Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 497023, 24 pages
doi:10.1155/2012/497023

Research Article

Construction of Optimal Derivative-Free Techniques without Memory

F. Soleymani,[1] D. K. R. Babajee,[2] S. Shateyi,[3] and S. S. Motsa[4]

[1] *Department of Mathematics, Islamic Azad University, Zahedan Branch, Zahedan, Iran*
[2] *Allied Network for Policy Research and Advocacy for Sustainability, IEEE, Mauritius, Mauritius*
[3] *Department of Mathematics, University of Venda, Private Bag X5050, Thohoyandou 0950, South Africa*
[4] *School of Mathematical Sciences, University of KwaZulu-Natal, Private Bag X01, Pietermaritzburg, South Africa*

Correspondence should be addressed to S. Shateyi, stanford.shateyi@univen.ac.za

Received 5 July 2012; Accepted 3 September 2012

Academic Editor: Alicia Cordero

Construction of iterative processes without memory, which are both optimal according to the Kung-Traub hypothesis and derivative-free, is considered in this paper. For this reason, techniques with four and five function evaluations per iteration, which reach to the optimal orders eight and sixteen, respectively, are discussed theoretically. These schemes can be viewed as the generalizations of the recent optimal derivative-free family of Zheng et al. in (2011). This procedure also provides an n-step family using $n+1$ function evaluations per full cycle to possess the optimal order 2^n. The analytical proofs of the main contributions are given and numerical examples are included to confirm the outstanding convergence speed of the presented iterative methods using only few function evaluations. The second aim of this work will be furnished when a hybrid algorithm for capturing all the zeros in an interval has been proposed. The novel algorithm could deal with nonlinear functions having finitely many zeros in an interval.

1. Introduction

The purpose of this study is to present some generalizations of both the celebrated second-order Steffensen's method [1], which was advanced by the Danish mathematician Johan Frederik Steffensen (1873–1961) as follows

$$x_{n+1} = x_n - \left(f\left(x_n + f(x_n)\right) - f(x_n)\right)^{-1} f(x_n)^2, \quad n = 0, 1, 2, \dots, \tag{1.1}$$

with higher orders of convergence and optimal efficiency indices, and also the extension of the family of methods in [2]. In 1974, Kung and Traub [3] conjectured on the optimality of multipoint iterative schemes without memory as comes next. An iterative method for solving single variable nonlinear equation $f(x) = 0$, with $n + 1$, $n \geq 1$, evaluations per iteration reaches to the maximum order of convergence 2^n and the optimal efficiency index $2^{n/(n+1)}$. Consequently, the efficiency of a pth-order method could be given by $p^{1/d}$, where d is the whole number of (functional) evaluations per iteration. Note that Steffensen's method possesses 1.414 as its efficiency index. By considering this conjecture, per iteration the contributed techniques in this research should reach the order 8 with four and order 16 with five evaluations.

The paper unfolds the contents as follows. A collection of pointers to known literature on derivative-free techniques will be presented in Section 2. This is followed by Section 3, whereas the central results are given and also by Section 4, where an optimal sixteenth-order derivative-free technique is constructed. Results and discussion on the comparisons with other famous derivative-free methods are presented in Section 5, entitled by Computational Tests. The only difficulty of iterative methods of this type is in the choice of the initial guess. Using the programming package MATHEMATICA 8, we will provide an algorithm to capture all the real solutions of a nonlinear equation in a short piece of time by presenting a hybrid algorithm in Section 6. The conclusions have finally been drawn in Section 7.

2. Selections from the Literature

The literature related to the present paper is substantial, and we do not present a comprehensive survey. The references of the papers we cite should be consulted for further reading. Consider iterative techniques for finding a simple root of the nonlinear equation $f(x) = 0$, where $f : D \subseteq \mathbb{R} \to \mathbb{R}$ for an open interval $D = (a, b)$ is a scalar function and it is sufficiently smooth in a neighborhood of $\alpha \in D$. The design of formulas for solving such equations is an absorbing task in numerical analysis [4]. The most frequent approach to solve the nonlinear equations consists of the implementation of rapidly convergent iterative methods starting from a reasonably good initial guess to the sought zero of $f(x)$.

Many iterative methods have been improved by using various techniques such as quadrature formulas and weight function; see, for example, [5, 6]. All these developments are aimed at increasing the local order of convergence with a special view of increasing their efficiency indices. The derivative-involved methods are well discussed in the literature; see, for example, [7–9] and the references therein. But another thing that should be mentioned is that for many particular choices of the function f, specially in hard problems, the calculations of the derivatives are not possible or it takes a deal of time.

Thus, in some situations the considered function $f(x)$ has an improper behavior or its derivative is close to 0, which causes that the applied iterative processes fail. That is why higher-order derivative-free methods are better root solvers and are in focus recently. The most important merit of Steffensen's method is that it has quadratic convergence like Newton's method. That is, both techniques estimate roots of the equation $f(x) = 0$ just as quickly. In this case, quickly means that the number of correct digits in the new obtained value doubles with each iteration for both. But the formula for Newton's method requires a separate function for the derivative; Steffensen's method does not. Now let us review some derivative-free techniques.

In 2010, an optimal fourth-order derivative-free method [10] was introduced by Liu et al. in the following form:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)^2}{f(x_n+f(x_n))-f(x_n)}, \\ x_{n+1} &= y_n - \frac{f[x_n,y_n]-f[y_n,z_n]+f[x_n,z_n]}{f[x_n,y_n]^2} f(y_n), \end{aligned} \tag{2.1}$$

where $z_n = x_n + f(x_n)$. This technique consists of three evaluations of the function per iteration to obtain the fourth-order convergence. We here remark that $f[x_n,y_n]$, $f[y_n,z_n]$, and $f[x_n,z_n]$ are divided differences. This scheme has $4^{1/3} \approx 1.587$ as its efficiency index. The notation of divided differences will be used throughout this paper.

In 2011, Cordero et al. [11] proposed a sixth-order method, which is free from derivative

$$\begin{aligned} y_n &= x_n - \frac{2f(x_n)^2}{f(x_n+f(x_n))-f(x_n-f(x_n))}, \\ z_n &= y_n - \frac{y_n-x_n}{2f(y_n)-f(x_n)} f(y_n), \\ x_{n+1} &= z_n - \frac{y_n-x_n}{2f(y_n)-f(x_n)} f(z_n), \end{aligned} \tag{2.2}$$

and includes 5 evaluations of the function per iteration to reach the efficiency index $6^{1/5} \approx 1.495$.

Zheng et al. in [2] presented the following eighth-order derivative-free family without memory:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n,w_n]}, \qquad w_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\ z_n &= y_n - \frac{f(y_n)}{f[x_n,y_n]+f[y_n,w_n]-f[x_n,w_n]}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f[z_n,y_n]+f[z_n,y_n,x_n](z_n-y_n)+f[z_n,y_n,x_n,w_n](z_n-y_n)(z_n-x_n)}, \end{aligned} \tag{2.3}$$

which is optimal in the sense of Kung-Traub.

Soleymani [12] proposed the following optimal three-step iteration without memory, including four function evaluations, just like (2.3), by using the weight function approach:

$$
\begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f[x_n, w_n]}, \qquad w_n = x_n + f(x_n), \\
z_n &= y_n - \frac{f(y_n)}{f[x_n, w_n]}\left(1 + \frac{f(y_n)}{f(x_n)} + \frac{f(y_n)}{f(w_n)}\right), \\
x_{n+1} &= z_n - \frac{f(z_n)}{f[x_n, y_n]}\left(1 + \frac{f(z_n)}{f(x_n)} + \frac{f(z_n)}{f(y_n)} + 2\frac{f(z_n)}{f(w_n)} - \left(\frac{f(y_n)}{f(x_n)}\right)^2 + \frac{f(y_n)}{f(w_n)}\right. \\
&\qquad \left. -\frac{1}{6}(30 + 6f[x_n, w_n](8 + f[x_n, w_n](5 + f[x_n, w_n])))\left(\frac{f(y_n)}{f(w_n)}\right)^3\right).
\end{aligned}
\tag{2.4}
$$

To see a recent paper including derivative-free methods with memory, refer to [13]. For further reading on this topic, refer to [14–19].

3. Construction of a New Eighth-Order Derivative-Free Class

In this paper, we derive a new way for constructing multistep methods of orders eight, sixteen, and so forth, requiring four, five, and so forth respectively, function evaluations per iteration. This means that the proposed techniques support the Kung-Traub hypothesis. Besides, the novel schemes do not use any derivative of a function f whose zeros are sought, which is another advantage since it is preferable to avoid calculations of derivatives of f in some situations. Let us consider a two-step cycle in which we have Steffensen's method in the first step and Newton's method in the second step as follows:

$$
\begin{aligned}
y_n &= x_n - \frac{f(x_n)^2}{f(x_n + f(x_n)) - f(x_n)}, \\
x_{n+1} &= y_n - \frac{f(y_n)}{f'(y_n)},
\end{aligned}
\tag{3.1}
$$

with four evaluations, that is, three evaluations of the function $f(x_n)$, $f(x_n+f(x_n))$, and $f(y_n)$ and one derivative evaluation $f'(y_n)$ per iteration. For simplicity, we assume that $f(x_n + f(x_n)) = f(k_n)$; that is, $x_n + f(x_n) = k_n$. At this time, the main challenge is to approximate $f'(y_n)$ as efficiently as possible such that the fourth-order convergence does not decrease and the efficiency index increases to 1.587 at the same time. To fulfill this aim, we must use all of the past three known data, that is, $f(x_n)$, $f(k_n)$, and $f(y_n)$. Now take into account the following approximation function for $f(t)$ in the domain D [20]:

$$
f(t) \approx w(t) = a_0 + a_1(t - x_n) + a_2(t - x_n)^2 + a_3(t - x_n)^3,
\tag{3.2}
$$

where its first derivative takes the form $w'(t) = a_1 + 2a_2(t - x_n) + 3a_3(t - x_n)^2$. Clearly, the unknown three parameters a_0, a_1, and a_2 will be obtained by substituting of the known values in (3.2). Note that a_3 is a free real parameter. Hence, we have

$$\begin{gathered}
a_0 = f(x_n), \\
a_2 = \frac{f[k_n, x_n] - f[y_n, x_n]}{k_n - y_n} - a_3(x_n + y_n + z_n), \\
a_1 = f[y_n, x_n] - (y_n - x_n)a_2 + a_3(x_n y_n + x_n z_n + y_n z_n).
\end{gathered} \tag{3.3}$$

Accordingly, by considering a_1, a_2 in the derivative form of (3.2), we attain the following two-step derivative-free technique:

$$\begin{gathered}
y_n = x_n - \frac{f(x_n)^2}{f(k_n) - f(x_n)}, \qquad k_n = x_n + f(x_n), \\
x_{n+1} = y_n - \frac{f(y_n)}{a_1 + 2a_2(y_n - x_n) + 3a_3(y_n - x_n)^2}, \quad a_3 \in \mathbb{R},
\end{gathered} \tag{3.4}$$

wherein there are three evaluations of the function per iteration only. Theorem 3.1 indicates that the order of (3.4) is four, and hence, it is an optimal derivative-free class with the efficiency index 1.587. Note that (3.4) is similar to the method given by Ren et al. in [20].

Theorem 3.1. *Assume $f(x)$ to be a sufficiently continuous real function in the domain D. Then the sequence generated by* (3.4) *converges to the simple root $\alpha \in D$ with fourth-order convergence and it satisfies the follow-up error equation*

$$e_{n+1} = \frac{c_2(c_2^2 - c_1(c_3 + a_3))(1 + c_1)^2}{c_1^3} e_n^4 + O\left(e_n^5\right), \tag{3.5}$$

wherein $c_j = (f^{(j)}(\alpha))/j!$, $j \geq 1$, and $e_n = x_n - \alpha$.

Proof. See [20]. □

The given approximation (3.3) of the first derivative of the function in the second step of our cycle can be applied on any optimal second-order derivative-free method to provide a new optimal fourth-order method. For example, if one chooses a two-step cycle in which the first derivative of the function in the first step estimated by backward finite difference, and after that applies the presented approximation in the second step, then another novel optimal fourth-order method could be obtained. This will be discussed more in the rest of the work.

Remark 3.2. The simplified form of the approximation for the derivative in the quotient of Newton's iteration at the second step of our cycle is as follows:

$$\begin{aligned}
&a_1 + 2a_2(y_n - x_n) + 3a_3(y_n - x_n)^2 \\
&\quad = f[y_n, x_n] + \frac{(f[k_n, x_n] - f[y_n, x_n])(-x_n + y_n)}{k_n - y_n} + 3a_3(y_n - x_n)^2 \\
&\quad = f[y_n, x_n] + f[k_n, x_n, y_n](y_n - x_n) + a_3(y_n - x_n)(y_n - k_n).
\end{aligned} \tag{3.6}$$

Inspired by the above approach, now we are about to construct *new* high-order derivative-free methods, which are optimal as well and can be considered as the generalizations of the methods in [2, 20, 21]. To construct such techniques, first we should consider an optimal fourth-order method in the first and second steps of a multistep cycle. Toward this end, we take into consideration the following three-step cycle in which Newton's method is applied in the third step:

$$\begin{gathered}
y_n = x_n - \frac{\beta f(x_n)^2}{f(k_n) - f(x_n)}, \qquad k_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\
z_n = y_n - \frac{f(y_n)}{f[y_n, x_n] + f[k_n, x_n, y_n](y_n - x_n) + a_3(y_n - x_n)(y_n - k_n)}, \quad a_3 \in \mathbb{R}, \\
x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}.
\end{gathered} \tag{3.7}$$

It is crystal clear that scheme (3.7) is an eighth-order method with five evaluations per iteration. The main question is that Is there any way to keep the order on eight but reduce the number of evaluation while the method be free from derivative? Fortunately, by using all four past known data, a very powerful approximation of $f'(z_n)$ will be obtained. Therefore, we approximate $f(t)$ in the domain D, by a new polynomial approximation of degree four with a free parameter b_4, as follows:

$$f(t) \approx l(t) = b_0 + b_1(t - x_n) + b_2(t - x_n)^2 + b_3(t - x_n)^3 + b_4(t - x_n)^4, \tag{3.8}$$

where its first derivative takes the following form:

$$f'(t) \approx l'(t) = b_1 + 2b_2(t - x_n) + 3b_3(t - x_n)^2 + 4b_4(t - x_n)^3. \tag{3.9}$$

Hopefully, we have four known values $f(x_n), f(k_n), f(y_n)$, and $f(z_n)$. Thus, by substituting these values in (3.8), we obtain the following linear system of four equations with four unknowns:

$$\begin{aligned}
f(x_n) &= b_0, \\
f(k_n) &= f(x_n) + b_1(k_n - x_n) + b_2(k_n - x_n)^2 + b_3(k_n - x_n)^3 + b_4(k_n - x_n)^4, \\
f(y_n) &= f(x_n) + b_1(y_n - x_n) + b_2(y_n - x_n)^2 + b_3(y_n - x_n)^3 + b_4(y_n - x_n)^4, \\
f(z_n) &= f(x_n) + b_1(z_n - x_n) + b_2(z_n - x_n)^2 + b_3(z_n - x_n)^3 + b_4(z_n - x_n)^4,
\end{aligned} \tag{3.10}$$

and consequently we can find the unknown parameters by solving linear system (3.10). We attain

$$\begin{aligned}
b_0 &= f(x_n), \\
b_3 &= \frac{(f[k_n, x_n] - f[z_n, x_n])/(k_n - z_n) - (f[y_n, x_n] - f[z_n, x_n])/(y_n - z_n)}{k_n - y_n} \\
&\quad - b_4(x_n + y_n + k_n + z_n) \\
&= \frac{f[z_n, x_n, k_n] - f[z_n, x_n, y_n]}{k_n - y_n} - b_4(x_n + y_n + k_n + z_n), \\
b_2 &= f[y_n, x_n, k_n] - b_3(k_n - 2x_n + y_n) \\
&\quad + b_4(x_n y_n + x_n k_n + x_n z_n + y_n k_n + y_n z_n + k_n z_n), \\
b_1 &= f[z_n, x_n] - (z_n - x_n)^2 b_3 - (z_n - x_n) b_2.
\end{aligned} \tag{3.11}$$

Now, an efficient, accurate, and optimal eighth-order family, which is free from any derivative, can be obtained in the following form:

$$\begin{aligned}
y_n &= x_n - \frac{\beta(f(x_n))^2}{f(k_n) - f(x_n)}, \qquad k_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\
z_n &= y_n - \frac{f(y_n)}{f[y_n, x_n] + f[k_n, x_n, y_n](y_n - x_n) + a_3(y_n - x_n)(y_n - k_n)}, \quad a_3 \in \mathbb{R}, \\
x_{n+1} &= z_n - \frac{f(z_n)}{b_1 + 2b_2(z_n - x_n) + 3b_3(z_n - x_n)^2 + 4b_4(z_n - x_n)^3}, \quad b_4 \in \mathbb{R}.
\end{aligned} \tag{3.12}$$

Remark 3.3. The simplified form of the approximation for the derivative in the quotient of Newton's iteration at the third step of our cycle is as comes next:

$$\begin{aligned}
&b_1 + 2b_2(z_n - x_n) + 3b_3(z_n - x_n)^2 + 4b_4(z_n - x_n)^3 \\
&\quad = f[x_n, z_n] + (f[k_n, x_n, y_n] - f[k_n, x_n, z_n] - f[y_n, x_n, z_n])(x_n - z_n) \\
&\qquad + b_4(z_n - x_n)(z_n - k_n)(z_n - y_n).
\end{aligned} \tag{3.13}$$

Using Remark 3.3, we can propose the following optimal eighth-order derivative-free iteration in this paper with some free parameters:

$$\begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f[k_n, x_n]}, \qquad k_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\
z_n &= y_n - \frac{f(y_n)}{f[y_n, x_n] + f[k_n, x_n, y_n](y_n - x_n) + a_3(y_n - x_n)(y_n - k_n)}, \quad a_3 \in \mathbb{R}, \\
x_{n+1} &\\
&= z_n - \frac{f(z_n)}{f[x_n, z_n] + (f[k_n, x_n, y_n] - f[k_n, x_n, z_n] - f[y_n, x_n, z_n])(x_n - z_n) + b_4(z_n - x_n)(z_n - k_n)(z_n - y_n)},
\end{aligned} \tag{3.14}$$

where $b_4 \in \mathbb{R}$ and it consists of four evaluations per iteration to reach the efficiency index $8^{1/4} \approx 1.682$. Theorem 3.4 illustrates its error equation and order of convergence. The main contribution of this section lies in the following Theorem.

Theorem 3.4. *Assume that the function $f : D \subseteq \mathbb{R} \to \mathbb{R}$ has a single root $\alpha \in D$, where D is an open interval. Then the convergence order of the derivative-free iterative method defined by* (3.14) *is eight and it satisfies the following the error equation:*

$$e_{n+1} = \left(\frac{1}{c_1^7}\right)\left(c_2^2\left(c_2^2 - c_1(c_3 + a_3)\right)(1 + c_1\beta)^4\left(c_2^3 - c_1c_2(c_3 + a_3) + c_1^2(c_4 + b_4)\right)\right)e_n^8 + O\left(e_n^9\right). \tag{3.15}$$

Proof. To find the asymptotic error constant of (3.14), wherein $c_j = f^{(j)}(\alpha)/j!$, $j \geq 1$, and $e_n = x_n - \alpha$, we expand any terms of (3.14) around the simple root α in the nth iterate. Thus, we write $f(x_n) = c_1e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + c_6e_n^6 + c_7e_n^7 + c_8e_n^8 + O(e_n^9)$. Accordingly, we attain

$$y_n = \alpha + c_2\left(\frac{1}{c_1} + \beta\right)e_n^2 + \cdots + O\left(e_n^9\right). \tag{3.16}$$

Now we should expand $f(y_n)$ around the simple root by using (3.16). We have

$$f(y_n) = (c_2 + c_1c_2\beta)e_n^2 + \cdots + O\left(e_n^9\right). \tag{3.17}$$

Using (3.17) and the second step of (3.14), we attain

$$\frac{f(y_n)}{f[y_n,x_n]+f[k_n,x_n,y_n](y_n-x_n)+a_3(y_n-x_n)(y_n-k_n)}=c_2\left(\frac{1}{c_1}+\beta\right)e_n^2+\cdots+O\left(e_n^9\right). \tag{3.18}$$

Now, the Taylor expansion of the second step of (3.14), using (3.18), gives us

$$z_n-\alpha=\frac{c_2\left(c_2^2-c_1(c_3+a_3)\right)(1+c_1\beta)^2}{c_1^3}e_n^4+\cdots+O\left(e_n^9\right). \tag{3.19}$$

A similar Taylor expansion is required for continuing. Hence, it is needed to write the Taylor expansion of $f(z_n)$ now. Thus, we have

$$f(z_n)=\left(\frac{\left(c_2\left(c_2^2-c_1(c_3+a_3)\right)(1+c_1\beta)^2\right)}{c_1^2}\right)e_n^4+\ldots+O\left(e_n^9\right). \tag{3.20}$$

Subsequently for the approximation of $f'(z_n)$, we obtain $f[x_n,z_n]+(f[k_n,x_n,y_n]-f[k_n,x_n,z_n]-f[y_n,x_n,z_n])(x_n-z_n)+b_4(z_n-x_n)(z_n-k_n)(z_n-y_n)=c_1+(c_2(2c_2^3+c_1^2c_4-2c_1c_2(c_3+a_3))(1+c_1\beta)^2e_n^4)/c_1^3+\cdots+O(e_n^9)$. Furthermore, we attain

$$\begin{aligned}&\frac{f(z_n)}{\left(f[x_n,z_n]+\left(f[k_n,x_n,y_n]-f[k_n,x_n,z_n]-f[y_n,x_n,z_n]\right)(x_n-z_n)+b_4(z_n-x_n)(z_n-k_n)(z_n-y_n)\right)}\\&\quad=\frac{\left(c_2\left(c_2^2-c_1(c_3+a_3)\right)(1+c_1\beta)^2e_n^4\right)}{c_1^3}+\cdots+O\left(e_n^9\right).\end{aligned} \tag{3.21}$$

Using (3.21) and the last step of (3.14), we have the error equation (3.15). This manifests that (3.14) is of optimal order eight with four function evaluations per iteration. Hence, the proof is complete and (3.14) reaches the efficiency index $8^{1/4}\approx 1.682$. □

Noting that (3.14) is an extension of the Steffensen and the Ren et al. methods, we should here pull the attention toward this fact that (3.14) is also a generalization of the family given by Zheng et al. in [2]. As a matter of fact, our scheme (3.14) in this paper includes two free parameters (more than that of Zheng et al.), which shows the generality of our technique. Clearly, choosing $a_3=b_4=0$ will result in the Zheng et al. method (2.4). Thus, it is only an special element from the family (3.14).

Remark 3.5. The introduced approximation for the first derivative of the function in the third step of (3.7) can be implemented on any optimal derivative-free fourth-order method without memory for presenting new optimal eighth-order techniques free from derivative. Namely, using (2.1) and (3.13) results in the following optimal eighth-order derivative-free method:

$$
\begin{gathered}
y_n = x_n - \frac{f(x_n)}{f[x_n, k_n]}, \qquad k_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\
z_n = y_n - \frac{f[x_n, y_n] - f[y_n, k_n] + f[x_n, k_n]}{f[x_n, y_n]^2} f(y_n), \\
x_{n+1} \\
= z_n - \frac{f(z_n)}{f[x_n, z_n] + (f[k_n, x_n, y_n] - f[k_n, x_n, z_n] - f[y_n, x_n, z_n])(x_n - z_n) + \gamma (z_n - x_n)(z_n - k_n)(z_n - y_n)},
\end{gathered}
\tag{3.22}
$$

where $\gamma \in \mathbb{R}$ with the following error equation:

$$
\begin{aligned}
e_{n+1} = {} & \left(\frac{1}{c_1^7}\right)\Big((c_2 + c_1 c_2 \beta)^2 \left(-c_1 c_3 (1 + c_1 \beta) + c_2^2 (2 + c_1 \beta)\right) \\
& \times \left(-c_1 c_2 c_3 (1 + c_1 \beta) + c_2^3 (2 + c_1 \beta) + c_1^2 (1 + c_1 \beta)(c_4 + \gamma)\right)\Big) e_n^8 + O\left(e_n^9\right),
\end{aligned}
\tag{3.23}
$$

and if one chooses any of the derivative-free third-order methods without memory in the first two steps and applied the introduced estimation (3.13), then a sixth-order derivative-free technique will be attained. This also shows that the given approximation doubles the convergence rate.

To show the generality of the proposed class and by using Remark 3.5, in what follows, we give some other optimal three-step four-point root solvers without memory. For the first example and by using (11) in [13], we have

$$
\begin{gathered}
y_n = x_n - \frac{f(x_n)}{f[x_n, k_n]}, \qquad k_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\
z_n = y_n - \frac{f(y_n)}{f[x_n, k_n]}\left(1 + \frac{f(y_n)}{f(x_n)} + \frac{f(y_n)}{f(k_n)}\right), \\
x_{n+1} \\
= z_n - \frac{f(z_n)}{f[x_n, z_n] + (f[k_n, x_n, y_n] - f[k_n, x_n, z_n] - f[y_n, x_n, z_n])(x_n - z_n) + \gamma (z_n - x_n)(z_n - k_n)(z_n - y_n)},
\end{gathered}
\tag{3.24}
$$

where $\gamma \in \mathbb{R}$ and

$$\begin{aligned} e_{n+1} = x_{n+1} - \alpha = \left(\frac{1}{c_1^7}\right)&\left(\left(c_2 + c_1c_2\beta\right)^2\left(-c_1c_3(1 + c_1\beta) + c_2^2(5 + c_1\beta(5 + c_1\beta))\right)\right) \\ &\times\left(-c_1c_2c_3(1 + c_1\beta) + c_2^3(5 + c_1\beta(5 + c_1\beta)) + c_1^2(1 + c_1\beta)(c_4 + \gamma)\right)\Big)e_n^8 + O\left(e_n^9\right). \end{aligned} \tag{3.25}$$

Using a different fourth-order method according to [13], we get that

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n, k_n]}, \qquad k_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\ z_n &= y_n - \frac{f(y_n)}{f[x_n, k_n]}\left(\frac{1}{1 - (f(y_n)/f(x_n)) - (f(y_n)/f(k_n))}\right), \\ x_{n+1} & \\ &= z_n - \frac{f(z_n)}{f[x_n, z_n] + (f[k_n, x_n, y_n] - f[k_n, x_n, z_n] - f[y_n, x_n, z_n])(x_n - z_n) + \gamma(z_n - x_n)(z_n - k_n)(z_n - y_n)}, \end{aligned} \tag{3.26}$$

wherein $\gamma \in \mathbb{R}$, with the following error relation:

$$e_{n+1} = \frac{c_2^2(c_2^2 - c_1c_3)(1 + c_1\beta)^4(c_2^3 - c_1c_2c_3 + c_1^2(c_4 + \gamma))}{c_1^7}e_n^8 + O\left(e_n^9\right). \tag{3.27}$$

We can also propose the following general family of three-step iterations using [22]:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n, k_n]}, \qquad k_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\ z_n &= y_n - \frac{f(y_n)}{f[x_n, k_n]}\left(1 + \frac{2 + \beta f[x_n, k_n]}{1 + \beta f[x_n, k_n]}\frac{f(y_n)}{f(x_n)}\right), \\ x_{n+1} & \\ &= z_n - \frac{f(z_n)}{f[x_n, z_n] + (f[k_n, x_n, y_n] - f[k_n, x_n, z_n] - f[y_n, x_n, z_n])(x_n - z_n) + \gamma(z_n - x_n)(z_n - k_n)(z_n - y_n)}, \end{aligned} \tag{3.28}$$

where ($\gamma \in \mathbb{R}$)

$$\begin{aligned} e_{n+1} = {} & \left(\frac{1}{c_1^7}\right)((c_2 + c_1c_2\beta))^2(-c_1c_3(1 + c_1\beta)) + c_2^2(5 + c_1\beta)5 + c_1\beta)))) \\ & \times \left(-c_1c_2c_3(1 + c_1\beta) + c_2^3(5 + c_1\beta)(5 + c_1\beta)\right)) + c_1^2(1 + c_1\beta))(c_4 + \gamma)))e_n^8 + O\left(e_n^9\right). \end{aligned} \tag{3.29}$$

Remark 3.6. Using backward finite difference approximation for the first derivative of the function in the first step of our cycles will end in other new methods as comes next:

$$\begin{gathered} y_n = x_n - \frac{f(x_n)}{f[x_n, k_n]}, \qquad k_n = x_n - \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\ z_n = y_n - \frac{f[x_n, y_n] - f[y_n, k_n] + f[x_n, k_n]}{f[x_n, y_n]^2} f(y_n), \end{gathered}$$

$$\begin{aligned} & x_{n+1} \\ & = z_n - \frac{f(z_n)}{f[x_n, z_n] + (f[k_n, x_n, y_n] - f[k_n, x_n, z_n] - f[y_n, x_n, z_n])(x_n - z_n) + \gamma(z_n - x_n)(z_n - k_n)(z_n - y_n)}, \end{aligned} \tag{3.30}$$

where ($\gamma \in \mathbb{R}$)

$$\begin{aligned} e_{n+1} = {} & \left(\frac{1}{c_1^7}\right)\left(c_2^2(-1 + c_1\beta)^2\left(c_1c_3(1 - c_1\beta) + c_2^2(-2 + c_1\beta)\right)\right. \\ & \times\left.\left(c_1c_2c_3(1 - c_1\beta) + c_2^3(-2 + c_1\beta) + c_1^2(-1 + c_1\beta)(c_4 + \gamma)\right)\right)e_n^8 + O\left(e_n^9\right). \end{aligned} \tag{3.31}$$

We can also have

$$\begin{gathered} y_n = x_n - \frac{f(x_n)}{f[x_n, k_n]}, \qquad k_n = x_n - \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\ z_n = y_n - \frac{f(y_n)}{f[y_n, x_n] + f[k_n, x_n, y_n](y_n - x_n) + a_3(y_n - x_n)(y_n - k_n)}, \quad a_3 \in \mathbb{R}, \end{gathered}$$

$$\begin{aligned} & x_{n+1} \\ & = z_n - \frac{f(z_n)}{f[x_n, z_n] + (f[k_n, x_n, y_n] - f[k_n, x_n, z_n] - f[y_n, x_n, z_n])(x_n - z_n) + \gamma(z_n - x_n)(z_n - k_n)(z_n - y_n)}, \end{aligned} \tag{3.32}$$

with the following error equation ($\gamma \in \mathbb{R}$):

$$e_{n+1} = \left(\frac{1}{c_1^7}\right)\left(c_2^2\left(c_2^2 - c_1(c_3 + a_3)\right)(-1 + c_1\beta)^4 \times\left(c_2^3 - c_1c_2(c_3 + a_3) + c_1^2(c_4 + \gamma)\right)\right)e_n^8 + O\left(e_n^9\right). \tag{3.33}$$

4. Higher-Order Optimal Schemes

In this section, we take a special heed to generalize the novel scheme (3.14) by using the same idea. To increase the local order of convergence and the efficiency index more, we should consider cycles in which there are four, five, and so forth steps. Here, we consider a four-step cycle at which (3.14) is in the first three steps and Newton's method is in the fourth step as follows ($a_3, b_4 \in \mathbb{R}$):

$$\begin{gathered} y_n = x_n - \frac{f(x_n)}{f[k_n, x_n]}, \qquad k_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\ z_n = y_n - \frac{f(y_n)}{f[y_n, x_n] + f[k_n, x_n, y_n](y_n - x_n) + a_3(y_n - x_n)(y_n - k_n)}, \\ w_n \\ = z_n - \frac{f(z_n)}{f[x_n, z_n] + (f[k_n, x_n, y_n] - f[k_n, x_n, z_n] - f[y_n, x_n, z_n])(x_n - z_n) + b_4(z_n - x_n)(z_n - k_n)(z_n - y_n)}, \\ x_{n+1} = w_n - \frac{f(w_n)}{f'(w_n)}. \end{gathered} \tag{4.1}$$

It is easy to check that (4.1) is a sixteenth-order method with six evaluations per iteration and it has the efficiency index $16^{1/6} \approx 1.587$. To make (4.1) optimal derivative-free multipoint technique with $16^{1/5} \approx 1.741$ as the efficiency index, it is required to approximate $f'(w_n)$ effectively. To do this, we estimate $f(t)$ in the domain D, by a novel polynomial of degree four like the similar cases in (3.2) and (3.8) as follows:

$$f(t) \approx r(t) = r_0 + r_1(t - x_n) + r_2(t - x_n)^2 + r_3(t - x_n)^3 + r_4(t - x_n)^4 + r_5(t - x_n)^5, \tag{4.2}$$

where its first derivative takes the form as follows: $f'(t) \approx r'(t) = r_1 + 2r_2(t-x_n) + 3r_3(t-x_n)^2 + 4r_4(t-x_n)^3 + 5r_5(t-x_n)^4$. Substituting the known values in (4.2) gives us the unknown values by solving a system of linear equations in what follows:

$$\begin{pmatrix} (k_n - x_n) & (k_n - x_n)^2 & (k_n - x_n)^3 & (k_n - x_n)^4 \\ (y_n - x_n) & (y_n - x_n)^2 & (y_n - x_n)^3 & (y_n - x_n)^4 \\ (z_n - x_n) & (z_n - x_n)^2 & (z_n - x_n)^3 & (z_n - x_n)^4 \\ (w_n - x_n) & (w_n - x_n)^2 & (w_n - x_n)^3 & (w_n - x_n)^4 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} f(k_n) - f(x_n) - r_5(k_n - x_n)^4 \\ f(y_n) - f(x_n) - r_5(y_n - x_n)^4 \\ f(z_n) - f(x_n) - r_5(z_n - x_n)^4 \\ f(w_n) - f(x_n) - r_5(w_n - x_n)^4 \end{pmatrix}. \tag{4.3}$$

Note that for the case of multipoint methods, it is more desirable to apply the symbolic calculations using one of the modern software in mathematics. Using Mathematica 8 gives us the unknown parameters by the command LinearSolve as in Algorithm 1.

Here to cut the long story short, we just provide the following theorem.

Theorem 4.1. *The four-step method*

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[k_n, x_n]}, \qquad k_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R} \setminus \{0\}, \\ z_n &= y_n - \frac{f(y_n)}{f[y_n, x_n] + f[k_n, x_n, y_n](y_n - x_n) + a_3(y_n - x_n)(y_n - k_n)}, \\ w_n &= z_n - \frac{f(z_n)}{f[x_n, z_n] + (f[k_n, x_n, y_n] - f[k_n, x_n, z_n] - f[y_n, x_n, z_n])(x_n - z_n) + b_4(z_n - x_n)(z_n - k_n)(z_n - y_n)}, \\ x_{n+1} &= w_n - \frac{f(w_n)}{r_1 + 2r_2(w_n - x_n) + 3r_3(w_n - x_n)^2 + 4r_4(w_n - x_n)^3 + 5r_5(w_n - x_n)^4}, \quad r_5 \in \mathbb{R}, \end{aligned} \tag{4.4}$$

converges to the simple root of $f(x) = 0$ *in the domain* D *with local sixteenth order of convergence, where the first three steps are the optimal eighth-order method* (3.14) *and* r_1, r_2, r_3 *and* r_4 *are given as follows:*

$$\begin{aligned} r_4 &= \frac{f[k_n, x_n] - f[z_n, x_n]}{(k_n - w_n)(k_n - y_n)(k_n - z_n)} + \frac{f[z_n, x_n] - f[w_n, x_n]}{(k_n - w_n)(w_n - y_n)(w_n - z_n)} \\ &\quad + \frac{f[y_n, x_n] - f[z_n, x_n]}{(k_n - y_n)(w_n - y_n)(y_n - z_n)}, \\ r_3 &= \frac{(f[k_n, x_n] - f[z_n, x_n])/(k_n - z_n) + (-f[y_n, x_n] + f[z_n, x_n])/(y_n - z_n)}{k_n - y_n}, \end{aligned}$$

$$r_2=\frac{-f[k_n,x_n]+f[y_n,x_n]+(k_n-x_n)^2r_3+(k_n-x_n)^3r_4-r_3(x_n-y_n)^2+r_4(x_n-y_n)^3}{y_n-k_n},$$

$$r_1=f[w_n,x_n]-r_2(w_n-x_n)-r_3(w_n-x_n)^2-r_4(w_n-x_n)^3. \tag{4.5}$$

Proof. The proof of this theorem is similar to the proofs of Theorems 3.1 and 3.4. Hence, it is omitted. □

Remark 4.2. The simplified form of the approximation for the derivative in the quotient of Newton's iteration at the fourth step of our cycle (without the index n) is as comes next:

$$\begin{aligned}&r_1+2r_2(w_n-x_n)+3r_3(w_n-x_n)^2+4r_4(w_n-x_n)^3+5r_5(w_n-x_n)^4\\&\quad=f[z_n,w_n]+f[w_n,z_n,y_n](w_n-z_n)+f[w_n,z_n,y_n,x_n](w_n-z_n)(w_n-y_n)\\&\qquad+f[w_n,z_n,y_n,x_n,k_n]\times(w_n-z_n)(w_n-y_n)(w_n-x_n)\\&\qquad+r_5(w_n-x_n)(w_n-k_n)(w_n-y_n)(w_n-z_n)=\Upsilon(n).\end{aligned}\tag{4.6}$$

Remark 4.3. Note that if we choose another optimal derivative-free eighth-order method in the first three steps of (4.4) and then implement the introduced approximation in the fourth-step, we will obtain a new technique of optimal sixteenth convergence rate to solve nonlinear scalar equations. In what follows, we give some of such optimal 16^{th}-order schemes:

$$\begin{aligned}y_n&=x_n-\frac{f(x_n)}{f[k_n,x_n]},\qquad k_n=x_n+\beta f(x_n),\quad \beta\in\mathbb{R}\setminus\{0\},\\z_n&=y_n-\frac{f[x_n,y_n]-f[y_n,k_n]+f[x_n,k_n]}{f[x_n,y_n]^2}f(y_n),\\w_n&=z_n-\frac{f(z_n)}{f[x_n,z_n]+(f[k_n,x_n,y_n]-f[k_n,x_n,z_n]-f[y_n,x_n,z_n])(x_n-z_n)+b_4(z_n-x_n)(z_n-k_n)(z_n-y_n)},\\x_{n+1}&=w_n-\frac{f(w_n)}{\Upsilon(n)},\end{aligned}\tag{4.7}$$

where b_4, $r_5\in\mathbb{R}$ and

$$\begin{aligned}y_n&=x_n-\frac{f(x_n)}{f[k_n,x_n]},\qquad k_n=x_n-\beta f(x_n),\quad \beta\in\mathbb{R}\setminus\{0\},\\z_n&=y_n-\frac{f[x_n,y_n]-f[y_n,k_n]+f[x_n,k_n]}{f[x_n,y_n]^2}f(y_n),\end{aligned}$$

$$
\begin{aligned}
&w_n \\
&= z_n - \frac{f(z_n)}{f[x_n,z_n] + \left(f[k_n,x_n,y_n] - f[k_n,x_n,z_n] - f[y_n,x_n,z_n]\right)(x_n - z_n) + b_4(z_n - x_n)(z_n - k_n)(z_n - y_n)}, \\
&x_{n+1} = w_n - \frac{f(w_n)}{\Upsilon(n)}.
\end{aligned}
\tag{4.8}
$$

Remark 4.4. If one considers an n-step derivative-free construction in which the first $n-1$ steps are any of the existing optimal derivative-free methods of order 2^{n-1}, then by considering a polynomial approximation of degree $n+1$ (with one free parameter, as in the idea above), as discussed above, an optimal derivative-free method of order 2^n with the optimal efficiency index 2 will be attained.

Now, let us list the efficiency indices of some of the well-known methods with the contributed schemes in this study. The results of comparisons are made in Table 1. As it shows, our schemes are totally better than the existing methods in the literature in terms of efficiency index.

5. Computational Tests

The practical utilities of the main contributions in this paper are given by solving a couple of numerical examples and comparison with other well-known methods of different orders in this section. The main purpose of demonstrating the new derivative-free methods for nonlinear equations is purely to illustrate the accuracy of the approximate solution, the stability of the convergence, and the consistency of the results and to determine the efficiency of the new iterative techniques.

We have computed the root of each test function for an initial approximation in the neighborhood D. We remark that we have chosen for comparison only the methods which do not require the computation of derivatives of the function to carry out iterations. We have used the fourth-order scheme of Ren et al. (RM4), the eighth-order derivative-free family of Zheng et al. (2.3) with $\beta = 1$, the eighth-order derivative-free method of Soleymani (2.4), and our novel contributed techniques (3.14) with $a_3 = 1$, $b_4 = -3$, (3.22) with $\gamma = 0$, and (3.30) with $\gamma = 0$. We here recall the family of derivative-free methods, which was given by Kung and Traub (KT8) [3] as comes next:

$$
\begin{aligned}
y_n &= x_n + \beta f(x_n), \\
z_n &= y_n - \beta \frac{f(x_n) f(y_n)}{f(y_n) - f(x_n)}, \\
w_n &= z_n - \frac{f(x_n) f(y_n)}{f(z_n) - f(x_n)} \left[\frac{1}{f[y_n,x_n]} - \frac{1}{f[z_n,y_n]} \right],
\end{aligned}
$$

$$x_{n+1}=w_n-\frac{f(x_n)f(y_n)f(z_n)}{f(w_n)-f(x_n)}$$
$$\times\left[\frac{1}{f(w_n)-f(y_n)}\left\{\frac{1}{f[w_n,z_n]}-\frac{1}{f[z_n,y_n]}\right\}-\frac{1}{f(z_n)-f(x_n)}\left\{\frac{1}{f[z_n,y_n]}-\frac{1}{f[y_n,x_n]}\right\}\right], \tag{5.1}$$

where $\beta \in \mathbb{R}\setminus\{0\}$; we have used this scheme in numerical comparison by choosing $\beta = 1$. It is completely clear that (4.4) performs better than any of the other methods; hence we do not include it in numerical comparisons. The test functions and their roots are given in Table 2.

The results are summarized in Tables 3 and 4, in terms of the required number of iterations to obtain an approximation of the root. In our numerical comparisons, we have used 2000-digit floating point in our calculations.

As Tables 3 and 4 manifest, our contributed methods are accurate and efficient in contrast by the other high-order schemes in terms of required number of iterations. It is well known that convergence of iteration formula is guaranteed only when the initial approximation is sufficiently near to the root. This issue will be the main body of the next section. However, our class of derivative-free method is much better and has better convergence rate in contrast to the existing high-order derivative-free methods in the literature.

6. A Hybrid Algorithm to Capture All the Solutions

An important aspect in implementing the iterative methods for the solution of nonlinear equations and systems relies on the choice of the initial approximation. There are a few known ways in the literature [23] to extract a starting point for the solutions of nonlinear functions. In practice, users need to find out robust approximations for all the zeros in an interval. Thus, to remedy this and to respond on this need, we provide a way to extract all the real zeros of nonlinear function in the interval $D = [a, b]$. We use the command Reduce[] in MATHEMATICA 8 [24, 25]. Hence, we give a hybrid algorithm including two main steps, a predictor and a corrector. In the predictor step, we extract initial approximations for all the zeros in an interval up to 8 decimal places. Then in the corrector step, an eighth-order method, for example, (3.28), will be used to boost up the accuracy of the starting points up to any tolerance. We also give some significant cautions for applying on different test functions.

In what follows, we keep going by choosing an oscillatory function $f(x) = -\cos(2 - x^2) + \log(x/7) + (1/10)$ in the domain $D = [0, 15]$.

Let us define the function and the domain for imposing the Reduce[] command as in Algorithm 2.

One may note that Reduce[] works with function of exact arithmetic. Hence, if a nonlinear function is the floating point arithmetic, that is, has inexact coefficients, thus we should write it in the exact arithmetic when we enter it into the above piece of code.

Now we store the list of initial approximations in initialValues, by the following piece of code, which also sort the initial points. The tol will specify that the accuracy of each member of the provided sequence to be correct up to utmost tol, decimal places (Algorithm 3).

```
LinearSolve[{{k-x,((k-x)^2), ((k-x)^3),((k-x)^4) } ,
{ y-x, ((y-x)^2),((y-x)^3), ((y-x)^4) } , { z-x, ((z-x)^2),
((z-x)^3),((z-x)^4) } , { w-x, ((w-x)^2), ((w-x)^3),((w-x)^4)}},
{ f[k]-f[x]-r5*(k-x)^4,f[y]-f[x]-r5*(y-x)^4, f[z]-f[x]
-r5*(z-x)^4,f[w]-f[x]-r5*(w-x)^4}]//FullSimplify
```

Algorithm 1

```
ClearAll[f, x, rts, initialValues, tol, a, b, j,
NumberOfGuesses, FD, nMax, beta, gamma, y, z, k,
fx, fk, fy, fz]
f[x_]:= Log[x/7] - Cos[x^2 - 2] + 1/10
a = 0; b = 15;
Plot[f[x], { x, a, b } , Background -> LightBlue,
 PlotStyle -> {Brown, Thick}, PlotRange -> All,
 PerformanceGoal -> "Quality"]
rts = Reduce[f[x] == 0 && a < x < b, x];
```

Algorithm 2

It is obvious that f is so oscillatory, and by the above predictor piece of Mathematica code, we attain that it has 69 real solutions. Note that the graph of the function f has been drawn in Figure 1. Now it is time to use one of the derivative-free methods given in Section 2 or 3 for correcting the elements of the obtained list. We use the method (3.28) in Algorithm 4.

In Algorithm 4, we have the ability to choose the nonzero free parameter β, as well as the parameter `gamma`. Note that for optimal eighth-order methods and with such an accurate list of initial points, we believe that only two full iterations are required. Hence, we have selected `nMax`= 2. Note that if a user needs much more accuracy, thus higher number of steps should be taken. It should be remarked that in order to work with such a high accuracy, we must then choose more than 2000 decimal places arithmetic in our calculations.

Due to page limitations, we are not able to list the results for all 69 zeros. However, running the above algorithm could capture all the real zeros of the nonlinear functions. We also remark three points. One is that for many oscillatory function or for nonsmooth functions, the best way is to first divide the whole interval into some subintervals and then find all the zeros of the function on the subintervals. And second, in case of having a root cluster, that is, when the zeros are concentrated on a very small area, then it would be better to increase the *first tolerance* of our algorithm in the predictor step, to find reliable starting points and then start the process. And last, if the nonlinear function has an exact solution, that is to say, an integer be the solution of a nonlinear function, then the first step of our algorithm finds this exact solution, and an error-like message would be generated by applying our second step.

For instance, the function $g(x) = (x^2 - 4)\sin(100x)$ on the interval $D = [0, 10]$ has 319 real solutions in which one of them (its plot is given in Figure 2), that is, 2, is an exact one.

```
tol = 8;
initialValues = Sort[N[x /.  { ToRules[rts]}, tol]]
Length[initialValues]
Accuracy[initialValues]
```

Algorithm 3

Table 1: Comparisons of efficiency indices for different types of derivative-free schemes.

M.	(1.1)	(2.1)	(2.2)	(2.3)	(2.4)	(5.1)	(3.14)	(4.4)
E.I.	$2^{1/2} \approx 1.414$	$4^{1/3} \approx 1.587$	$6^{1/5} \approx 1.495$	$8^{1/4} \approx 1.682$	$8^{1/4} \approx 1.682$	$8^{1/4} \approx 1.682$	$8^{1/4} \approx 1.682$	$16^{1/5} \approx 1.741$

Table 2: The test functions considered in this study.

Test functions	Simple zeros	Initial guesses
$f_1(x) = 3x + \sin(x) - e^x$	$\alpha_1 \approx 0.360421702960324$	$x_0 = 0.2$
$f_2(x) = \sin(x) - 0.5$	$\alpha_2 \approx 0.523598775598299$	$x_0 = 0.3$
$f_3(x) = x^2 - e^x - 3x + 2$	$\alpha_3 \approx 0.257530285439861$	$x_0 = 0.4$
$f_4(x) = x^3 + 4x^2 - 10$	$\alpha_4 \approx 1.365230013414097$	$x_0 = 1.37$
$f_5(x) = xe^{-x} - 0.1$	$\alpha_5 \approx 0.111832559158963$	$x_0 = 0.2$
$f_6(x) = x^3 - 10$	$\alpha_6 \approx 2.15443490031884$	$x_0 = 2.16$
$f_7(x) = 10xe^{-x^2} - 1$	$\alpha_7 \approx 1.679630610428450$	$x_0 = 1.4$
$f_8(x) = \cos(x) - x$	$\alpha_8 \approx 0.739085133215161$	$x_0 = 0.3$

Table 3: Results of comparisons for different methods after two full iterations.

Test functions	RM4	(2.3)	(2.4)	(5.1)	(3.14)	(3.22)	(3.30)
$\lvert f_1(x_2)\rvert$	$0.4e-13$	$0.1e-57$	$0.4e-45$	$0.2e-52$	$0.1e-44$	$0.2e-57$	$0.2e-65$
$\lvert f_2(x_2)\rvert$	$0.1e-14$	$0.4e-64$	$0.7e-51$	$0.2e-57$	$0.7e-41$	$0.1e-63$	$0.1e-96$
$\lvert f_3(x_2)\rvert$	$0.4e-20$	$0.1e-83$	$0.2e-82$	$0.7e-82$	$0.2e-61$	$0.5e-83$	$0.5e-79$
$\lvert f_4(x_2)\rvert$	$0.1e-28$	$0.1e-124$	$0.1e-94$	$0.4e-115$	$0.1e-118$	$0.6e-124$	$0.2e-126$
$\lvert f_5(x_2)\rvert$	$0.6e-15$	$0.1e-59$	$0.1e-34$	$0.6e-49$	$0.1e-53$	$0.1e-53$	$0.1e-73$
$\lvert f_6(x_2)\rvert$	$0.7e-29$	$0.4e-125$	$0.1e-95$	$0.2e-115$	$0.2e-117$	$0.2e-124$	$0.1e-127$
$\lvert f_7(x_2)\rvert$	$0.1e-4$	$0.2e-24$	$0.1e-14$	$0.5e-9$	$0.3e-26$	$0.1e-33$	$0.1e-16$
$\lvert f_8(x_2)\rvert$	$0.1e-15$	$0.2e-71$	$0.1e-62$	$0.1e-58$	$0.2e-44$	$0.3e-78$	$0.1e-34$

Table 4: Results of comparisons for different methods after three full iterations.

Test functions	RM4	(2.3)	(2.4)	(5.1)	(3.14)	(3.22)	(3.30)
$\lvert f_1(x_3)\rvert$	$0.3e-54$	$0.1e-466$	$0.1e-363$	$0.1e-422$	$0.3e-361$	$0.5e-463$	$0.6e-529$
$\lvert f_2(x_3)\rvert$	$0.1e-59$	$0.5e-516$	$0.1e-408$	$0.9e-462$	$0.7e-328$	$0.1e-510$	$0.4e-780$
$\lvert f_3(x_3)\rvert$	$0.2e-84$	$0.1e-676$	$0.3e-668$	$0.1e-663$	$0.1e-497$	$0.1e-673$	$0.2e-640$
$\lvert f_4(x_3)\rvert$	$0.2e-117$	$0.9e-1004$	$0.2e-761$	$0.3e-927$	$0.1e-955$	$0.2e-999$	$0.6e-1019$
$\lvert f_5(x_3)\rvert$	$0.6e-60$	$0.5e-478$	$0.2e-275$	$0.7e-391$	$0.1e-427$	$0.2e-429$	$0.7e-591$
$\lvert f_6(x_3)\rvert$	$0.1e-118$	$0.1e-1008$	$0.3e-768$	$0.7e-930$	$0.2e-946$	$0.1e-1002$	$0.4e-1030$
$\lvert f_7(x_3)\rvert$	$0.9e-20$	$0.4e-199$	$0.3e-119$	$0.1e-75$	$0.2e-214$	$0.2e-273$	$0.1e-136$
$\lvert f_8(x_3)\rvert$	$0.3e-66$	$0.1e-578$	$0.3e-507$	$0.1e-476$	$0.3e-360$	$0.1e-634$	$0.4e-283$

```
nMax = 2; beta = SetAccuracy[1, 2000];
gamma = SetAccuracy[0, 2000];
For[j = 1, j <= NumberOfGuesses, j++,
  {x = SetAccuracy[initialValues[[j]], 2000],
   Do[
    fx = SetAccuracy[f[x], 2000];
    k = SetAccuracy[x + beta fx, 2000];
    fk = SetAccuracy[f[k], 2000];
    FD = SetAccuracy[(fk - fx)/(beta fx), 2000];
    y = SetAccuracy[x - fx/FD, 2000];
    fy = SetAccuracy[f[y], 2000];
    z = SetAccuracy[y - (fy/FD)*(1 + ((2 + beta FD)/(1
              + beta FD)) (fy/fx)), 2000];
    fz = SetAccuracy[f[z], 2000];
    FDxz = SetAccuracy[(fz - fx)/(z - x), 2000];
    FDxy = SetAccuracy[(fx - fy)/(x - y), 2000];
    FDxk = SetAccuracy[(fk - fx)/(k - x), 2000];
    x = SetAccuracy[z - fz/(FDxz + ((FDxk - FDxy)/(k - y)
            - (FDxk - FDxz)/(k - z)
            - (FDxy - FDxz)/(y - z))*(x - z)
      + gamma (z - x)*(z - k)*(z - y)), 2000];
    , {n, nMax}];
  Print[Column[{
     N[x, 128],
     N[f[x]]}
    , Background -> {{LightGreen, LightGray}},
    Frame -> True]];
   }
  ];
```

Algorithm 4

Thus, the first step of the previously mentioned algorithm finds the following very efficient list of starting points in which 2, is the exact solution:

{0.031415927, 0.062831853, 0.094247780, 0.12566371, 0.15707963, 0.18849556, 0.21991149,
0.25132741, 0.28274334, 0.31415927, 0.34557519, 0.37699112, 0.40840704, 0.43982297,
0.47123890, 0.50265482, 0.53407075, 0.56548668, 0.59690260, 0.62831853, 0.65973446,
0.69115038, 0.72256631, 0.75398224, 0.78539816, 0.81681409, 0.84823002, 0.87964594,
0.91106187, 0.94247780, 0.97389372, 1.0053096, 1.0367256, 1.0681415, 1.0995574,
1.1309734, 1.1623893, 1.1938052, 1.2252211, 1.2566371, 1.2880530, 1.3194689, 1.3508848,
1.3823008, 1.4137167, 1.4451326, 1.4765485, 1.5079645, 1.5393804, 1.5707963, 1.6022123,
1.6336282, 1.6650441, 1.6964600, 1.7278760, 1.7592919, 1.7907078, 1.8221237, 1.8535397,
1.8849556, 1.9163715, 1.9477874, 1.9792034, 2.0000000, 2.0106193, 2.0420352, 2.0734512,
2.1048671, 2.1362830, 2.1676989, 2.1991149, 2.2305308, 2.2619467, 2.2933626, 2.3247786,

$$
\begin{gathered}
2.3561945, 2.3876104, 2.4190263, 2.4504423, 2.4818582, 2.5132741, 2.5446900, 2.5761060, \\
2.6075219, 2.6389378, 2.6703538, 2.7017697, 2.7331856, 2.7646015, 2.7960175, 2.8274334, \\
2.8588493, 2.8902652, 2.9216812, 2.9530971, 2.9845130, 3.0159289, 3.0473449, 3.0787608, \\
3.1101767, 3.1415927, 3.1730086, 3.2044245, 3.2358404, 3.2672564, 3.2986723, 3.3300882, \\
3.3615041, 3.3929201, 3.4243360, 3.4557519, 3.4871678, 3.5185838, 3.5499997, 3.5814156, \\
3.6128316, 3.6442475, 3.6756634, 3.7070793, 3.7384953, 3.7699112, 3.8013271, 3.8327430, \\
3.8641590, 3.8955749, 3.9269908, 3.9584067, 3.9898227, 4.0212386, 4.0526545, 4.0840704, \\
4.1154864, 4.1469023, 4.1783182, 4.2097342, 4.2411501, 4.2725660, 4.3039819, 4.3353979, \\
4.3668138, 4.3982297, 4.4296456, 4.4610616, 4.4924775, 4.5238934, 4.5553093, 4.5867253, \\
4.6181412, 4.6495571, 4.6809731, 4.7123890, 4.7438049, 4.7752208, 4.8066368, 4.8380527, \\
4.8694686, 4.9008845, 4.9323005, 4.9637164, 4.9951323, 5.0265482, 5.0579642, 5.0893801, \\
5.1207960, 5.1522120, 5.1836279, 5.2150438, 5.2464597, 5.2778757, 5.3092916, 5.3407075, \\
5.3721234, 5.4035394, 5.4349553, 5.4663712, 5.4977871, 5.5292031, 5.5606190, 5.5920349, \\
5.6234508, 5.6548668, 5.6862827, 5.7176986, 5.7491146, 5.7805305, 5.8119464, 5.8433623, \\
5.8747783, 5.9061942, 5.9376101, 5.9690260, 6.0004420, 6.0318579, 6.0632738, 6.0946897, \\
6.1261057, 6.1575216, 6.1889375, 6.2203535, 6.2517694, 6.2831853, 6.3146012, 6.3460172, \\
6.3774331, 6.4088490, 6.4402649, 6.4716809, 6.5030968, 6.5345127, 6.5659286, 6.5973446, \\
6.6287605, 6.6601764, 6.6915924, 6.7230083, 6.7544242, 6.7858401, 6.8172561, 6.8486720, \\
6.8800879, 6.9115038, 6.9429198, 6.9743357, 7.0057516, 7.0371675, 7.0685835, 7.0999994, \\
7.1314153, 7.1628313, 7.1942472, 7.2256631, 7.2570790, 7.2884950, 7.3199109, 7.3513268, \\
7.3827427, 7.4141587, 7.4455746, 7.4769905, 7.5084064, 7.5398224, 7.5712383, 7.6026542, \\
7.6340701, 7.6654861, 7.6969020, 7.7283179, 7.7597339, 7.7911498, 7.8225657, 7.8539816, \\
7.8853976, 7.9168135, 7.9482294, 7.9796453, 8.0110613, 8.0424772, 8.0738931, 8.1053090, \\
8.1367250, 8.1681409, 8.1995568, 8.2309728, 8.2623887, 8.2938046, 8.3252205, 8.3566365, \\
8.3880524, 8.4194683, 8.4508842, 8.4823002, 8.5137161, 8.5451320, 8.5765479, 8.6079639, \\
8.6393798, 8.6707957, 8.7022117, 8.7336276, 8.7650435, 8.7964594, 8.8278754, 8.8592913, \\
8.8907072, 8.9221231, 8.9535391, 8.9849550, 9.0163709, 9.0477868, 9.0792028, 9.1106187, \\
9.1420346, 9.1734505, 9.2048665, 9.2362824, 9.2676983, 9.2991143, 9.3305302, 9.3619461, \\
9.3933620, 9.4247780, 9.4561939, 9.4876098, 9.5190257, 9.5504417, 9.5818576, 9.6132735, \\
9.6446894, 9.6761054, 9.7075213, 9.7389372, 9.7703532, 9.8017691, 9.8331850, 9.8646009, \\
9.8960169, 9.9274328, 9.9588487, 9.9902646\}.
\end{gathered}
\tag{6.1}
$$

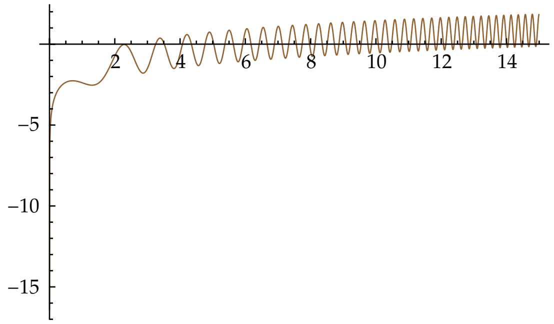

Figure 1: The plot of the function $f(x)$ including the zeros.

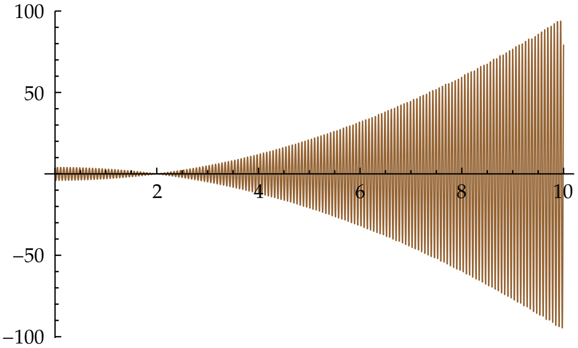

Figure 2: The plot of the function $g(x)$ including the zeros.

Now we are able to solve nonlinear equations with finitely many roots in an interval and find all the real zeros in a short piece of time. Finding robust ways, to capture the complex solutions along working with complex nonlinear functions, can be taken into account as future works.

7. Concluding Comments

In a word, a general way for producing optimal methods based upon the still unproved hypothesis of Kung-Traub [3] has been discussed fully. In these methods, it is necessary to begin with one initial approximation x_0. The convergence results of the iterative sequences generated by new algorithms were provided. Our contributed techniques without memory reach the optimal efficiency indices $8^{1/4} \approx 1.682$ and $16^{1/5} \approx 1.741$.

In the limitation case, based on a polynomial of degree $n + 1$ for a method with n steps, we can obtain an iterative scheme with efficiency index 2, which is the highest possible efficiency in this field of study for methods without memory. By numerical experiments above, we verify that new methods are effective and accurate. We conclude that the new methods presented in this paper are competitive with other recognized efficient equation solvers.

Note that construction of some ways, which launch n-step Steffensen-type methods *with memory* with the order up to $2^n + 2^{n-1}$ (50% of an improvement) requiring the same computational cost and applying elegant and quite natural accelerating procedure, could be considered as future works. Note that the second aim of this paper was given by presenting a robust algorithm for solving nonlinear equations with finitely many roots in an interval. In fact, a way for extracting very robust initial approximations as the seeds to start the iterative Steffensen-type methods was presented. Now we are able to capture all the solutions with any desire number of decimal places.

Acknowledgment

The second author is an IEEE member.

References

[1] J. F. Steffensen, "Remarks on iteration," *Skand Aktuar Tidsr*, vol. 16, pp. 64–72, 1933.

[2] 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.

[3] 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.

[4] T. Sauer, *Numerical Analysis*, Pearson, Boston, Mass, USA, 2006.

[5] F. Soleymani and S. Karimi Vanani, "Optimal Steffensen-type methods with eighth order of convergence," *Computers & Mathematics with Applications*, vol. 62, no. 12, pp. 4619–4626, 2011.

[6] F. Soleymani, "On a bi-parametric class of optimal eighth-order derivative-free methods," *International Journal of Pure and Applied Mathematics*, vol. 72, no. 1, pp. 27–37, 2011.

[7] F. Soleymani, S. K. Khattri, and S. Karimi Vanani, "Two new classes of optimal Jarratt-type fourth-order methods," *Applied Mathematics Letters*, vol. 25, no. 5, pp. 847–853, 2012.

[8] F. Soleymani, M. Sharifi, and B. S. Mousavi, "An improvement of Ostrowski's and King's techniques with optimal convergence order eight," *Journal of Optimization Theory and Applications*, vol. 153, no. 1, pp. 225–236, 2012.

[9] D. K. R. Babajee, *Analysis of higher order variants of Newton's method and their applications to differential and integral equations and in ocean acidification [Ph.D. thesis]*, University of Mauritius, 2010.

[10] Z. Liu, Q. Zheng, and P. Zhao, "A variant of Steffensen's method of fourth-order convergence and its applications," *Applied Mathematics and Computation*, vol. 216, no. 7, pp. 1978–1983, 2010.

[11] A. Cordero, J. L. Hueso, E. Martínez, and J. R. Torregrosa, "Steffensen type methods for solving nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 236, no. 12, pp. 3058–3064, 2012.

[12] F. Soleymani, "An optimally convergent three-step class of derivative-free methods," *World Applied Sciences Journal*, vol. 13, no. 12, pp. 2515–2521, 2011.

[13] M. S. Petković, S. Ilić, and J. Džunić, "Derivative free two-point methods with and without memory for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 217, no. 5, pp. 1887–1895, 2010.

[14] B. H. Dayton, T.-Y. Li, and Z. Zeng, "Multiple zeros of nonlinear systems," *Mathematics of Computation*, vol. 80, no. 276, pp. 2143–2168, 2011.

[15] J. F. Traub, *Iterative Methods for the Solution of Equations*, Chelsea Publishing Company, New York, NY, USA, 1982.

[16] F. Soleymani, "Optimized Steffensen-type methods with eighth order of convergence and high efficiency index," *International Journal of Mathematics and Mathematical Sciences*, vol. 2012, Article ID 932420, 18 pages, 2012.

[17] F. Soleymani, D. K. R. Babajee, and M. Sharifi, "Modified Jarratt method without memory with twelfth-order convergence," *Annals of the University of Craiova*, vol. 39, pp. 21–34, 2012.

[18] F. Soleymani, "On a new class of optimal eighth-order derivative-free methods," *Acta Universitatis Sapientiae. Mathematica*, vol. 3, pp. 169–180, 2011.

[19] F. Soleymani and F. Soleimani, "Novel computational derivative-free methods for simple roots," *Fixed Point Theory*, vol. 13, no. 1, pp. 247–258, 2012.

[20] H. Ren, Q. Wu, and W. Bi, “A class of two-step Steffensen type methods with fourth-order convergence,” *Applied Mathematics and Computation*, vol. 209, no. 2, pp. 206–210, 2009.

[21] A. Cordero, J. L. Hueso, E. Martinez, and J. R. Torregrosa, “Generating optimal derivative free iterative methods for nonlinearequations by using polynomial interpolation,” *Mathematical and Computer Modelling*. In press.

[22] F. Soleymani, “Two classes of iterative schemes for approximating simple roots,” *Journal of Applied Sciences*, vol. 11, no. 19, pp. 3442–3446, 2011.

[23] http://mathematica.stackexchange.com/questions/275/updating-wagons-findallcrossings2d-function.

[24] R. Hazrat, *Mathematica: A Problem-Centered Approach*, Springer, New York, NY, USA, 2010.

[25] J. Hoste, *Mathematica Demystified*, McGraw-Hill, New York, NY, USA, 2009.