Research Article

Efficient Iterative Methods with and without Memory Possessing High Efficiency Indices

T. Lotfi,¹ F. Soleymani,² Z. Noori,¹ A. Kılıçman,³ and F. Khaksar Haghani²

¹ Department of Mathematics, Islamic Azad University, Hamedan Branch, Hamedan, Iran

² Department of Mathematics, Islamic Azad University, Shahrekord Branch, Shahrekord, Iran

³ Department of Mathematics and Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Malaysia

Correspondence should be addressed to A. Kılıçman; akilic@upm.edu.my

Received 2 June 2014; Accepted 20 July 2014; Published 3 September 2014

Academic Editor: Krzysztof Ciepliński

Copyright © 2014 T. Lotfi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Two families of derivative-free methods without memory for approximating a simple zero of a nonlinear equation are presented. The proposed schemes have an accelerator parameter with the property that it can increase the convergence rate without any new functional evaluations. In this way, we construct a method with memory that increases considerably efficiency index from $8^{1/4} \approx 1.681$ to $12^{1/4} \approx 1.861$. Numerical examples and comparison with the existing methods are included to confirm theoretical results and high computational efficiency.

1. Preliminaries

The main goal and motivation in constructing iterative methods for solving nonlinear equations is to obtain as high as possible order of convergence with minimal computational cost (see, e.g., [1–3]). Hence, many researchers (see, e.g., [4, 5]) have paid much attention to construct optimal multipoint methods without memory based on the unproved conjecture of Kung and Traub [6], which indicates that any multipoint iterative method without memory using $n + 1$ functional evaluations can reach the optimal order 2^n .

Let α be a simple real zero of a real function $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$ and let x_0 be an initial approximation to α . In many practical situations, it is preferable to avoid calculations of derivatives of f . This makes the construction of iterative derivative-free methods important [7].

This paper follows two main goals. First, developing some optimal three-step derivative-free families of methods without memory. And second, developing the proposed families to methods with memory in a way that reaches convergence R-orders 6 and 12 without any new functional evaluation.

The main idea in methods with memory is based on the use of suitable two-valued functions and the variation of a free

parameter in each iterative step. This parameter is calculated using information from the current and previous iteration so that the developed methods may be regarded as methods with memory following Traub's classification [8]. A supplemental inspiration for studying methods with memory stands up from an astonishing fact that such classes of iterative methods have been considered in the literature very rarely despite their high efficiency indices.

The paper is organized as follows. First, two families of optimal methods with orders four and eight consuming three and four function evaluations per iteration, respectively, are proposed in Section 2. Then in Section 3, we state methods with memory of very high computational efficiencies. The increase of convergence speed is achieved without additional function evaluations, which is the main advantage of the methods with memory. Section 4 is assigned to numerical results connected to the order of the methods with and without memory. Concluding remarks are given in Section 5.

2. Construction of New Families

This section deals with constructing new multipoint methods for solving nonlinear equations. The discussions are divided into two subsections. Let the scalar function $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$

and $f(\alpha) = 0 \neq f'(\alpha) = c_1$; that is, α is a simple zero of $f(x) = 0$.

2.1. New Two-Step Methods without Memory. In this subsection, we start with the two-step scheme:

$$y = x - \frac{f(x)}{f'(x)}, \quad \bar{x} = y - \frac{f(y)}{f'(y)}. \quad (1)$$

Note that we omit the iteration index k for the sake of simplicity only. The order of convergence of scheme (1) is four, but it does not contain any contributions. We substitute derivatives in the first and second steps by suitable approximations that use available data; thus, we introduce an approximation as follows.

Using the points x and $w = x + \gamma f(x)$ (γ is a nonzero real constant), we can apply Lagrange's interpolatory polynomial for approximating $f'(x)$:

$$P_1(t) = \frac{t-w}{x-w}f(x) + \frac{t-x}{w-x}f(w). \quad (2)$$

Fix $f'(x) \approx P_1'(x)$, and by setting $t = x$, we have

$$P_1'(x) = \frac{f(x)}{x-w} + \frac{f(w)}{w-x} = \frac{f(x) - f(w)}{x-w} = f[x, w]. \quad (3)$$

Also, Lagrange's interpolatory polynomial at points x, w , and y for approximating $f'(y)$ can be given as follows:

$$P_2(t) = \frac{(t-w)(t-y)}{(x-w)(x-y)}f(x) + \frac{(t-x)(t-y)}{(w-x)(w-y)}f(w) + \frac{(t-w)(t-x)}{(y-w)(y-x)}f(y). \quad (4)$$

We obtain

$$\begin{aligned} f'(y) &\approx P_2'(y) \\ &= \frac{2y-w-y}{(x-w)(x-y)}f(x) \\ &\quad + \frac{2y-x-y}{(w-x)(w-y)}f(w) \\ &\quad + \frac{2y-w-x}{(y-w)(y-x)}f(y). \end{aligned} \quad (5)$$

Finally, we set above approximations in the denominator of (1), and so our derivative-free two-step iterative method is derived in what follows:

$$\begin{aligned} y &= x - \frac{f(x)}{f[x, w]}, \quad w = x + \gamma f(x), \\ z &= y - \frac{f(y)}{P_2'(y)}. \end{aligned} \quad (6)$$

Now to check the convergence order of (6), we avoid retyping the widely practiced approach in the literature and put forward the following self-explained Mathematica code:

```
f[e_] = f1a * (e + Sum[c_k * e^k, {k, 2, 4}]);
ew = e + γf[e] (*ew = w - α*);
```

```
f[x_, y_] := (f[x] - f[y]) / (x - y);
p1[t_] := (t - ew) / (e - ew) f[e] + (t - e) / (ew - e) f[ew];
dp1[t_] := (f[e] / (e - ew) + f[ew] / (ew - e));
ey = e - Series[f[e] / p1'[ew], {e, 0, 3}] // FullSimplify (*ey = y - α*);
Out[a] : (1 + f1a γ) c_2 e^2 + 0[e]^3.
p2[t_] = (t - ew)(t - ey) / ((e - ew)(e - ey)) f[e]
+ ((t - e)(t - ey) / ((ew - e)(ew - ey)) f[ew]
+ ((t - ew)(t - e) / ((ey - ew)(ey - e)) f[ey];
dp2[t_] := (2t - ew - ey) / ((e - ew)(e - ey)) f[e]
+ (2t - e - ey) / ((ew - e)(ew - ey)) f[ew]
+ (2t - ew - e) / ((ey - ew)(ey - e)) f[ey];
ez = ey - f[ey] / dp2[ey] // FullSimplify
Out[b] : (1 + f1a γ)^2 c_2 (c_2^2 - c_3) e^4 + 0[e]^5.
```

Considering Out [b] of the above Mathematica program, we observe that the order of convergence of the family (6) is four, and so we can state a convergence theorem in what follows.

Theorem 1. *If an initial approximation x_0 is sufficiently close to a simple zero α of f , then the convergence order of the two-step approach (6) is equal to four. And its error equation is given by*

$$e_{k+1} = (1 + \gamma f'(\alpha))^2 c_2 (c_2^2 - c_3) e_k^4 + O(e_k^5). \quad (7)$$

2.2. New Three-Step Family. Now, we construct a three-step uniparametric family of methods based on the two-step method (6). We start from a three-step scheme where the first two steps are given by (6), and also the third step is Newton's method; that is,

$$\begin{aligned} y &= x - \frac{f(x)}{f[x, w]}, \quad w = x + \gamma f(x), \\ z &= y - \frac{f(y)}{P_2'(y)}, \\ \hat{x} &= z - \frac{f(z)}{f'(z)}. \end{aligned} \quad (8)$$

The derivative $f'(z)$ in the third step of (8) should be substituted by a suitable approximation in order to obtain as high as possible of convergence order and to make the scheme optimal. To provide this approximation, we apply Lagrange's

interpolatory polynomial at the points $w = x + \gamma f(x)$, x , y , and z ; that is,

$$P_3(t) = \frac{(t-w)(t-y)(t-z)}{(x-w)(x-y)(x-z)}f(x) + \frac{(t-x)(t-y)(t-z)}{(w-x)(w-y)(w-z)}f(w) + \frac{(t-w)(t-x)(t-z)}{(y-w)(y-x)(y-z)}f(y) + \frac{(t-w)(t-x)(t-y)}{(z-w)(z-x)(z-y)}f(z). \quad (9)$$

It is obvious that $P_3(z) = f(z)$. By differentiating (9) and setting $t = z$, we obtain

$$P'_3(z) = \frac{wy + wz + yz - 2(w + y + z)z + 3z^2}{((x-w)(x-y)(x-z))}f(x) + \frac{xy + xz + yz - 2(x + y + z)z + 3z^2}{(w-x)(w-y)(w-z)}f(w) + \frac{xw + xz + wz - 2(x + w + z)z + 3z^2}{(y-w)(y-x)(y-z)}f(y) + \frac{xw + xy + wy - 2(x + w + y)z + 3z^2}{(z-w)(z-x)(z-y)}f(z). \quad (10)$$

By substituting $f'(z) \approx P'_3(z)$ in (8), we have

$$y = x - \frac{f(x)}{f[x, w]}, \quad w = x + \gamma f(x), \\ z = y - \frac{f(y)}{P'_2(y)}, \\ \hat{x} = z - \frac{f(z)}{P'_3(z)}, \quad (11)$$

where $P'_2(y)$ is defined by (5) and $P'_3(z)$ given by (10).

In the following theorem, we state suitable conditions for deriving an optimal three-step scheme without memory.

Theorem 2. Let $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ be a scalar function which has the simple root α in the open interval D ; also initial approximation x_0 is sufficiently close to a simple zero α of f . The three-step iterative method (11) has eighth order and satisfies the following error equation:

$$e_{k+1} = (1 + \gamma f'(\alpha)^4) c_2^2 (c_2^2 - c_3) (c_2^3 - c_2 c_3 + c_4) e_k^8 + O(e_k^9). \quad (12)$$

Proof. We are going to employ symbolic computation in the computational software package Mathematica. We write the following:

$$f(x_k) = f'(\alpha) (e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 + c_5 e_k^5 + c_6 e_k^6 + c_7 e_k^7 + c_8 e_k^8 + O(e_k^9)). \quad (13)$$

Now by introducing the abbreviations $e = x - \alpha$, $ew = w - \alpha$, $ey = y - \alpha$, $ez = z - \alpha$, and $c_k = f^{(k)}(\alpha)/(k!f'(\alpha))$, we provide

the following Mathematica program in order to obtain the convergence order of (11).

Program Written in Mathematica. Consider the following.

```
f[e_] = fla * (e + Sum[c_k * e^k, {k, 2, 4}];
ew = e + gamma f[e];
f[x_, y_] := (f[x] - f[y]) / (x - y);
p1[t_] := (t - ew) f[e] + (t - e) f[ew] / (e - ew);
dp1[t_] := (f[e] / (e - ew) + f[ew] / (ew - e));
ey = e - Series[f[e] / p1'[ew], {e, 0, 8}] // FullSimplify
Out[a] : (1 + gamma fla) c_2 e^2 + O[e]^3.
p2[t_] := (t - ew)(t - ey) f[e] / ((e - ew)(e - ey)) + (t - e)(t - ey) f[ew] / ((ew - e)(ew - ey)) + (t - ew)(t - e) f[ey] / ((ey - ew)(ey - e));
dp2[t_] := (2t - ew - ey) f[e] / ((e - ew)(e - ey)) + (2t - e - ey) f[ew] / ((ew - e)(ew - ey)) + (2t - ew - e) f[ey] / ((ey - ew)(ey - e));
ez = ey - f[ey] / dp2[ey] // FullSimplify
Out[b] : (1 + gamma fla)^2 c_2 (c_2^2 - c_3) e^4 + O[e]^5.
dp3[t_] := (ewey + ewez + eyez - 2(ew + ey + ez)t + 3t^2) f[e] / ((e - ew)(e - ey)(e - ez)) + (eey + eez + eyez - 2(e + ey + ez)t + 3t^2) f[ew] / ((ew - e)(ew - ey)(ew - ez)) + (eew + eez + ewez - 2(e + ew + ez)t + 3t^2) f[ey] / ((ey - ew)(ey - e)(ey - ez)) + (eew + eey + ewey - 2(e + ew + ey)t + 3t^2) f[ez] / ((ez - ew)(ez - e)(ez - ey));
ex = ez - f[ez] / dp3[ez] // FullSimplify
Out[c] : (1 + gamma fla)^4 c_2^2 (c_2^2 - c_3) (c_2^3 - c_2 c_3 + c_4) e^8 + O[e]^9.
```

As a result, the proof of the theorem is finished. According to Out[c], it possesses eighth order. $\square$

Error equations (7) and (12) indicate that the orders of methods (6) and (11) are four and eight, respectively.

In the next section, we will modify the proposed methods and introduce new methods with memory. With the use of accelerator parameters, the order of convergence will significantly increase.

3. Extension to Methods with Memory

This section is concerned with extraction of efficient methods with memory from (6) and (11), through a careful inspection of their error equations containing the parameter γ , which can be approximated in such a way that increases the local order of convergence.

Toward this goal, we set $\gamma = \gamma_k$ as the iteration proceeds by the formula $\gamma_k = -1/\bar{f}'(\alpha)$ for $k = 1, 2, \dots$, where $\bar{f}'(\alpha)$ is an approximation of $f'(\alpha)$. We therefore use the approximation

$$\gamma_k = -\frac{1}{\bar{f}'(\alpha)} = -\frac{1}{N'_3(x_k)}, \quad (14)$$

for (6) and the following one

$$\gamma_k = -\frac{1}{\bar{f}'(\alpha)} = -\frac{1}{N'_4(x_k)}, \quad (15)$$

for (11). Here, we consider Newton's interpolating polynomial of third degree as the method for approximating $f'(\alpha)$ in two-step method (6) and Newton's interpolating polynomial of fourth degree for approximating $f'(\alpha)$ in the three-step method (11), where $N_3(t)$ is the Newton's interpolation polynomial of third degree, set through four available approximations $x_k, x_{k-1}, y_{k-1}, w_{k-1}$ and $N_4(t)$ is the Newton's interpolation polynomial of fourth degree, set through five available approximations $x_k, z_{k-1}, y_{k-1}, w_{k-1}, x_{k-1}$. Consider

$$\begin{aligned} N'_3(x_k) &= \left[\frac{d}{dt} N_3(t) \right]_{t=x_k} \\ &= \left[\frac{d}{dt} (f(x_k) + f[x_k, x_{k-1}](t - x_k) \right. \\ &\quad + f[x_k, x_{k-1}, y_{k-1}](t - x_k)(t - x_{k-1}) \\ &\quad + f[x_k, x_{k-1}, y_{k-1}, w_{k-1}](t - x_k) \\ &\quad \left. \times (t - x_{k-1})(t - y_{k-1})) \right]_{t=x_k} \\ &= f[x_k, x_{k-1}] + f[x_k, x_{k-1}, y_{k-1}](x_k - x_{k-1}) \\ &\quad + f[x_k, x_{k-1}, y_{k-1}, w_{k-1}](x_k - x_{k-1})(x_k - y_{k-1}), \\ &\quad (16) \\ N'_4(x_k) &= \left[\frac{d}{dt} N_4(t) \right]_{t=x_k} \\ &= \left[\frac{d}{dt} (f(x_k) + f[x_k, z_{k-1}](t - x_k) \right. \\ &\quad + f[x_k, z_{k-1}, y_{k-1}](t - x_k)(t - z_{k-1}) \\ &\quad + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}](t - x_k) \end{aligned}$$

$$\begin{aligned} &\times (t - z_{k-1})(t - y_{k-1}) \\ &\quad + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}, w_{k-1}] \\ &\quad \left. \times (t - x_k)(t - z_{k-1})(t - y_{k-1})(t - x_{k-1})) \right]_{t=x_k} \\ &= f[x_k, z_{k-1}] + f[x_k, z_{k-1}, y_{k-1}](x_k - z_{k-1}) \\ &\quad + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}](x_k - z_{k-1})(x_k - y_{k-1}) \\ &\quad + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}, w_{k-1}](x_k - z_{k-1}) \\ &\quad \times (x_k - y_{k-1})(x_k - x_{k-1}). \end{aligned} \quad (17)$$

Note that a divided difference of order m , defined recursively as

$$\begin{aligned} &f[x_0, x_1, \dots, x_m] \\ &= \frac{f[x_1, \dots, x_m] - f[x_0, x_1, \dots, x_{m-1}]}{x_m - x_0}, \quad m \geq 2, \end{aligned} \quad (18)$$

has been used throughout this paper. Hence, the with memory developments of (6) and (11) can be presented as follows:

$$y_k = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \gamma_k f(x_k), \quad (19)$$

$$\begin{aligned} x_{k+1} &= y_k - \frac{f(y_k)}{P'_2(y_k)}, \\ y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \gamma_k f(x_k), \\ z_k &= y_k - \frac{f(y_k)}{P'_2(t_k)}, \\ \hat{x} &= x_{k+1} = z_k - \frac{f(z_k)}{P'_3(z_k)}. \end{aligned} \quad (20)$$

Remark 3. Accelerating methods obtained by recursively calculated free parameter may also be called self-accelerating methods. The initial value γ_0 should be chosen before starting the iterative process, for example, using one of the ways proposed in [8].

Here, we attempt to prove that the methods with memory (19) and (20) have convergence orders six and twelve provided that we use accelerator γ_k as in (14) and (15). For ease of continuing analysis, we introduce the convenient notation as follows. If the sequence $\{x_k\}$ converges to the zero α of f with the order p , then we write $e_{k+1} \sim e_k^p$, where $e_k = x_k - \alpha$. The following lemma will play a crucial role in improving the convergence order of the methods with memory to be proposed in this paper.

Lemma 4. *If $\gamma_k = -1/N'_3(x_k)$, $e_k = x_k - \alpha$, $e_{k,y} = y_k - \alpha$, and $e_{k,w} = w_k - \alpha$, then the following relation holds:*

$$1 + \gamma_k f'(\alpha) \sim c_4 e_{k-1} e_{k-1,w} e_{k-1,y} \sim e_{k-1} e_{k-1,w} e_{k-1,y}. \quad (21)$$

Proof. Following the same terminology as in Theorem 2 and the symbolic software Mathematica, it would be easy to obtain (21) via writing the following code:

```
ClearAll["Global`*"]
A[t_] := InterpolatingPolynomial[{{e, fx},
{ew, fw}, {ey, fy}, {el, fx1}}, t]//Simplify
Approximation = -1/A'[e1]//Simplify;
fx = fla * (e + c2 * e^2 + c3 * e^3 + c4 * e^4);
fw = fla * (ew + c2 * ew^2 + c3 * ew^3
+ c4 * ew^4);
fy = fla * (ey + c2 * ey^2 + c3 * ey^3
+ c4 * ey^4);
fx1 = fla * (el + c2 * el^2 + c3 * el^3
+ c4 * el^4);
b = Series[Approximation, {e, 0, 2}, {ew, 0, 2},
{ey, 0, 2}, {el, 0, 2}]//Simplify;
Collect[Series[1 + b * fla, {e, 0, 1}, {ew, 0, 1},
{ey, 0, 1}, {el, 0, 0}],
{e, ew, ey, el}, Simplify].
```

The proof is complete. $\square$

In order to obtain the R-order of convergence [9] of the method with memory (19), we establish the following theorem.

Theorem 5. *If an initial approximation x_0 is sufficiently close to the zero α of $f(x)$ and the parameter γ_k in the iterative scheme (19) is recursively calculated by the forms given in (14), then the R-order of convergence for (19) is at least six.*

Proof. Let $\{x_k\}$ be a sequence of approximations generated by the iterative method with memory (19). If this sequence converges to the zero α of f with the order p , then we write

$$e_{k+1} \sim e_k^p, \quad e_k = x_k - \alpha. \quad (22)$$

Thus

$$e_{k+1} \sim (e_{k-1}^p)^p = e_{k-1}^{p^2}. \quad (23)$$

Moreover, assume that the iterative sequences w_k and y_k have the orders p_1 and p_2 , respectively. Then, (22) gives

$$e_{k,w} \sim e_k^{p_1} \sim (e_{k-1}^p)^{p_1} = e_{k-1}^{pp_1}, \quad (24)$$

$$e_{k,y} \sim e_k^{p_2} \sim (e_{k-1}^p)^{p_2} = e_{k-1}^{pp_2}. \quad (25)$$

Since

$$e_{k,w} \sim (1 + \gamma_k f'(\alpha)) e_k, \quad (26)$$

$$e_{k,y} \sim (1 + \gamma_k f'(\alpha)) e_k^2, \quad (27)$$

$$e_{k+1} \sim (1 + \gamma_k f'(\alpha))^2 e_k^4, \quad (28)$$

using Lemma 4 and (27), induce

$$e_{k,w} \sim (1 + \gamma_k f'(\alpha)) e_k \sim (e_{k-1} e_{k-1,y} e_{k-1,w}) e_k = e_{k-1}^{p+p_1+p_2+1}, \quad (29)$$

$$e_{k,y} \sim (1 + \gamma_k f'(\alpha)) e_k^2 \sim (e_{k-1} e_{k-1,y} e_{k-1,w}) e_k^2 = e_{k-1}^{2p+p_1+p_2+1}, \quad (30)$$

$$\begin{aligned} e_{k+1} &\sim (1 + \gamma_k f'(\alpha))^2 e_k^4 \sim (e_{k-1} e_{k-1,y} e_{k-1,w})^2 e_k^4 \\ &= e_{k-1}^{4p+2p_1+2p_2+2}. \end{aligned} \quad (31)$$

Matching the powers of e_{k-1} on the right hand sides of (24)–(29), (25)–(30), and (23)–(31), one can obtain

$$\begin{aligned} pp_1 - p - p_1 - p_2 - 1 &= 0, \\ pp_2 - 2p - p_1 - p_2 - 1 &= 0, \\ p^2 - 4p - 2p_1 - 2p_2 - 2 &= 0. \end{aligned} \quad (32)$$

The nontrivial solution of this system is $p_1 = 2$, $p_2 = 3$, and $p = 6$. This completes the proof. $\square$

Using symbolic computations and Taylor expansions, it is easy to derive the following lemma.

Lemma 6. *Assuming (15) and (17), we have*

$$1 + \gamma_k f'(\alpha) \sim e_{k-1} e_{k-1,w} e_{k-1,y} e_{k-1,z}, \quad (33)$$

where $\gamma_k = -1/N'_4(x_k)$, $e_k = x_k - \alpha$, $e_{k,z} = z_k - \alpha$, $e_{k,y} = y_k - \alpha$, and $e_{k,w} = w_k - \alpha$.

Proof. The proof of this lemma is similar to Lemma 4. It is hence omitted. $\square$

Similarly for the three-step method with memory (20), we have the following theorem.

Theorem 7. *If an initial approximation x_0 is sufficiently close to the zero α of $f(x)$ and the parameter γ_k in the iterative scheme (20) is recursively calculated by the forms given in (15), then the order of convergence for (20) is at least twelve.*

Proof. Let $\{x_k\}$ be a sequence of approximations generated by the iterative method with memory (20). If this sequence converges to the zero α of f with the order p , then we write

$$e_{k+1} \sim e_k^p, \quad e_k = x_k - \alpha. \quad (34)$$

So

$$e_{k+1} \sim (e_{k-1}^p)^p = e_{k-1}^{p^2}. \quad (35)$$

Moreover, assume that the iterative sequences w_k , y_k , and z_k have the orders p_1 , p_2 , and p_3 , respectively. Then, (34) gives

$$e_{k,w} \sim e_k^{p_1} \sim (e_{k-1}^p)^{p_1} = e_{k-1}^{pp_1}, \quad (36)$$

$$e_{k,y} \sim e_k^{p_2} \sim (e_{k-1}^p)^{p_2} = e_{k-1}^{pp_2}, \quad (37)$$

$$e_{k,z} \sim e_k^{p_3} \sim (e_{k-1}^p)^{p_3} = e_{k-1}^{pp_3}. \quad (38)$$

Since

$$e_{k,w} \sim (1 + \gamma_k f'(\alpha)) e_k, \quad (39)$$

$$e_{k,y} \sim (1 + \gamma_k f'(\alpha)) e_k^2, \quad (40)$$

$$e_{k,z} \sim (1 + \gamma_k f'(\alpha))^2 e_k^4, \quad (41)$$

$$e_{k+1} \sim (1 + \gamma_k f'(\alpha))^4 e_k^8, \quad (42)$$

by Lemma 6 and (40), we obtain

$$\begin{aligned} e_{k,w} &\sim (1 + \gamma_k f'(\alpha)) e_k \sim (e_{k-1} e_{k-1,y} e_{k-1,w} e_{k-1,z}) e_k \\ &= e_{k-1}^{p+p_1+p_2+p_3+1}, \end{aligned} \quad (43)$$

$$\begin{aligned} e_{k,y} &\sim (1 + \gamma_k f'(\alpha)) e_k^2 \sim (e_{k-1} e_{k-1,y} e_{k-1,w} e_{k-1,z}) e_k^2 \\ &= e_{k-1}^{2p+p_1+p_2+p_3+1}, \end{aligned} \quad (44)$$

$$\begin{aligned} e_{k,z} &\sim (1 + \gamma_k f'(\alpha)) e_k^2 \sim (e_{k-1} e_{k-1,y} e_{k-1,w} e_{k-1,z})^2 e_k^4 \\ &= e_{k-1}^{4p+2p_1+2p_2+2p_3+2}, \end{aligned} \quad (45)$$

$$\begin{aligned} e_{k+1} &\sim (1 + \gamma_k f'(\alpha))^2 e_k^4 \sim (e_{k-1} e_{k-1,y} e_{k-1,w} e_{k-1,z})^4 e_k^8 \\ &= e_{k-1}^{8p+4p_1+4p_2+4p_3+4}. \end{aligned} \quad (46)$$

Matching the powers of e_{k-1} on the right hand sides of (36)–(43), (37)–(44), (38)–(45), and (35)–(46), one can obtain

$$\begin{aligned} pp_1 - p - p_1 - p_2 - p_3 - 1 &= 0, \\ pp_2 - 2p - p_1 - p_2 - p_3 - 1 &= 0, \\ pp_3 - 4p - 2p_1 - 2p_2 - 2p_3 - 2 &= 0, \\ p^2 - 8p - 4p_1 - 4p_2 - 4p_3 - 2 &= 0. \end{aligned} \quad (47)$$

This system has the solutions $p_1 = 2$, $p_2 = 3$, $p_3 = 6$, and $p = 12$. The proof is complete. $\square$

Remark 8. The advantage of the proposed methods is in their higher computational efficiency indices. We emphasize that the increase of the R-order of convergence has been obtained without any additional function evaluations, which points to very high computational efficiency. Indeed, the efficiency index $12^{1/4} \approx 1.861$ of the proposed three-step twelfth-order method with memory is higher than the efficiency index, $6^{1/3} \approx 1.817$ of (19), $8^{1/4} \approx 1.682$ of the optimal three-point method (11), and $4^{1/3} \approx 1.587$ of (6).

Remark 9. We observe that the methods (19) and (20) with memory are considerably accelerated (up to 50%) in contrast to the corresponding method (11) without memory.

4. Numerical Experiments

In this section, we test our proposed methods and compare their results with some other methods of the same order of

convergence. The results are reported using the programming package Mathematica 8 in multiple precision arithmetic environment. We have considered 1000 digits floating point arithmetic so as to minimize the round-off errors as much as possible. The errors $|x_k - \alpha|$ denote approximations to the sought zeros, and $a(-b)$ stands for $a \times 10^{-b}$. Moreover, coc indicates the computational order of convergence and is computed by

$$\text{coc} = \frac{\log(|f(x_k)/f(x_{k-1})|)}{\log(|f(x_{k-1})/f(x_{k-2})|)}. \quad (48)$$

It is assumed that the initial estimate γ_0 should be chosen before starting the iterative process, and also x_0 is given suitably.

Several iterative methods of optimal orders four and eight, for comparing with our proposed methods, have been chosen as comes next.

Derivative-free Kung-Traub's two-step method (KT4) [6] is as follows:

$$y_k = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \gamma f(x_k), \quad (49)$$

$$x_{k+1} = y_k - \frac{f(y_k) f(w_k)}{[f(w_k) - f(y_k)] f[x_k, y_k]}.$$

Two-step method by Zheng et al. (ZLH4) [10] is as follows:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \gamma f(x_k), \\ x_{k+1} &= y_k - \frac{f(y_k)}{f[y_k, w_k] + f[y_k, x_k, w_k] (y_k - x_k)}. \end{aligned} \quad (50)$$

Two-step method by Lotfi and Tavakoli (LT4) [11] is as follows:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \\ x_{k+1} &= y_k - H(t_k, u_k) \frac{f(y_k)}{f[y_k, w_k]}, \\ t_k &= \frac{f(y_k)}{f(x_k)}, \quad u_k = \frac{f(w_k)}{f(x_k)}, \end{aligned} \quad (51)$$

where $H(t_k, u_k) = 1 + t_k$.

TABLE 1: $f_1(x) = \sin(\pi x)e^{(x^2+x\cos(x)-1)} + x \log(x \sin(x) + 1)$, $\alpha = 0$, and $x_0 = 0.6$.

Methods	$ x_1 - \alpha $	$ x_2 - \alpha $	$ x_3 - \alpha $	coc
Without memory				
New method (6), $\gamma = -1$	0.6742(-2)	0.1587(-10)	0.4307(-45)	4.0049
KT4, $\gamma = -0.01$	0.6012(-1)	0.1486(-4)	0.6094(-19)	4.0171
ZLH4, $\gamma = -0.01$	0.5882(-1)	0.1320(-5)	0.8291(-24)	3.9365
LT, $\gamma = -0.01$	0.6178(-1)	0.3334(-4)	0.2739(-17)	4.0368
With memory				
New method (19), $\gamma = -1$	0.6742(-2)	0.3177(-11)	0.8372(-62)	5.4213
KT4, $\gamma = -1$	0.2102(-1)	0.9935(-9)	0.2326(-48)	5.4031
ZLH4, $\gamma = -0.01$	0.5882(-1)	0.2204(-8)	0.1503(-45)	5.0216
LT4, $\gamma = -1$	0.1752(-1)	0.4934(-9)	0.5220(-50)	5.4214

Derivative-free Kung-Traub's three-step method (KT8) [6] is as follows:

$$\begin{aligned}
 y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \gamma f(x_k), \\
 z_k &= y_k - \frac{f(y_k) f(w_k)}{[f(w_k) - f(y_k)] f[x_k, y_k]}, \\
 x_{k+1} &= z_k - \frac{f(y_k) f(w_k) (y_k - x_k + f(x_k) / f[x_k, z_k])}{[f(y_k) - f(z_k)] [f(w_k) - f(z_k)]} \\
 &\quad + \frac{f(y_k)}{f[y_k, z_k]}.
 \end{aligned} \tag{52}$$

Three-step methods developed by Zheng et al. (ZLH8) [10] is as follows:

$$\begin{aligned}
 y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \gamma f(x_k), \\
 z_k &= y_k - \frac{f(y_k)}{f[y_k, w_k] + f[y_k, x_k, w_k] (y_k - x_k)}, \\
 x_{k+1} &= z_k \\
 &\quad - \left((f(z_k)) \right. \\
 &\quad \times (f[z_k, y_k] + f[z_k, y_k, x_k] (z_k - y_k) \\
 &\quad \left. + f[z_k, y_k, x_k, w_k] (z_k - y_k) (z_k - x_k)^{-1} \right).
 \end{aligned} \tag{53}$$

Three-step method by Lotfi and Tavakoli (LT8) [11] is as follows:

$$\begin{aligned}
 y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \\
 z_k &= y_k - H(t_k, u_k) \frac{f(y_k)}{f[y_k, w_k]}, \\
 t_k &= \frac{f(y_k)}{f(x_k)}, \quad u_k = \frac{f(w_k)}{f(x_k)},
 \end{aligned}$$

$$\begin{aligned}
 x_{k+1} &= z_k - G(t_k, s_k) W(v_k, s_k) \frac{f(z_k)}{f[z_k, w_k]}, \\
 s_k &= \frac{f(z_k)}{f(y_k)}, \quad v_k = \frac{f(z_k)}{f(x_k)},
 \end{aligned} \tag{54}$$

where $W(s_k, v_k) = 1 + s_k^2 + v_k^2$, $G(t_k, s_k) = 1 + t_k + s_k + 2t_k s_k + (-1 - \phi_k)t_k^3$ and $(\phi_k = 1/(1 + \gamma_k f[x_k, w_k]))$, and $H(t_k, u_k) = 1 + t_k$ are the weight functions.

In Tables 1, 2, and 3 our two-step proposed classes (6) and (19) have been compared with optimal two-step methods KT4, ZLH4, and LT4. We observe that all these methods behave very well practically and confirm their theoretical results.

Also Tables 4, 5, and 6 present numerical results for our three-step classes (11) and (20) and methods KT8, ZLH8, and LT8. It is also clear that all these methods behave very well practically and confirm their relevant theories.

We remark the importance of the choice of initial guesses. If they are chosen sufficiently close to the sought roots, then the expected (theoretical) convergence speed will be reached in practice; otherwise, all iterative root-finding methods show slower convergence, especially at the beginning of the iterative process. Hence, a special attention should be paid to finding good initial approximations. We note that efficient ways for the determination of initial approximations of great accuracy were discussed thoroughly in the works [12–14].

5. Conclusions

We have constructed two families of iterative methods without memory which are optimal in the sense of Kung and Traub's conjecture in this paper. Our proposed methods do not need any derivative.

In addition, they contain an accelerator parameter which raises convergence order without any new functional evaluations. In other words, the efficiency index of the three-step with memory hit $12^{1/4} \approx 1.861$.

We finalize this work by suggesting some points for future researches: first developing the proposed methods for some matrix functions, such as the ones in [15, 16], second

TABLE 2: $f_2(x) = e^{-5x}(x-2)(x^{10} + x + 2)$, $\alpha = 2$, and $x_0 = 2.2$.

Methods	$ x_1 - \alpha $	$ x_2 - \alpha $	$ x_3 - \alpha $	coc
Without memory				
New method (6), $\gamma = -0.01$	0.9445(-3)	0.1550(-13)	0.1289(-56)	3.9945
KT4, $\gamma = -0.01$	0.1125(-2)	0.3033(-13)	0.1891(-55)	3.9932
ZLH4, $\gamma = -0.1$	0.9349(-3)	0.1478(-13)	0.1074(-56)	3.9939
LT, $\gamma = -0.01$	0.1343(-2)	0.5965(-13)	0.2821(-54)	3.9918
With memory				
New method (19), $\gamma = 1$	0.1057(-2)	0.1102(-19)	0.8335(-109)	5.2480
KT4, $\gamma = 1$	0.1267(-2)	0.2180(-19)	0.3580(-107)	5.2364
ZLH4, $\gamma = -0.01$	0.9445(-3)	0.2868(-20)	0.8014(-112)	5.2264
LT4, $\gamma = -1$	0.1343(-2)	0.1072(-19)	0.1150(-108)	5.2035

TABLE 3: $f_3(x) = e^{x^3-x} - \cos(x^2 - 1) + x^3 + 1$, $\alpha = -1$, and $x_0 = -1.65$.

Methods	$ x_1 - \alpha $	$ x_2 - \alpha $	$ x_3 - \alpha $	coc
Without memory				
New method (6), $\gamma = -0.01$	0.6849(-3)	0.8725(-13)	0.2284(-52)	4.0003
KT4, $\gamma = -0.01$	0.6610(-3)	0.8682(-13)	0.2564(-52)	4.0004
ZLH4, $\gamma = 0.1$	0.1214(-2)	0.2388(-12)	0.3560(-51)	4.0004
LT4, $\gamma = -0.01$	0.6398(-3)	0.8549(-13)	0.2714(-52)	4.0002
With memory				
New method (19), $\gamma = -1$	0.4838(-2)	0.1348(-11)	0.1230(-64)	5.5506
KT4, $\gamma = 1$	0.1170(-1)	0.3517(-7)	0.8936(-41)	6.0860
ZLH4, $\gamma = -0.01$	0.6849(-3)	0.7630(-17)	0.1662(-92)	5.4226
LT4, $\gamma = -0.01$	0.6398(-3)	0.6666(-17)	0.7373(-93)	5.4324

TABLE 4: $f_1(x) = \sin(\pi x)e^{(x^2+x\cos(x)-1)} + x \log(x \sin(x) + 1)$, $\alpha = 0$, and $x_0 = 0.6$.

Methods	$ x_1 - \alpha $	$ x_2 - \alpha $	$ x_3 - \alpha $	coc
Without memory				
New method (11), $\gamma = -1$	0.2345(-3)	0.1042(-32)	0.1593(-267)	7.9999
KT8, $\gamma = -1$	0.3101(-3)	0.2671(-31)	0.8196(-256)	7.9999
ZLH8, $\gamma = -0.01$	0.1742(-2)	0.1558(-22)	0.6471(-183)	7.9999
LT8, $\gamma = -1$	0.5758(-3)	0.7106(-29)	0.3880(-236)	7.9998
With memory				
New method (20), $\gamma = -0.1$	0.3440(-3)	0.1220(-41)	0.8636(-503)	11.9935
KT8, $\gamma = -0.01$	0.9773(-3)	0.2960(-33)	0.1505(-399)	12.0021
ZLH8, $\gamma = -0.01$	0.1742(-2)	0.3012(-33)	0.4438(-402)	11.9900
LT8, $\gamma = -0.1$	0.7107(-4)	0.2040(-49)	0.4971(-596)	12.0024

TABLE 5: $f_2(x) = e^{-5x}(x-2)(x^{10} + x + 2)$, $\alpha = 2$, and $x_0 = 2.2$.

Methods	$ x_1 - \alpha $	$ x_2 - \alpha $	$ x_3 - \alpha $	coc
Without memory				
New method (11), $\gamma = -1$	0.3124(-6)	0.9972(-56)	0.1074(-451)	8.0000
KT8, $\gamma = -1$	0.8055(-6)	0.1413(-52)	0.1268(-426)	8.0000
ZLH8, $\gamma = 1$	0.4955(-6)	0.5798(-54)	0.2038(-437)	8.0000
LT8, $\gamma = -1$	0.4052(-6)	0.5819(-55)	0.1053(-445)	8.0000
With memory				
New method (20), $\gamma = -1$	0.3124(-6)	0.1213(-81)	0.3093(-985)	11.9822
KT8, $\gamma = -1$	0.8055(-6)	0.2641(-78)	0.9143(-945)	11.9538
ZLH8, $\gamma = -0.01$	0.3943(-6)	0.1889(-80)	0.6308(-971)	11.9817
LT8, $\gamma = -0.01$	0.4532(-6)	0.4774(-76)	0.5848(-853)	11.1023

TABLE 6: $f_3(x) = e^{x^3-x} - \cos(x^2 - 1) + x^3 + 1$, $\alpha = -1$, and $x_0 = -1.65$.

Methods	$ x_1 - \alpha $	$ x_2 - \alpha $	$ x_3 - \alpha $	coc
Without memory				
New method (11), $\gamma = -1$	0.2768(-3)	0.3670(-27)	0.3499(-218)	8.0000
KT8, $\gamma = -1$	0.2915(-4)	0.5324(-34)	0.6583(-272)	8.0000
ZLH8, $\gamma = 1$	0.2322(-1)	0.5044(-11)	0.2253(-88)	8.0081
LT, $\gamma = -1$	0.1094(-2)	0.1454(-22)	0.1323(-181)	8.0014
With memory				
New method (20), $\gamma = -1$	0.2768(-3)	0.2660(-42)	0.1805(-509)	11.9734
KT8, $\gamma = -1$	0.2915(-4)	0.5214(-53)	0.1157(-642)	12.0961
ZLH8, $\gamma = -0.01$	0.6052(-5)	0.2355(-65)	0.4206(-786)	11.9310
LT8, $\gamma = -1$	0.1094(-2)	0.3522(-34)	0.1717(-415)	12.1081

exploring its dynamic or basins of attractions, and lastly extending the developed methods with memory using two or three accelerators.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The first author thanks Islamic Azad University, Hamedan Branch, for the financial support of the present research. The authors are also thankful for insightful comments of three referees whom helped with the readability and reliability of the present paper. The fourth author also gratefully acknowledges that this research was partially supported by the University Putra Malaysia under the GP-IBT Grant Scheme having project number GP-IBT/2013/9420100.

References

- [1] N. Kyurkchiev and A. Iliev, "On some multipoint methods arising from optimal in the sense of Kung-Traub algorithms," *Biomath*, vol. 2, no. 1, Article ID 1305155, 7 pages, 2013.
- [2] T. Lotfi, F. Soleymani, S. Sharifi, S. Shateyi, and F. K. Haghani, "Multi-point iterative methods for finding all the simple zeros in an interval," *Journal of Applied Mathematics*, vol. 2014, Article ID 601205, 14 pages, 2014.
- [3] J. F. Traub, G. W. Wasilkowski, and H. Wozniakowski, *Information, Uncertainty, Complexity*, Addison-Wesley, 1983.
- [4] M. S. Rhee and Y. I. Kim, "A general class of optimal fourth-order multiple-root finders without memory for nonlinear equations," *Applied Mathematical Sciences*, vol. 7, pp. 5537–5551, 2013.
- [5] H. Wozniakowski, "Maximal order of multipoint iterations using n evaluations," in *Analytic Computational Complexity*, J. F. Traub, Ed., pp. 75–107, Academic Press, New York, NY, USA, 1976.
- [6] H. T. Kung and J. F. Traub, "Optimal order of one-point and multipoint iteration," *Journal of the Association for Computing Machinery*, vol. 21, pp. 643–651, 1974.
- [7] Y. H. Geum and Y. I. Kim, "An optimal family of fast 16th-order derivative-free multipoint simple-root finders for nonlinear equations," *Journal of Optimization Theory and Applications*, vol. 160, no. 2, pp. 608–622, 2014.
- [8] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, New York, NY, USA, 1964.
- [9] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, 1970.
- [10] Q. Zheng, J. Li, and F. Huang, "An optimal Steffensen-type family for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 217, no. 23, pp. 9592–9597, 2011.
- [11] T. Lotfi and E. Tavakoli, "On a new efficient Steffensen-like iterative class by applying a suitable self-accelerator parameter," *The Scientific World Journal*, vol. 2014, Article ID 769758, 9 pages, 2014.
- [12] F. Soleymani and S. Shateyi, "Two optimal eighth-order derivative-free classes of iterative methods," *Abstract and Applied Analysis*, vol. 2012, Article ID 318165, 14 pages, 2012.
- [13] F. Soleymani and D. K. R. Babajee, "Computing multiple zeros using a class of quartically convergent methods," *Alexandria Engineering Journal*, vol. 52, no. 3, pp. 531–541, 2013.
- [14] F. Soleymani, "Some high-order iterative methods for finding all the real zeros," *Thai Journal of Mathematics*, vol. 12, pp. 313–327, 2014.
- [15] F. Soleymani, E. Tohidi, S. Shateyi, and F. K. Haghani, "Some matrix iterations for computing matrix sign function," *Journal of Applied Mathematics*, vol. 2014, Article ID 425654, 9 pages, 2014.
- [16] F. Soleymani, P. S. Stanimirović, S. Shateyi, and F. K. Haghani, "Approximating the matrix sign function using a novel iterative method," *Abstract and Applied Analysis*, vol. 2014, Article ID 105301, 9 pages, 2014.