from the following of system of equations*

$$\begin{cases} rp-(1+p)-r=0, \\ r^2-2(1+p)-2r=0. \end{cases} \tag{20}$$

This system of equations has the solution $p=\frac{1}{4}(3+\sqrt{17})\simeq 1.78078$, and $r=\frac{1}{2}(3+\sqrt{17})\simeq 3.56155$. This special case give the given result by Dzunic [14] and denoted by DM. If $k=2$, the system of equations(14) becomes:

$$\begin{cases} r^2p-(1+p+rp+r+r^2)=0, \\ r^3-2(1+p+rp+r+r^2)=0. \end{cases} \tag{21}$$

This system of equations has the solution: $p\simeq 1.95029$ and $r\simeq 3.90057$.
Also, Positive solution of the system(14) for $k=3$, is given by $p\simeq 1.98804$ and $r\simeq 3.97609$. And, positive solution of the system(14) for $k=4$, is given by(has been shown by TAM4) $p\simeq 1.99705$ and $r\simeq 3.9941$.

As can be seen the order of convergence is very close to 4, so its efficiency index is very close to 2. This efficiency is astonishingly remarkable.

3.2 Two-steps adaptive method

Let us look at the error equation of the modified method (6). It is clear that there are some possibilities to vanish the coefficient of e_k^4. For example, if $1+\gamma f'(\alpha)=0, q+c_2=0$, or $(\lambda+f'(\alpha)q^2(1+\gamma f'(\alpha))+f'(\alpha)c_2(2q(2+\gamma f'(\alpha))+(3+\gamma f'(\alpha))c_2)-f'(\alpha)c_3)=0$, then the coeffcient of e_k^4 vanishes at once. To get the best result, we suggest that all these relations hold simultaneously. We note that this can happen theoretically. To be more precise, it can be seen that these relations lead to $\gamma=\frac{-1}{f'(\alpha)}, q=-c_2=-\frac{f''(\alpha)}{2f'(\alpha)}$, and $\lambda=f'(\alpha)c_3=\frac{f'''(\alpha)}{6}$. Since α is not at hand, it is impossible to compute $f'(\alpha), f''(\alpha)$, and $f'''(\alpha)$. Even worse, if we assume that α is known, computing $f'(\alpha)$, $f''(\alpha)$, and $f'''(\alpha)$ is not suggested since it increases these function evaluations, and therefore, it spoils that optimality of the method (6). Following the same idea in the methods with memory, this issue can be resaved. However, we are going to do it in a more efficient way, say recursive adaptively. Note that we have considered the best approximations. Hence

$$\begin{cases} \gamma_k=-\frac{1}{N_3'(x_k)}\simeq\frac{-1}{f'(\alpha)}, \\ q_k=-\frac{N_4''(w_k)}{2N_4'(w_k)}\simeq-\frac{f''(\alpha)}{2f'(\alpha)}, \\ \lambda_k=\frac{N_5'''(y_k)}{6}\simeq f'(\alpha)c_3=\frac{f'''(\alpha)}{6}, \end{cases} \tag{22}$$

where $N_3'(x_k)$, $N_4''(w_k)$ and $N_5'''(y_k)$ are Newton's interpolation polynomials go through the nodes $\{x_k, x_{k-1}, w_{k-1}, y_{k-1}\}$, $\{w_k, x_k, x_{k-1}, w_{k-1}, y_{k-1}\}$, and $\{y_k, w_k, x_k, x_{k-1}, w_{k-1}, y_{k-1}\}$, respectively. This situation has been well studied in [10,11,27,28,39]. Such methods are not adaptive. To construct a recursive adaptive method with memory, we use the information not only in the current and its previous iterations, but also in all the previous iterations, i.e., from the beginning to the current iteration. Thus, as iterations proceed, the degree of interpolation polynomials increases, and the best updated approximations for computing the self-accelerator γ_k, q_k, and λ_k are obtained. Let x_0, γ_0, q_0, and λ_0 be given suitably. Then,

$$\begin{cases} \gamma_k = -\frac{1}{N'_{3k}(x_k)}, q_k = -\frac{N''_{3k+1}(w_k)}{2N'_{3k+1}(w_k)}, \lambda_k = \frac{N'''_{3k+2}(y_k)}{6}, \ k = 1, 2, 3, \ldots, \\ y_k = x_k - \frac{f(x_k)}{f[x_k, w_k] + q_k f(w_k)}, \ k = 0, 1, 2, \cdots, \\ x_{k+1} = y_k - \frac{f(y_k)}{f[w_k, y_k] + q_k f(w_k) + \lambda_k (y_k - x_k)(y_k - w_k)} (1 + \frac{f(y_k)}{f(x_k)}). \end{cases} \tag{23}$$

In what follows, we discuss the general convergence analysis of the recursive adaptive method with memory (23). It should be noted that the convergence order varies as the iteration go ahead. First, we need the following lemma.

Lemma 2 *If* $\gamma_k = -\frac{1}{N'_{3k}(x_k)}, q_k = -\frac{N''_{3k+1}(w_k)}{2N'_{3k+1}(w_k)}$, *and* $\lambda_k = \frac{N'''_{3k+2}(y_k)}{6}$, *then :*

$$(1 + \gamma_k f'(\alpha)) \sim \prod_{s=0}^{k-1} e_s e_{s,w} e_{s,y}, \tag{24}$$

$$(c_2 + q_k) \sim \prod_{s=0}^{k-1} e_s e_{s,w} e_{s,y}, \tag{25}$$

$$(\lambda_k + f'(\alpha) q^2 (1 + \gamma_k f'(\alpha)) + f'(\alpha) c_2 (2q(2 + \gamma_k f'(\alpha)) + (3 + \gamma_k f'(\alpha)) c_2) - f'(\alpha) c_3) \sim \prod_{s=0}^{k-1} e_s e_{s,w} e_{s,y}, \tag{26}$$

where $e_s = x_s - \alpha, e_{s,w} = w_s - \alpha, e_{s,y} = y_s - \alpha$.

Proof : The proof is similar to Lemmas 2.1 and 2.2 in [46].

Theorem 5 *Let x_0 be a suitable initial guess to the simple root α of $f(x) = 0$. Also, suppose the initial values γ_0, q_0, and λ_0 are chosen appropriately. Then the R-order of the recursive adaptive method with memory (23) can be obtained from the following system of nonlinear equations:*

$$\begin{cases} r^k p_1 - (1 + p_1 + p_2)(1 + r + r^2 + r^3 + \ldots + r^{k-1}) - r^k = 0, \\ r^k p_2 - 2(1 + p_1 + p_2)(1 + r + r^2 + r^3 + \ldots + r^{k-1}) - 2r^k = 0, \\ r^{k+1} - 4(1 + p_1 + p_2)(1 + r + r^2 + r^3 + \ldots + r^{k-1}) - 4r^k = 0, \end{cases} \tag{27}$$

where r, p_1 and p_2 are the order of convergence of the sequences $\{x_k\}$, $\{w_k\}$, and $\{y_k\}$, respectively. Also, k, indicates the number of iterations.
Proof : Let $\{x_k\}$, $\{w_k\}$, and $\{y_k\}$, be convergent with orders r, p_1, and p_2, respectively. Then:

$$\begin{cases} e_{k+1} \sim e_k^r \sim e_{k-1}^{r^2} \sim \ldots \sim e_0^{r^{k+1}}, \\ e_{k,w} \sim e_k^{p_1} \sim e_{k-1}^{p_1 r} \sim \ldots \sim e_0^{p_1 r^k}, \\ e_{k,y} \sim e_k^{p_2} \sim e_{k-1}^{p_2 r} \sim \ldots \sim e_0^{p_2 r^k}, \end{cases} \tag{28}$$

where $e_k = x_k - \alpha, e_{k,w} = w_k - \alpha$ *and* $e_{k,y} = y_k - \alpha$. *Now, by Lemma (2) and Eq (27), we obtain*

$$\begin{aligned}(1+\gamma_k f'(\alpha)) \sim \prod_{s=0}^{k-1} e_s e_{s,w} e_{s,y} &= (e_0 e_{0,w} e_{0,y}) \dots (e_{k-1} e_{k-1,w} e_{k-1,y}) \\ &= (e_0 e_0^{p_1} e_0^{p_2})(e_0^r e_0^{p_1 r} e_0^{p_2 r}) \dots (e_0^{r^{k-1}} e_0^{r^{k-1} p_1} e_0^{r^{k-1} p_2}) \\ &= e_0^{(1+p_1+p_2)+(1+p_1+p_2)r+\dots+(1+p_1+p_2)r^{k-1}} \\ &= e_0^{(1+p_1+p_2)(1+r+\dots+r^{k-1})}. \end{aligned} \tag{29}$$

Similarly, we get :

$$(q_k + c_2) \sim e_0^{(1+p_1+p_2)(1+r+\dots+r^{k-1})}, \tag{30}$$

and

$$\begin{aligned}&(\lambda_k + f'(\alpha) q^2 (1+\gamma_k f'(\alpha)) + f'(\alpha) c_2 (2q(2+\gamma_k f'(\alpha)) + (3+\gamma_k f'(\alpha)) c_2) - f'(\alpha) c_3) \\ &\sim e_0^{(1+p_1+p_2)(1+r+\dots+r^{k-1})}. \end{aligned} \tag{31}$$

By considering the errors of w_k, y_k, *and* x_{k+1} *in Eq. (27), and Eqs. (29)-(31), we conclude:*

$$e_{k,w} \sim (1+\gamma_k f'(\alpha)) e_k \sim e_0^{(1+p_1+p_2)(1+r+\dots+r^{k-1})} e_0^{r^k}, \tag{32}$$

$$e_{k,y} \sim (1+\gamma_k f'(\alpha))(q_k + c_2) e_k^2 \sim e_0^{((1+p_1+p_2)(1+r+\dots+r^{k-1}))^2} e_0^{2r^k}, \tag{33}$$

$$\begin{aligned}&e_{k+1} \sim (1+\gamma_k f'(\alpha))^2 (q_k + c_2)(\lambda_k + f'(\alpha) q^2 (1+\gamma_k f'(\alpha)) + f'(\alpha) c_2 (2q(2+\gamma_k f'(\alpha)) + \\ &(3+\gamma_k f'(\alpha)) c_2) - f'(\alpha) c_3) e_k^4 \sim e_0^{((1+p_1+p_2)(1+r+\dots+r^{k-1}))^4} e_0^{4r^k}. \end{aligned} \tag{34}$$

To obtain the desired result, it is enough to match the right-hand-side of the Eqs. (27), (32), (33), and (34). Then

$$\begin{cases} r^k p_1 - (1+p_1+p_2)(1+r+r^2+r^3+\dots+r^{k-1}) - r^k = 0, \ k = 1, 2, \dots, \\ r^k p_2 - 2(1+p_1+p_2)(1+r+r^2+r^3+\dots+r^{k-1}) - 2r^k = 0, \\ r^{k+1} - 4(1+p_1+p_2)(1+r+r^2+r^3+\dots+r^{k-1}) - 4r^k = 0. \end{cases}$$

This completes the proof of the Theorem. □

Remark 2 *Positive solution of system (23), ($k = 1$), is specified through (It been shown by TAM7):* $p_1 = \frac{1}{8}(7+\sqrt{65}) \simeq 1.88, p_2 = \frac{1}{4}(7+\sqrt{65}) \simeq 3.76$ *and* $r = \frac{1}{2}(7+\sqrt{65}) \simeq 7.53$. *Therefore, the convergence order of the new method with memory (23) is at least 7.5311.*
And, if $k = 2$, *we obtain the order of convergence:* $p_1 \simeq 1.98612, p_2 \simeq 3.97225$ *and* $r \simeq 7.94449$. *Also,* $k = 3$, *the system of equations(27) has the solution :* $p_1 \simeq 1.99829, p_2 \simeq 3.99657$ *and* $r \simeq 7.99315$.
Likewise, for $k = 4$, *we obtain the order of convergence:*

$$p_1 \simeq 1.99979, \ p_2 \simeq 3.99957 \ \text{and}\, r \simeq 7.99915 \tag{35}$$

(been shown by TAM8). In this case the efficiency index is $7.99915^{\frac{1}{3}} = 1.99993 \cong 2$ *which shows that our developed method compets all the existing methods with memory.*

3.3 Three-steps adaptive method

This section introduced a new efficient adaptive method with memory. We continue as before, and develope a three steps method with memory with the best efficiency index. Indeed, we achieve the efficiency index 2. Also, note that we have considered the best approximations (8). Hence

$$\begin{cases} \gamma_k = -\frac{1}{N_4'(x_k)} \simeq \frac{-1}{f'(\alpha)}, \\ q_k = -\frac{N_5''(w_k)}{2N_5'(w_k)} \simeq -\frac{f''(\alpha)}{2f'(\alpha)}, \\ \lambda_k = \frac{N_6'''(y_k)}{6} \simeq f'(\alpha)c_3 = \frac{f'''(\alpha)}{6}, \\ \beta_k = \frac{N_7'''(z_k)}{24} \simeq f'(\alpha)c_4 = \frac{f''''(\alpha)}{24}. \end{cases} \tag{36}$$

where $N_4'(x_k), N_5''(w_k), N_6'''(y_k)$ and $N_7''''(z_k)$ are Newton's interpolation polynomials go through the nodes $\{x_k, x_{k-1}, w_{k-1}, y_{k-1}, z_{k-1}\}, \{w_k, x_k, x_{k-1}, w_{k-1}, y_{k-1}, z_{k-1}\}$, $\{y_k, w_k, x_k, x_{k-1}, w_{k-1}, y_{k-1}, z_{k-1}\}$, and $\{z_k y_k, w_k, x_k, x_{k-1}, w_{k-1}, y_{k-1}, z_{k-1}\}$, respectively. The degree of interpolation polynomials increases, and the best updated approximations for computing the self-accelerator γ_k, q_k, λ_k and β_k are obtained. Now, we can present the first three-step adaptive memory method as follows:(Let $w_0 = x_0 + \gamma_0 f(x_0), x_0, \gamma_0, q_0, \lambda_0$ and β_0 be given suitably.)

$$\begin{cases} \gamma_k = -\frac{1}{N_{4k}'(x_k)}, q_k = -\frac{N_{4k+1}''(w_k)}{2N_{4k+1}'(w_k)}, \lambda_k = \frac{N_{4k+2}'''(y_k)}{6}, \beta_k = \frac{N_{4k+3}''''(z_k)}{24}, \ k = 1, 2, 3, \ldots, \\ w_k = x_k + \gamma_k f(x_k), \ y_k = x_k - \frac{f(x_k)}{f[x_k, w_k] + q_k f(w_k)}, \ k = 0, 1, 2, \cdots, \\ z_k = y_k - \frac{f(y_k)}{f[y_k, x_k] + f[w_k, x_k, y_k](y_k - x_k) + \lambda_k (y_k - x_k)(y_k - w_k)}, \\ x_{k+1} = z_k - \frac{f(z_k)}{f[x_k, z_k] + (f[w_k, x_k, y_k] - f[w_k, x_k, z_k] - f[y_k, x_k, z_k])(x_k - z_k) + \beta_k (z_k - y_k)(z_k - x_k)(z_k - w_k)}. \end{cases} \tag{37}$$

In what follows, we discuss the general convergence analysis of the recursive adaptive method with memory (37). It should be noted that the convergence order varies as the iteration go ahead. We need the following lemma.

Lemma 1 *If* $\gamma_k = -\frac{1}{N_{4k}'(x_k)}$, $q_k = -\frac{N_{4k+1}''(w_k)}{2N_{4k+1}'(w_k)}$, $\lambda_k = \frac{N_{4k+2}'''(y_k)}{6}$, *and* $\beta_k = \frac{N_{4k+3}''''(z_k)}{24}$, *then :*

$$(1 + \gamma_k f'(\alpha)) \sim \prod_{s=0}^{k-1} e_s e_{s,w} e_{s,y} e_{s,z}, \tag{38}$$

$$(c_2 + q_k) \sim \prod_{s=0}^{k-1} e_s e_{s,w} e_{s,y} e_{s,z}, \tag{39}$$

$$(\lambda_k + f'(\alpha)c_2(q_k + c_2) - f'(\alpha)c_3) \sim \prod_{s=0}^{k-1} e_s e_{s,w} e_{s,y} e_{s,z}, \tag{40}$$

$$(\beta_k + c_2(\lambda_k + f'(\alpha)c_2(q_k + c_2) - f'(\alpha)c_3) - f'(\alpha)c_4) \sim \prod_{s=0}^{k-1} e_s e_{s,w} e_{s,y} e_{s,z}, \tag{41}$$

where $e_s = x_s - \alpha, e_{s,w} = w_s - \alpha, e_{s,y} = y_s - \alpha, e_{s,z} = z_s - \alpha$.
Proof : The proof is similar to Lemma 1 in [46].

Theorem 6 *Let x_0 be a suitable initial guess to the simple root α of $f(x) = 0$. Also, suppose the initial values γ_0, q_0, λ_0, and β_0 are chosen appropriately.*
Then the R-order of the recursive adaptive method with memory (37) can be obtained from the following system of nonlinear equations:

$$\begin{cases} r^k p_1 - (1+p_1+p_2+p_3)(1+r+r^2+r^3+\ldots+r^{k-1}) - r^k = 0, \\ r^k p_2 - 2(1+p_1+p_2+p_3)(1+r+r^2+r^3+\ldots+r^{k-1}) - 2r^k = 0, \\ r^k p_3 - 4(1+p_1+p_2+p_3)(1+r+r^2+r^3+\ldots+r^{k-1}) - 4r^k = 0, \\ r^{k+1} - 8(1+p_1+p_2+p_3)(1+r+r^2+r^3+\ldots+r^{k-1}) - 8r^k = 0, \end{cases} \tag{42}$$

where r, p_1, p_2, and p_3 are the order of convergence of the sequences $\{x_k\}$, $\{w_k\}$, $\{y_k\}$, and $\{z_k\}$, respectively. Also, k, indicates the number of iterations.

Proof : Let $\{x_k\}, \{w_k\}, \{y_k\}$ and $\{z_k\}$ be convergent with orders r, p_1, p_2 and p_3 respectively. Then:

$$\begin{cases} e_{k+1} \sim e_k^r \sim e_{k-1}^{r^2} \sim \ldots \sim e_0^{r^{k+1}}, \\ e_{k,w} \sim e_k^{p_1} \sim e_{k-1}^{rp_1} \sim \ldots \sim e_0^{p_1 r^k}, \\ e_{k,y} \sim e_k^{p_2} \sim e_{k-1}^{rp_2} \sim \ldots \sim e_0^{p_2 r^k}, \\ e_{k,z} \sim e_k^{p_3} \sim e_{k-1}^{rp_3} \sim \ldots \sim e_0^{p_3 r^k}, \end{cases} \tag{43}$$

where $e_k = x_k - \alpha$, $e_{k,w} = w_k - \alpha$, $e_{k,y} = y_k - \alpha$ and $e_{k,z} = z_k - \alpha$. Now, by Lemma (1) and Eq. (43), we obtain

$$\begin{aligned} (1+\gamma_k f'(\alpha)) &\sim \prod_{s=0}^{k-1} e_s e_{s,w} e_{s,y} e_{s,z} = (e_0 e_{0,w} e_{0,y} e_{0,z}) \ldots (e_{k-1} e_{k-1,w} e_{k-1,y} e_{k-1,z}) \\ &= (e_0 e_0^{p_1} e_0^{p_2} e_0^{p_3})(e_0^r e_0^{p_1 r} e_0^{p_2 r} e_0^{p_3 r}) \ldots (e_0^{r^{k-1}} e_0^{r^{k-1} p_1} e_0^{r^{k-1} p_2} e_0^{r^{k-1} p_3}) \\ &= e_0^{(1+p_1+p_2+p_3)+(1+p_1+p_2+p_3)r+\ldots+(1+p_1+p_2+p_3)r^{k-1}} \\ &= e_0^{(1+p_1+p_2+p_3)(1+r+\ldots+r^{k-1})}. \end{aligned} \tag{44}$$

Similarly, we get

$$(q_k + c_2) \sim e_0^{(1+p_1+p_2+p_3)(1+r+\ldots+r^{k-1})}, \tag{45}$$

and

$$(\lambda_k + f'(\alpha)c_2(q_k + c_2) - f'(\alpha)c_3) \sim e_0^{(1+p_1+p_2+p_3)(1+r+\ldots+r^{k-1})}, \tag{46}$$

$$\begin{aligned} &(\beta_k + c_2(\lambda_k + f'(\alpha)c_2(q_k + c_2) - f'(\alpha)c_3) - f'(\alpha)c_4) \\ &\qquad \sim e_0^{(1+p_1+p_2+p_3)(1+r+\ldots+r^{k-1})}. \end{aligned} \tag{47}$$

By considering the errors of w_k, y_k, z_k and x_{k+1} in Eq. (37) and Eqs. (44)-(47), we conclude:

$$e_{k,w} \sim (1+\gamma_k f'(\alpha)) e_k \sim e_0^{(1+p_1+p_2+p_3)(1+r+\ldots+r^{k-1})} e_0^{r^k}, \tag{48}$$

$$e_{k,y} \sim (1+\gamma_k f'(\alpha))(q_k + c_2) e_k^2 \sim e_0^{((1+p_1+p_2+p_3)(1+r+\ldots+r^{k-1}))^2} e_0^{2r^k}, \tag{49}$$

$$e_{k,z} \sim (1+\gamma_k f'(\alpha))^2(q_k+c_2)(\lambda_k+f'(\alpha)c_2(q_k+c_2)-f'(\alpha)c_3)e_k^4$$
$$\sim e_0^{((1+r_1+r_2+r_3)(1+r+\ldots+r^{k-1}))^4} e_0^{4r^k}, \quad (50)$$

$$e_{k+1} \sim (1+\gamma_k f'(\alpha))^4(q_k+c_2)^2(\lambda_k+f'(\alpha)c_2(q_k+c_2)-f'(\alpha)c_3)$$
$$(\beta_k+c_2(\lambda_k+f'(\alpha)c_2(q_k+c_2)-f'(\alpha)c_3)-f'(\alpha)c_4)e_k^8$$
$$\sim e_0^{((1+p_1+p_2+p_3)(1+r+\ldots+r^{k-1}))^8} e_0^{8r^k}. \quad (51)$$

To obtain the desired result, it is enough to match the right-hand-side of the Eqs. (43), (48), (49), (50) and (51). Then:

$$\begin{cases} r^k p_1 - (1+p_1+p_2+p_3)(1+r+r^2+r^3+\ldots+r^{k-1}) - r^k = 0, \; k \in \mathbb{N} \\ r^k p_2 - 2(1+p_1+p_2+p_3)(1+r+r^2+r^3+\ldots+r^{k-1}) - 2r^k = 0, \\ r^k p_3 - 4(1+p_1+p_2+p_3)(1+r+r^2+r^3+\ldots+r^{k-1}) - 4r^k = 0, \\ r^{k+1} - 8(1+p_1+p_2+p_3)(1+r+r^2+r^3+\ldots+r^{k-1}) - 8r^k = 0. \end{cases}$$

This completes the proof of the Theorem. □

Remark 3 *For $k = 1$, we use the information from the current and the one previous steps. In this case, the order of convergence of the with memory method (37) can be computed from the following system*

$$\begin{cases} rp_1 - (1+p_1+p_2+p_3) - r = 0, \\ rp_2 - 2(1+p_1+p_2+p_3) - 2r = 0, \\ rp_3 - 4(1+p_1+p_2+p_3) - 4r = 0, \\ r^2 - 8(1+p_1+p_2+p_3) - 8r = 0. \end{cases} \quad (52)$$

After solving these equations, we have: $p_1 = \frac{1}{16}(15+\sqrt{257}) \simeq 1.93945$, $p_2 = \frac{1}{8}(15+\sqrt{257}) \simeq 3.8789, p_3 = \frac{1}{4}(15+\sqrt{257}) \simeq 7.7578$ and $r = \frac{1}{2}(15+\sqrt{257}) \simeq 15.5156$. This special case determines the given result by Lotfi-Assari [27].

If $k = 2$, we obtain the order of convergence: $r_1 \simeq 1.99632$, $r_2 \simeq 3.99265$, $r_3 \simeq 7.9853$ and $r \simeq 15.9706$.

And, if $k = 3$, The positive real solution (42) is: $p_1 \simeq 1.99977$, $p_2 \simeq 3.99954$, $p_3 \simeq 7.99908$ and $r \simeq 15.9982$.

Also, for $k = 4$, the system (42) has the solution :(shown by TAM16)

$$p_1 = 2, \; p_2 = 4, \; p_3 = 8, \; \text{and} \; r = 16. \quad (53)$$

This shows that the R-order of convergence for (37) is 16.

Remark 4 *As can be easily seen that the improvement the order of convergence from 2, 4 and 8 to 4, 8 and 16 (100% of an improvement) is attained without any additional functional evaluations, which points to very high computational efficiency of the proposed methods. Therefore, the efficiency index of the proposed method (11), (23) and (37) is $4^{1/2} = 8^{1/3} = 16^{1/4} = 2$.*

4 Numerical results and comparisons

The errors $|x_k - \alpha|$ of approximations to the sought zeros, produced by the different methods at the first three iterations are given in Table 2 where $m(-n)$ stands for $m \times 10^{-n}$. Tables $2-4$ also include, for each test function, the initial estimation values and the last value of the computational order of convergence COC [18] computed by the expression

$$COC = \frac{\log|f(x_n)/f(x_{n-1})|}{\log|f(x_{n-1})/f(x_{n-2})|}, \tag{54}$$

The package Mathematica 10, with 5000 arbitrary precision arithmetic, has been used in our computations. Iterative methods with and without memory, for comparing with our proposed scheme have been chosen as comes next.
Four-step without memory Geum et al. order 16 (GKM) [17]:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, u_k = \frac{f(y_k)}{f(x_k)}, m_k = \frac{1+2u_k-4u_k^2}{1-3u_k^2}, \, k = 0, 1, 2, \ldots, \\ z_k = y_k - m_k \frac{f(y_k)}{f'(x_k)}, v_k = \frac{f(z_k)}{f(y_k)}, w_k = \frac{f(z_k)}{f(x_k)}, h_k = \frac{1+2u_k}{1-v_k-2w_k}, t_k = \frac{f(s_k)}{f(z_k)}, \\ W_k = \frac{1+2u_k}{1-v_k-2w_k-t_k-2v_kw_k} - \\ \frac{1}{2}(u_kw_k(6+12u_k+2u_k^2+48u_k^3-8)) + (-2u_k+2)w_k^2, \\ s_k = z_k - h_k\frac{f(z_k)}{f'(x_k)}, x_{k+1} = s_k - W_k\frac{f(s_k)}{f'(x_k)}. \end{cases} \tag{55}$$

One-step with memory Dzunic order 3.56 (DM) [14]:

$$\begin{cases} \gamma_k = \frac{-1}{N_2'(x_k)}, q_k = \frac{N_3''(w_k)}{-2N_3'(w_k)}, \, k = 1, 2, \ldots, \\ w_k = x_k - \gamma_k f(x_k), \, x_{k+1} = x_k - \frac{f(x_k)}{f[x_k,w_k]+q_kf(w_k)}, \, k = 0, 1, 2, \ldots. \end{cases} \tag{56}$$

One-step Abbasbandy's method order 3 (AM) [1]:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{f^2(x_k)f''(x_k)}{2f'^3(x_k)} + \frac{f^3(x_k)f''(x_k)}{2f'^5(x_k)}. \tag{57}$$

Two-step with memory Soleymani et al. order 7.22 (SLTKM) [39]:

$$\begin{cases} \gamma_k = -\frac{1}{N_3'(x_k)}, q_k = -\frac{N_4''(w_k)}{2N_4'(w_k)}, \lambda_k = \frac{N_5'''(w_k)}{6}, \, k = 1, 2, 3, \ldots, \\ y_k = x_k - \frac{f(x_k)}{f[x_k,w_k]+q_kf(w_k)}, \, k = 0, 1, 2, \cdots, \\ x_{k+1} = y_k - \frac{f(y_k)}{f[w_k,y_k]+q_kf(w_k)+\lambda_k(y_k-x_k)(y_k-w_k)}(1+\frac{f(y_k)}{f(x_k)}). \end{cases} \tag{58}$$

Three-step without memory Thukral-Petkovic. order 8 (TPM) [41]:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, u_k = \frac{f(y_k)}{f(x_k)}, \, k = 0, 1, 2, \ldots, \\ z_k = y_k - \frac{f(y_k)}{f'(x_k)}\frac{f(x_k)+bf(y_k)}{f(x_k)+(b-2)f(y_k)}, \\ \phi_k = 1 + 2u_k + (5-2b)u_k^2 + (2b^2-12b+12)u_k^3, \\ x_{k+1} = z_k - \frac{f(z_k)}{f'(x_k)}(\phi_k + \frac{f(z_k)}{f(y_k)-af(z_k)} + \frac{4f(z_k)}{f(x_k)}). \end{cases} \tag{59}$$

Table 1 lists the exact roots α and initial approximations x_0. Tables $2-4$ show that the proposed methods compete the previous methods. In addition, its efficiency index is much better than the previous works. In other words, TAM4, TAM7, TAM8 and TAM16 have efficiency indices $4^{\frac{1}{2}} = 2$, $7.53^{\frac{1}{3}} \simeq 1.96$, $8^{\frac{1}{3}} = 2$ and $16^{\frac{1}{4}} = 2$, respectively. In order to check the effectiveness of the proposed iterative methods, we have considered 10 test nonlinear functions. All the numerical computations are carried out on the computer algebra system MATHEMATICA 10 using 3000 digits floating-point arithmetic. The results of comparisons are given in Tables $2-4$. The errors $|x_k-\alpha|$ of approximations to the sought zeros, produced by the different methods at the first, two, and three iterations. These tables also include, for each test function, the initial estimation values and the last value of the computational order of convergence in companion with convergence rate and EI the each method. A comparison between without memory, with memory and adaptive methods in terms of the maximum convergence order alongside the number of steps per cycle is given in Figure. 1.

Nonlinear function	Zero	Initial guess
$f_1(x)=t\log(1+x\sin(x))+e^{-1+x^2+x\cos(x)}\sin(\pi x)$	$\alpha=0$	$x_0=0.6$
$f_2(x)=1+\frac{1}{x^4}-\frac{1}{x}-x^2$	$\alpha=1$	$x_0=1.4$
$f_3(x)=e^{x^3-x}-\cos(x^2-1)+x^3+1$	$\alpha=-1$	$x_0=-1.65$
$f_4(x)=\frac{-5x^2}{2}+x^4+x^5+\frac{1}{1+x^2}$	$\alpha=1$	$x_0=1.5$
$f_5(x)=\log(1+x^2)+e^{-3x+x^2}\sin(x)$	$\alpha=0$	$x_0=0.5$
$f_6(x)=x\log(1-\pi+x^2)-\frac{1+x^2}{1+x^3}\sin(x^2)+\tan(x^2)$	$\alpha=\sqrt{\pi}$	$x_0=1.7$
$f_7(x)=x^3+4x^2-10$	$\alpha=1$	$x_0=1.3652$

Table 1: Test functions

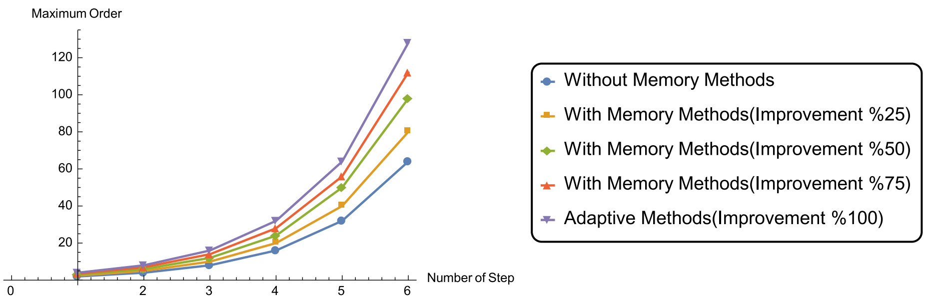

Fig. 1: Comparison of methods without memory, with memory and adaptive (%25, %50, %75, and %100 of improvements) in terms of highest possible convergence order.

$f_1(x) = x\log(1+x\sin(x)) + e^{-1+x^2+x\cos(x)}\sin(\pi x), \alpha = 0, x_0 = 0.6, q_0 = \gamma_0 = 0.1$					
Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	COC	EI
AM [1]	0.60000(0)	0.44377(0)	0.10028(0)	3.0000	1.44225
DM [14]	0.60000(0)	0.36450(0)	0.54166(−1)	3.4590	1.85984
TM [43]	0.60000(0)	0.47811(0)	0.56230(−1)	2.3950	1.54758
TAM4 (11) k=4	0.36450(0)	0.54166(−1)	0.23973(−5)	4.0148	2.00370
$f_2(x) = 1 + \frac{1}{x^4} - \frac{1}{x} - x^2, \alpha = 1, x_0 = 1.4, q_0 = \gamma_0 = 0.1$					
AM [1]	0.40000(0)	0.69117(−1)	0.84282(−3)	3.0000	1.44225
DM [14]	0.40000(0)	0.46538(−1)	0.12681(−3)	3.5552	1.88552
TM [43]	0.40000(0)	0.60801(−1)	0.28094(−2)	2.4157	1.55425
TAM4 (11) k=4	0.46538(−1)	0.12681(−3)	0.35998(−15)	3.9986	1.99965
$f_3(x) = e^{x^3-x} - \cos(x^2-1) + x^3 + 1, \alpha = -1, x_0 = -1.5, q_0 = \gamma_0 = 0.1$					
AM [1]	0.50000(0)	0.48941(−1)	0.16236(−3)	3.0000	1.44225
DM [14]	0.50000(0)	0.15659(−1)	0.10877(−5)	3.4075	1.84594
TM [43]	0.50000(0)	0.22068(−1)	0.12109(−5)	2.3993	1.54897
TAM4 (11) k=4	0.15659(−1)	0.10877(−5)	0.83019(−24)	4.0020	2.00050
$f_4(x) = \frac{-5x^2}{2} + x^4 + x^5 + \frac{1}{1+x^2}, \alpha = 1, x_0 = 1.5, q_0 = \gamma_0 = 0.1$					
AM [1]	0.50000(0)	0.18311(0)	0.33638(−1)	3.0000	1.44225
DM [14]	0.50000(0)	0.41826(0)	0.71739(−1)	3.5453	1.88290
TM [43]	0.50000(0)	0.41154(0)	0.13304(0)	2.2867	1.51218
TAM4 (11) k=4	0.41826(0)	0.71739(−1)	0.30353(−3)	3.9997	1.99992
$f_5(x) = \log(1+x^2) + e^{-3x+x^2}\sin(x), \alpha = 0, x_0 = 0.5, q_0 = \gamma_0 = 0.1$					
AM [1]	0.50000(0)	0.88441(−4)	0.45840(−12)	3.0000	1.44225
DM [14]	0.50000(0)	0.64108(−1)	0.12721(−2)	3.5500	1.88414
TM [43]	0.50000(0)	0.42599(−1)	0.11207(−2)	2.4134	1.55351
TAM4 (11) k=4	0.64108(−1)	0.12721(−2)	0.10199(−10)	3.9993	3.99982
$f_6(x) = x\log(1-\pi+x^2) - \frac{1+x^2}{1+x^3}\sin(x^2) + \tan(x^2), \alpha = \sqrt{\pi}, x_0 = 1.7, q_0 = \gamma_0 = 0.1$					
AM [1]	0.72454(−1)	0.58079(−3)	0.33518(−9)	3.0000	1.44225
DM [14]	0.72454(−1)	0.10543(−1)	0.44438(−6)	3.5005	1.87096
TM [43]	0.72454(−1)	0.11486(−1)	0.28170(−5)	2.4090	1.55210
TAM4 (11) k=4	0.10543(−1)	0.44438(−6)	0.48786(−24)	3.9992	1.99980
$f_7(x) = x^3 + 4x^2 - 10, \alpha = 1.3652, x_0 = 1, q_0 = \gamma_0 = 0.1$					
AM [1]	0.36520(0)	0.47568(−1)	0.22845(−4)	3.0000	1.44225
DM [14]	0.36520(0)	0.36340(0)	0.34044(−3)	3.7222	1.92930
TM [43]	0.36520(0)	0.27996(0)	0.64692(−2)	2.4053	1.55090
TAM4 (11) k=4	0.36340(0)	0.34044(−3)	0.30013(−4)	4.0000	2.00000

Table 2: Comparison of the absolute error of proposed method with one-step methods at first, second and third iterations for the test functions

$f_1(x) = x\log(1 + x\sin(x)) + e^{-1+x^2+x\cos(x)}\sin(\pi x),\ \alpha = 0,\ x_0 = 0.6, q_0 = \gamma_0 = \lambda_0 = 0.1$					
Methods	$\lvert x_1 - \alpha\rvert$	$\lvert x_2 - \alpha\rvert$	$\lvert x_3 - \alpha\rvert$	COC	EI
SLTKM [39]	0.60000(0)	0.22353(0)	0.30292(−5)	7.1871	1.92982
TAM7 (23) k=3	0.60000(0)	0.22353(0)	0.24015(−5)	7.5523	1.96197
TAM8 (23) k=4	0.22353(0)	0.24015(−5)	0.61245(−39)	8.1640	2.01357
$f_2(x) = 1 + \frac{1}{x^4} - \frac{1}{x} - x^2, \alpha = 1, x_0 = 1.4, q_0 = \gamma_0 = \lambda_0 = 0.1$					
SLTKM [39]	0.40000(0)	0.37115(−2)	0.28982(−15)	7.2315	1.93379
TAM7 (23) k=3	0.40000(0)	0.37115(−2)	0.37500(−16)	7.5218	1.95933
TAM8 (23) k=4	0.37115(−2)	0.37500(−16)	0.79732(−112)	8.1885	2.01559
$f_3(x) = e^{x^3-x} - \cos(x^2 - 1) + x^3 + 1, \alpha = -1, x_0 = -1.5, q_0 = \gamma_0 = \lambda_0 = 0.1$					
SLTKM [39]	0.50000(0)	0.48587(−4)	0.32385(−4)	7.2038	1.93132
TAM7 (23) k=3	0.50000(0)	0.48587(−4)	0.67397(−26)	7.4964	1.95712
TAM8 (23) k=4	0.48587(−4)	0.67397(−26)	0.54183(−180)	8.2048	2.01692
$f_4(x) = \frac{-5x^2}{2} + x^4 + x^5 + \frac{1}{1+x^2}, \alpha = 1, x_0 = 1.5, q_0 = \gamma_0 = \lambda_0 = 0.1$					
SLTKM [39]	0.50000(0)	0.31600(0)	0.42113(−2)	7.2133	1.93217
TAM7 (23) k=3	0.50000(0)	0.31600(0)	0.38872(−2)	7.9861	1.99884
TAM8 (23) k=4	0.31600(0)	0.38872(−2)	0.22090(−16)	8.0000	2.00000
$f_5(x) = \log(1 + x^2) + e^{-3x+x^2}\sin(x), \alpha = 0, x_0 = 0.5, q_0 = \gamma_0 = \lambda_0 = 0.1$					
SLTKM [39]	0.50000(0)	0.22780(−1)	0.13805(−10)	7.2390	1.93446
TAM7 (23) k=3	0.50000(0)	0.22780(−1)	0.91585(−11)	7.4955	1.95704
TAM8 (23) k=4	0.22780(−1)	0.91585(−11)	0.54892(−74)	8.0000	2.00000
$f_6(x) = x\log(1 - \pi + x^2) - \frac{1+x^2}{1+x^3}\sin(x^2) + \tan(x^2), \alpha = \sqrt{\pi}, x_0 = 1.7, q_0 = \gamma_0 = \lambda_0 = 0.1$					
SLTKM [39]	0.72454(−1)	0.70222(−5)	0.25138(−29)	7.2120	1.93205
TAM7 (23) k=3	0.72454(−1)	0.70222(−5)	0.25789(−31)	7.4793	1.95563
TAM8 (23) k=4	0.70222(−5)	0.25789(−31)	0.33555(−214)	7.9996	1.99997
$f_7(x) = x^3 + 4x^2 - 10, \alpha = 1.3652, x_0 = 1, q_0 = \gamma_0 = \lambda_0 = 0.1$					
SLTKM [39]	0.36520(0)	0.61026(0)	0.23768(−3)	8.0000	2.00000
TAM7 (23) k=3	0.36500(0)	0.61026(0)	0.23768(−3)	8.0000	2.00000
TAM8 (23) k=4	0.61026(0)	0.23768(−3)	0.23001(−3)	8.0000	2.00000

Table 3: Comparison evaluation function and efficiency index of the proposed method by two-step methods with and without memory

$f_1(x) = x\log(1+x\sin(x)) + e^{-1+x^2+x\cos(x)}\sin(\pi x),\ \alpha=0,\ x_0=0.6, q_0=\gamma_0=\lambda_0=\beta_0=0.1$					
Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	COC	EI
TPM [41] $(a=b=0)$	0.60000(0)	0.15946(−1)	0.31506(−13)	8.0000	1.68179
LAM [27]	0.60000(0)	0.19386(−1)	0.12850(−28)	15.5240	1.98496
GKM [17]	0.60000(0)	0.20973(−3)	0.67159(−57)	16.0000	1.74110
SSSLM [35]	0.60000(0)	0.15176(−3)	0.77529(−57)	16.0000	1.74110
TAM16 (37) k=4	0.19386(−1)	0.12850(−28)	0.373(−464)	16.0010	2.00003
$f_2(x) = 1+\frac{1}{x^4}-\frac{1}{x}-x^2, \alpha=1, x_0=1.4, q_0=\gamma_0=\lambda_0=\beta_0=0.1$					
TPM [41] $(a=b=0)$	0.40000(0)	0.20584(−3)	0.94711(−27)	8.0000	1.68179
LAM [27]	0.40000(0)	0.33505(−4)	0.80238(−66)	15.5120	1.98457
GKM [17]	0.40000(0)	0.11559(−3)	0.22543(−57)	16.0000	1.74110
SSSLM [35]	0.40000(0)	0.11522(−2)	0.50671(−39)	16.0000	1.74110
TAM16 (37) k=4	0.33505(−4)	0.80238(−66)	0.12659(−1055)	16.0000	2.00000
$f_3(x) = e^{x^3-x}-\cos(x^2-1)+x^3+1, \alpha=-1, x_0=-1.5, q_0=\gamma_0=\lambda_0=\beta_0=0.1$					
TPM [41] $(a=b=0)$	0.50000(0)	0.22661(−5)	0.12409(−44)	8.0000	1.68179
LAM [27]	0.50000(0)	0.32145(−6)	0.34397(−105)	15.5100	1.98451
SSSLM [35]	0.50000(0)	0.18741(−10)	0.23265(−170)	16.0000	1.74110
GKM [17]	0.50000(0)	0.71640(−9)	0.10752(−146)	16.0000	1.74110
TAM16 (37) k=4	0.32145(−6)	0.34397(−105)	0.70235(−1693)	16.0000	2.00000
$f_4(x) = \frac{-5x^2}{2}+x^4+x^5+\frac{1}{1+x^2}, \alpha=1, x_0=1.5, q_0=\gamma_0=\lambda_0=\beta_0=0.1$					
TPM [41] $(a=b=0)$	0.50000(0)	0.52236(−1)	0.29347(−5)	8.0000	1.68179
LAM [27]	0.50000(0)	0.73848(−1)	0.60939(−12)	15.6030	1.98748
GKM [17]	0.50000(0)	0.10118(−1)	0.29389(−19)	16.0000	1.74110
SSSLM [35]	0.50000(0)	0.25125(−1)	0.16956(−15)	16.0000	1.74110
TAM16 (37) k=4	0.73848(−1)	0.60939(−12)	0.10730(−188)	16.0000	2.00000
$f_5(x) = \log(1+x^2)+e^{-3x+x^2}\sin(x), \alpha=0, x_0=0.5, q_0=\gamma_0=\lambda_0=\beta_0=0.1$					
TPM [41] $(a=b=0)$	0.50000(0)	0.54581(−2)	0.75773(−15)	8.0000	1.68179
LAM [27]	0.50000(0)	0.55075(−3)	0.30763(−48)	15.5080	1.98444
GKM [17]	0.50000(0)	0.46311(−6)	0.28713(−94)	16.0000	1.74110
SSSLM [35]	0.50000(0)	0.55428(−3)	0.10344(−42)	16.0000	1.74110
TAM16 (37) k=4	0.55075(−3)	0.30763(−48)	0.11682(−771)	16.0000	2.00000
$f_6(x) = x\log(1-\pi+x^2)-\frac{1+x^2}{1+x^3}\sin(x^2)+\tan(x^2), \alpha=\sqrt{\pi}, x_0=1.7, q_0=\gamma_0=\lambda_0=\beta_0=0.1$					
Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	COC	EI
TPM [41] $(a=b=0)$	0.72454(−1)	0.21456(−8)	0.94711(−69)	8.0000	1.68179
GKM [17]	0.72454(−1)	0.43034(−19)	0.25205(−310)	16.0000	1.74110
LAM [27]	0.72454(−1)	0.27167(−6)	0.15849(−97)	15.5120	1.98457
SSSLM [35]	0.72454(−1)	0.11233(−14)	0.10052(−237)	16.0000	1.74110
TAM16 (37) k=4	0.27167(−6)	0.15849(−97)	0.19365(−1557)	16.0000	2.00000
$f_7(x) = x^3+4x^2-10, \alpha=1.3652, x_0=1, q_0=\gamma_0=\lambda_0=\beta_0=0.1$					
TPM [41] $(a=b=0)$	0.36520(0)	0.69070(−3)	0.30013(−4)	8.0000	1.68179
LAM [27]	0.36520(0)	0.75262(−3)	0.30013(−4)	16.0000	2.00000
GKM [17]	0.36520(0)	0.84783(−3)	0.30013(−4)	16.0013	1.74392
SSSLM [35]	0.36520(0)	0.19417(−1)	0.30013(−4)	16.0000	1.74110
TAM16 (37) k=4	0.95262(−3)	0.23001(−3)	0.23001(−3)	16.0000	2.00000

Table 4: Comparison evaluation function and efficiency index of the proposed method by three- and four-step methods with and without memory

5 Conclusion

In this study, we have increased convergence - order methods $2, 4$ and 8 without imposing new evaluation on different recursive methods, with a convergence order of $4, 8$ and 16, 100% improvement, respectively. To this end, based on Newton's interpolation, the parameters of self-evaluation are interpolated. The numerical results show that proposed method is very useful to find an acceptable approximation of the exact solution of nonlinear equations, specially when the function is non-differentiable. Table 2 compares one-step iterative with and without memory and the proposed method on functions $f_i(t)$, $i = 1, 2, \ldots, 7$. Similarly, Table 3 compares two-step iterative methods. Also Table 4 compares three- and four-step iterative methods with the proposed schemes. Last column of Tables show efficiency index defined by $EI = COC^{1/n}$, which is asymptotically 2. In other words, the proposed adaptive method with memory (11), (23), and (37) show a behavior as optimal n-point methods without memory. Therefore, we have developed a family iterative methods adaptive with memory which have efficiency index 2. The efficiency index of the proposed adaptive family with memory is $4^{\frac{1}{2}} = 8^{\frac{1}{3}} = 16^{\frac{1}{4}} = 2$ which is much better than optimal one-,..., five-point optimal methods without memory having efficiency indexes $2^{1/2} \simeq 1.414$, $4^{1/3} \simeq 1.587$, $8^{1/4} \simeq 1.681$, $16^{1/5} \simeq 1.741$, $32^{1/6} \simeq 1.781$, $64^{1/7} \simeq 1.811$, respectively. Adaptive methods with memory have minimum evaluation function, not evaluation derivative, and most efficiency index, hence compets with existing methods with- and without memory.

References

1. S. Abbasbandy, Modified homotopy perturbation method for nonlinear equations and comparison with Adomian decomposition method, Appl. Math. Comput. 172 (2006) 431–438.
2. I.K. Argyros, A.A. Magrenan, On the convergence of an optimal fourth-order family of methods and its dynamics, Appl. Math. Comput. 252 (2015) 336–346.
3. J. Biazar, B. Ghanbari, A new third-order family of nonlinear solvers for multiple roots, Computers and Mathematics with Applications. 59 (2010) 3315–3319.
4. R. Behl, A. Cordero, S.S. Motsa, J.R. Torregrosa, Stable high-order iterative mehods for solving nonlinear models, Appl. Math. Comput. 303 (2017) 70–88.
5. W. Bi, H. Ren, Q. Wu, Three-step iterative methods with eighth-order convergence for solving nonlinear equations, J. Comput. Appl. Math. 225 (2009) 105–112.
6. C. Chun, Cunstruction of Newton-like iteration methods for solving nonlinear equations, Numer. Math. 104 (2006) 297–315.
7. C. Chun, B. Neta, An analysis of a new family of eighth-order optimal methods, Appl. Math. Comput. 245 (2014) 86–107.
8. A. Cordero, J.L. Hueso, E. Martinez, J.R. Torregrosa, A family of derivative-free methods with high order of convergence and its application to nonsmooth equations, Abst. Appl. Anal. 2012 (2012) 1–15.
9. A. Cordero, J.L. Hueso, E. Martinez, J.R. Torregrosa, Generating optimal derivative free iterative methods for nonlinear equations by using polynomial interpolation, Math. Comput. Model. 57 (2013) 1950–1956.
10. A. Cordero, T. Lotfi, A. Khoshandi, J.R. Torregrosa, An efficient Steffensen-like iterative method with memry, Bull. Math. Soc. Sci. Math. Roum, Tome 58 (106) (1) (2015) 49–58.
11. A. Cordero, T. Lotfi, K. Mahdiani, J.R. Torregrosa, Two optimal general classes of iterative methods with eighth-order, Acta. Appl. Math. 134 (1) (2014) 61–74.
12. A. Cordero, T. Lotfi, J.R. Torregrosa, P. Assari, K. Mahdiani, Some new bi-accelarator two-point methods for solving nonlinear equations, Comput. Appl. Math. 35 (2016) 251–267.
13. A. Cordero, F. Soleymani, J.R. Torregrosa, S. Shateyi, Basins of attraction for various Steffensen-type methods, J. Appl. Math. 2014 (2014) 1–17.

14. J. Dzunic, On efficient two-parameter methods for solving nonlinear equations, Numer. Algor. 63 (2013) 549–569.
15. J. Dzunic, M.S. Petkovic, A cubicaly convergent Steffensen-like method for solving nonlinear equations, Appl. Math. Lett. 25 (2012) 1881–1886.
16. T. Eftekhari, An efficient class of multipoint root-solvers with and without memory for nonlinear equations, Acta Math. Vietnam. 41 (2) (2016) 299–311.
17. Y. H. Geum, Y. I. Kim, A biparametric family optimally convergent sixteenth-order multipoint methods with their fourth-step weighting function as a sum of a rational and a generic two-variable function, J. Comput. Appl. Math. 235 (2011) 3178–3188.
18. L.O. Jay, A note on Q-order of convergence, BIT. Numer. Math. 41 (2) (2001) 422–429.
19. B. Kalantari, On the Order of Convergence of a Determinantal Family of Root-Finding Methods, BIT. Numer. Math. 39(1) (1999) 96–109
20. R. F. King, A family of fourth order methods for nonlinear equations, Siam. J. Numer. Anal. 10 (5) (1973) 876–879.
21. R. F. King, A fifth-order family of modified Newton methods, BIT. Numer. Math. 11 (4) (1971) 409–412.
22. J. Kou, Y. Li, A variant of super-Halley method with accelerated fourth-order convergence, Appl. Math. Comput. 186 (2007) 535–539.
23. H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, J. Assoc. Comput. Mach. 21 (4) (1974) 643–651.
24. M.J. Lalehchini, T. Lotfi, K. Mahdiani, On Developing an Adaptive Free-Derivative Kung and Traub's Method with Memory, J. MATH. EXTEN. 14 (3) (2020) 1-20.
25. T. Lotfi, K. Mahdiani, P. Bakhtiarib, F. Soleymani, Constructing two-step iterative methods with and without memory, Comput. Math. Math. Physics, (2015) 55(2)183–193.
26. T. Lotfi, P. Assari, A new two-step class of methods with memory for solving nonlinear equations with high efficiency index, Int. J. Math. Model. Comput. 4 (3) (2014) 277–288.
27. T. Lotfi, P. Assari, New three-and four-parametric iterative with memory methods with efficiency index near 2, Appl. Math. Comput. 270 (2015) 1004–1010.
28. T. Lotfi, F. Soleymani, M. Ghorbanzadeh, P. Assari, On the cunstruction of some triparametric iterative methods with memory, Numer. Algor. 70 (4) (2015) 835–845.
29. A.K. Maheshwari, A fourth order iterative method for solving nonlinear equations, Appl. Math. Comput. 211 (2009) 383–391.
30. J.M. Ortega, W.G. Rheinboldt, Iterative solutions of nonlinear equations in several variables, Academic Press, New York, 1970.
31. A.M. Ostrowski, Solution of equations and systems of equations, Academic press, New York, 1960.
32. M.S. Petkovic, B. Neta, L.D. Petkovic, J. Dzunic, Multipoint methods for solving nonlinear equations, Elsevier, Amsterdam, 2013.
33. D. B. Popovski, A note on King's fifth-order family of methods for solving equations, BIT. Numer. Math. 21(1) (1981) 129-130.
34. J.Raj. Sharma, H. Arora, An efficient family of weighted-Newton methods with optimal eighth order convergence, Appl. Math. Lett. 29 (2014) 1–6.
35. S. Sharifi, M. Salimi, S. Siegmund, T. Lotfi, A new class of optimal four-point methods with convergence order 16 for solving nonlinear equations, Math. Comput. Simul. 119 (c) (2016) 69–90.
36. F. Soleymani, Concerning some sixth-order iterative methods for finding the simple roots of nonlinear equations, Bull. Math. Anal. Appl. 2 (4) (2010) 146–151.
37. F. Soleymani, On a bi-parametric classes of optimal eight-order derivative-free methods, Int. J. Pur. Appl. Math. 72 (1) (2011) 27–37.
38. F. Soleymani, V. Hossienabadi, New third-and sixth-order derivative-free techniques for nonlinear equations, J. Math. Rese. 3 (2) (2011) 107–112.
39. F. Soleymani, T. Lotfi, E. Tavakoli, F. Khaksar Haghani, Several iterative methods with memory using self-accelerators, Appl. Math. Comput. 254 (2015) 452–458.
40. J.F. Steffensen, Remarks on iteration, Scandinavian Aktuarietidskr, 16 (1933) 64–72.
41. R. Thukral, M.S. Petkovic, A family of three-point methods of optimal order for solving nonlinear equations, J. Comput. Appl. Math. 233 (2010) 2278–2284.
42. V. Torkashvand, T. Lotfi, M.A.Fariborzi Araghi, A New Family of Adaptive Methods with Memory for Solving Nonlinear Equations, Math. Sci. 13 (2019) 1–20.
43. J.F. Traub, Iterative Methods for the Solution of Equations, Prentice Hall, New York, USA, 1964.

44. H. Veiseh, T. Lotfi, T. Allahviranloo, A study on the local convergence and dynamics of the two-step and derivative-free Kung-Traub's method, Comput. Appl. Math. 37 (3) (2018) 2428–2444.
45. X. Wang, An Ostrowski-type method with memory using a novel self-accelerating parameter, J. Comput. Appl. Math. 330 (2017) 1–18.
46. X. Wang, T. Zhang, High-order Newton-type iterative methods with memory for solving nonlinear equations, Math. Commun. 19 (2014) 91–109.
47. Q. Zheng, J. Li, F. Huang, An optimal Steffensen-type family for solving nonlinear equations, Appl. Math. Comput. 217 (2011) 9592–9597.