Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 134673, 9 pages
http://dx.doi.org/10.1155/2014/134673

Research Article

On a New Three-Step Class of Methods and Its Acceleration for Nonlinear Equations

T. Lotfi,[1] K. Mahdiani,[1] Z. Noori,[1] F. Khaksar Haghani,[2] and S. Shateyi[3]

[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 Applied Mathematics, School of Mathematical and Natural Sciences, University of Venda, Thohoyandou 0950, South Africa*

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

Received 20 July 2014; Accepted 16 August 2014; Published 3 September 2014

Academic Editor: Hassan Saberi Nik

A class of derivative-free methods without memory for approximating a simple zero of a nonlinear equation is presented. The proposed class uses four function evaluations per iteration with convergence order eight. Therefore, it is an optimal three-step scheme without memory based on Kung-Traub conjecture. Moreover, the proposed class has an accelerator parameter with the property that it can increase the convergence rate from eight to twelve without any new functional evaluations. Thus, we construct a with memory method that increases considerably efficiency index from $8^{1/4} \approx 1.681$ to $12^{1/4} \approx 1.861$. Illustrations are also included to support the underlying theory.

1. Introduction

The first attempts for classifying iterative root-finding methods were done by Traub [1]. He divided iterative methods for finding zeros of a function into two sets: one-point and multipoint methods. There is a fact about how to create a new method. As Traub investigated in his book [1], and Kung and Traub mentioned in [2], construction of one-point methods is not a useful task. In other words, to construct an optimal one-point method with convergence order n, we need n functional evaluations, while for construction an optimal method without memory having convergence order 2^n; only $n + 1$ function evaluations are required.

To be more precise, constructing an optimal one-point method with eighth-order convergence needs eight function evaluations, while constructing an optimal three-point method without memory having the same convergence order requires four functional evaluations. As a result, many researchers have paid much attention to construct optimal multipoint iterations without memory based on the unproved conjecture due to Kung and Traub: any multipoint iteration without memory using $n + 1$ function evaluations can reach the optimal order 2^n.

This work follows two main goals: first developing a new optimal three-step derivative-free class of methods without memory and second developing the proposed class to methods with memory. In this way, it reaches convergence order 12 without any new functional evaluations. Because of the derivative-free property of the proposed class, it can be used for finding zeros of not only smooth functions but also nonsmooth ones. Moreover, as we pointed out above we can reach the convergence order 12 using the same functional evaluations (to three-step without memory iterations) and, therefore, increasing 50% convergence order is the other aspect and contribution of this work.

Note that in most test problems for nonlinear equations computing derivatives are an easy exercise. However, for some practical problems computing the derivative might be a cumbersome task and we have to relay on methods free of derivatives. For further reading on this topic, one may refer to [3–6].

The paper is organized as follows. First, a new without memory family of optimal order eight, consuming four function evaluations per iteration, is proposed by using two weight functions in Section 2. A different way to compute the order of convergence for iterative methods that use divided differences instead of derivatives is presented in Section 3, when we derive a method with memory. The significant increase of convergence speed is achieved without additional function evaluations, which is the main advantage of such methods. Section 4 is devoted to numerical results connected to the order of the methods with and without memory. And finally, concluding remarks will be drawn in Section 5.

2. Construction of a New Three-Step Class

Let the scalar function $f : D \subset \mathbb{R} \to \mathbb{R}$ and $f(\alpha) = 0 \neq f'(\alpha) = c_1$. In other words, α is a simple zero of $f(x) = 0$. In this section, we start with the three-step scheme:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= y_k - \frac{f(y_k)}{f'(y_k)}, \\ x_{k+1} &= z_k - \frac{f(z_k)}{f'(z_k)}, \end{aligned} \tag{1}$$

where $k = 0, 1, \ldots,$ is the iteration indicator. The order of convergence for (1) is eight but its computational efficiency is low. We substitute derivatives in all three steps by suitable approximations that use available data; thus we introduce the following approximations:

$$\begin{aligned} & f'(x_k) \approx f[x_k, w_k], \\ & \text{where } w_k = x_k + \beta f(x_k), \\ & f'(y_k) \approx \frac{f[y_k, w_k]}{H(u_k, v_k)}, \quad u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \\ & f'(z_k) \approx \frac{f[y_k, z_k] + f[w_k, y_k, z_k](z_k - y_k)}{W(s_k)}, \\ & s_k = \frac{f(z_k)}{f(x_k)}, \end{aligned} \tag{2}$$

in the first, second, and third steps of (1), where H and W are weight functions. The following iterative family of three-step methods is obtained:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta f(x_k), \\ & \beta \in R, \\ z_k &= y_k - H(u_k, v_k)\frac{f(y_k)}{f[y_k, w_k]}, \quad u_k = \frac{f(y_k)}{f(x_k)}, \\ & v_k = \frac{f(y_k)}{f(w_k)}, \\ x_{k+1} &= z_k - W(s_k) \\ & \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)}, \\ & s_k = \frac{f(z_k)}{f(x_k)}. \end{aligned} \tag{3}$$

In the following theorem, we state suitable conditions for deriving a new optimal class without memory according to the Kung and Traub conjecture [2] (also known as K-T hypothesis).

Theorem 1. *Let $f : D \subset \mathbb{R} \to \mathbb{R}$ be a scalar function which has a simple root α in the open interval D, and also the initial approximation x_0 is sufficiently close to the simple zero. Then, the three-step iterative method* (3) *has eighth-order under the conditions $W(0) = W'(0) = 1$, $H(0,0) = H_u(0,0) = H_{uu}(0,0) = 1$, $H_v(0,0) = H_{vv}(0,0) = 0$, and $H_{uv}(0,0) = 2$ and satisfies the following error equation:*

$$\begin{aligned} e_{k+1} = {} & \left(1 + \beta f'(\alpha)\right)^4 c_2^2 c_3 \\ & \times \left(\left(9 + \beta f'(\alpha)\left(7 + \beta f'(\alpha)\right)\right) c_2^3 \right. \\ & \left. + 2c_2c_3 - c_4\right) e_k^8 + O\left(e_k^9\right). \end{aligned} \tag{4}$$

Proof. By using Taylor's expansion of $f(x)$ around α and taking into account that $f(\alpha) = 0$, we obtain

$$\begin{aligned} f(x) = f'(\alpha) & \left(e_k + c_2 e_k^2 + c_3 e_k^3 + c_4 e_k^4 \right. \\ & \left. + c_5 e_k^5 + c_6 e_k^6 + c_7 e_k^7 + c_8 e_k^8 + O\left(e_k^9\right)\right), \end{aligned} \tag{5}$$

where $e_k = x_k - \alpha$, $e_{k,w} = w_k - \alpha$, $e_{k,y} = y_k - \alpha$, $e_{k,z} = z_k - \alpha$, and $c_k = f^{(k)}(\alpha)/k!f'(\alpha)$. Therefore,

$$\begin{aligned} e_{k,w} = {} & \left(1 + \beta f'(\alpha)\right) e_k + \beta f'(\alpha) c_2 e_k^2 + \beta f'(\alpha) c_3 e_k^3 \\ & + \beta f'(\alpha) c_4 e_k^4 + \beta f'(\alpha) c_5 e_k^5 + \beta f'(\alpha) c_6 e_k^6 \\ & + \beta f'(\alpha) c_7 e_k^7 + \beta f'(\alpha) c_8 e_k^8 + O\left(e_k^9\right), \end{aligned} \tag{6}$$

$$\begin{aligned} f(w_k) = {} & f'(\alpha)\left(1 + f'(\alpha)\beta\right) e_k \\ & + f'(\alpha)\left(1 + f'(\alpha)\beta\left(3 + f'(\alpha)\beta\right)\right) c_2 e_k^2 \\ & + \cdots + O\left(e_k^9\right). \end{aligned} \tag{7}$$

Note that we used $\cdots$ in order to avoid writing further terms of the Taylor expansions due to symbolic computations. By using (5) and (7), we obtain

$$\begin{aligned} f[x_k, w_k] = {} & f'(\alpha)\left(1+\left(2+\beta f'(\alpha)\right)\right)c_2 e_k \\ & + f'(\alpha)\left(\beta f'(\alpha) c_2^2 + \left(3+\beta f'(\alpha)\right.\right. \\ & \left.\left.\times\left(3+\beta f'(\alpha)\right)\right)c_3\right)e_k^2 \\ & + \cdots + O\left(e_k^8\right). \end{aligned} \tag{8}$$

Dividing (5) by (8) gives us

$$\begin{aligned} \frac{f(x_k)}{f[x_k, w_k]} = {} & e_k - \left(1+\beta f'(\alpha)\right)c_2 e_k^2 \\ & + \left(\left(2+\beta f'(\alpha)\left(2+\beta f'(\alpha)\right)\right)c_2^2\right. \\ & \left. - \left(1+\beta f'(\alpha)\right)\left(2+\beta f'(\alpha)\right)c_3\right)e_k^3 \\ & + \cdots + O\left(e_k^9\right). \end{aligned} \tag{9}$$

And thus,

$$e_{k,y} = -\left(1+\beta f'(\alpha)\right)^2 c_2 c_3 e_k^4 + \cdots + O\left(e_k^9\right). \tag{10}$$

Subsequently, we have

$$\begin{aligned} f[y_k, w_k] = {} & f'(\alpha) + f'(\alpha)\left(1+\beta f'(\alpha)\right)c_2 e_k \\ & + f'(\alpha)\left(\left(1+2\beta f'(\alpha)\right)c_2^2\right. \\ & \left. + \left(1+\beta f'(\alpha)\right)^2 c_3\right)e_k^2 \\ & + \cdots + O\left(e_k^8\right). \end{aligned} \tag{11}$$

Thus,

$$\begin{aligned} h(u_k, v_k)\frac{f(y_k)}{f[y_k, w_k]} = {} & \left(1+\beta f'(\alpha)\right)c_2 e_k^2 \\ & + \left(-\left(2+\beta f'(\alpha)\left(2+\beta f'(\alpha)\right)\right)c_2^2\right. \\ & \left. + \left(1+\beta f'(\alpha)\right)\left(2+\beta f'(\alpha)\right)c_3\right) \\ & \times e_k^3 + \cdots + O\left(e_k^9\right), \\ f[w_k, y_k, z_k] = {} & f'(\alpha)c_2 + f'(\alpha)\left(1+\beta f'(\alpha)\right)c_3 e_k \\ & + f'(\alpha)\left(\left(1+2\beta f'(\alpha)\right)c_2 c_3\right. \\ & \left. + \left(1+\beta f'(\alpha)\right)^2 c_4\right)e_k^2 \\ & + \cdots + O\left(e_k^9\right). \end{aligned} \tag{12}$$

Therefore, we attain

$$\begin{aligned} & \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)} \\ & \quad = -\left(1+\beta f'(\alpha)\right)^2 c_2 c_3 e_k^4 + \cdots + O\left(e_k^9\right). \end{aligned} \tag{13}$$

Finally, according to the above analysis, the general error equation is given by

$$\begin{aligned} e_{k+1} = {} & e_{k,z} - W(s_k)\frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)} \\ = {} & \left(1+\beta f'(\alpha)\right)^4 c_2^2 c_3 \\ & \times\left(\left(9+\beta f'(\alpha)\left(7+\beta f'(\alpha)\right)\right)c_2^3\right. \\ & \left. + 2c_2 c_3 - c_4\right)e_k^8 + O\left(e_k^9\right), \end{aligned} \tag{14}$$

so that the proof of the theorem is finished. □

We provide some specific weight functions that satisfy the conditions of Theorem 1 as follows:

$$\begin{aligned} H_1(u_k, v_k) &= 1 + u_k + 2u_k v_k + u_k^2, \\ H_2(u_k, v_k) &= \frac{1}{1 - u_k - 2u_k v_k}, \\ W_1(s_k) &= \cos(s_k) + \sin(s_k), \\ W_2(s_k) &= \frac{1}{1 - s_k}, \\ W_3(s_k) &= 1 + s_k, \\ W_4(s_k) &= e^{s_k}. \end{aligned} \tag{15}$$

We consider these weight functions in, without and with memory methods, (3) and (18) in the forthcoming sections.

There are some measures for comparing various iterative techniques. Traub [1] introduced the informational efficiency and efficiency index, which can be expressed in terms of the order (r) of the method and the number of function (and derivative) evaluations (ρ). In fact, the efficiency index (or computational efficiency) is given by $E = r^{1/\rho}$.

Clearly, the efficiency index of the proposed optimal class of method is $8^{1/4} \approx 1.682$ which is optimal in the sense of K-T hypothesis and is higher than two- or one-step methods without memory.

It is worth emphasizing that the maximal order of convergence is not the only goal in constructing root-finding methods and, consequently, the ultimate measure of efficiency of the designed method. Complexity of the formulae involved, often called combinatorial cost, makes another important parameter, which should be taken into account. Hence, we wish to construct a new method with memory possessing a high order 12 requiring only 4 functional evaluations (just like (3)).

In the next section, we will modify the proposed method and introduce a new method. We use an accelerator parameter to increase the order of convergence significantly.

3. Construction of a Method with Memory

Error equation (14) indicates that the order of convergence for class (3) is equal to eight. This section is concerned with extracting an efficiency with memory method from (3) since its error equation contains the parameter β which can be approximated in such a way that increases the local order of convergence.

We set $\beta = \beta_k$ as the iteration proceeds by the formula $\beta_k = -1/\overline{f'}(\alpha)$ for $k = 1, 2, \ldots$, where $\overline{f'}(\alpha)$ is an approximation of $f'(\alpha)$. We have a method via the following forms of β_k:

$$\beta_k = -\frac{1}{\overline{f'}(\alpha)} = -\frac{1}{N_4'(x_k)}. \tag{16}$$

The key idea that provides the order acceleration lies in a special form of the error relation and a convenient choice of a free parameter. We define a self-accelerating parameter, which is calculated during the iterative process using Newton's interpolating polynomial.

Hence, we consider Newton's interpolation as the method for approximating $f'(\alpha)$, where $N_4(t)$ is Newton's interpolation polynomial of fourth degree, set through five available approximations $x_k, z_{k-1}, y_{k-1}, w_{k-1}, x_{k-1}$ as follows:

$$\begin{aligned}
N_4'(x_k) &= \left[\frac{d}{dt}N_4(t)\right]_{t=x_k} \\
&= \left[\frac{d}{dt}\left(f(x_k) + f[x_k, z_{k-1}](t - x_k)\right.\right. \\
&\qquad + f[x_k, z_{k-1}, y_{k-1}](t - x_k) \\
&\qquad \times (t - z_{k-1}) + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}](t - x_k) \\
&\qquad \times (t - z_{k-1})(t - y_{k-1}) \\
&\qquad + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}, w_{k-1}] \\
&\qquad \times (t - x_k)(t - z_{k-1})(t - y_{k-1}) \\
&\qquad \left.\left. \times\ (t - x_{k-1}))\right]\right._{t=x_k} \\
&= f[x_k, z_{k-1}] + f[x_k, z_{k-1}, y_{k-1}](x_k - z_{k-1}) \\
&\quad + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}](x_k - z_{k-1})(x_k - y_{k-1}) \\
&\quad + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}, w_{k-1}](x_k - z_{k-1}) \\
&\quad \times (x_k - y_{k-1})(x_k - x_{k-1}).
\end{aligned} \tag{17}$$

Here, the with memory development of (3) can be presented as follows:

$$\begin{aligned}
y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k), \\
z_k &= y_k - H(u_k, v_k)\frac{f(y_k)}{f[y_k, w_k]}, \quad u_k = \frac{f(y_k)}{f(x_k)}, \\
&\qquad\qquad v_k = \frac{f(y_k)}{f(w_k)}, \\
x_{k+1} &= z_k - W(s_k) \\
&\quad \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)} \\
&\qquad\qquad s_k = \frac{f(z_k)}{f(x_k)}.
\end{aligned} \tag{18}$$

We attempt to prove that the method with memory (18) has convergence order twelve provided that we use accelerator β_k as in (16). It should be remarked that we have applied the Herzberger's matrix method [7].

Theorem 2. *If an initial approximation x_0 is sufficiently close to the zero α of $f(x)$ and the parameter β_k in the iterative scheme* (18) *is recursively calculated by the forms given in* (16), *then the order of convergence is twelve.*

Proof. We will use Herzberger's matrix method to determine the order of convergence. Note that the lower bound of order for a single-step s-point method $x_k = G(x_{k-1}, x_{k-2}, \ldots, x_{k-s})$ is the spectral radius of a matrix $M^{(s)} = (m_{ij})$, associated to the method with elements:

$$\begin{aligned}
m_{1,j} &= \text{amount of information required at point } x_{k-j} \\
&\qquad j = 1, 2, \ldots, s, \\
m_{i,i-1} &= 1 \quad (i = 2, 3, \ldots, s), \\
m_{i,j} &= 0, \quad \text{otherwise}.
\end{aligned} \tag{19}$$

On the other hand, the lower bound of order of an s-step method $G = G_1 \circ G_2 \circ \cdots \circ G_s$ is the spectral radius of the product of matrices $M = M_1 \cdot M_2 \cdots M_s$.

We can express each approximation x_{k+1}, z_k, y_k, and w_k as a function of available information $f(z_k)$, $f(y_k)$, $f(w_k)$, and $f(x_k)$ from the kth iteration and $f(z_{k-1})$, $f(y_{k-1})$, $f(w_{k-1})$, and $f(x_{k-1})$ from the previous iteration, depending on the accelerating technique. Now, we determine the order of convergence for (18) applied for the calculation of β_k.

Method (N4). We use the following matrices to express informational dependence:

$$
\begin{aligned}
&x_{k+1} = \varphi_1\left(z_k, y_k, w_k, x_k, z_{k-1}\right) \longrightarrow M_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \\
&z_k = \varphi_2\left(y_k, w_k, x_k, z_{k-1}, y_{k-1}\right) \longrightarrow M_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \\
&y_k = \varphi_3\left(w_k, x_k, z_{k-1}, y_{k-1}, w_{k-1}\right) \longrightarrow M_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \\
&w_k = \varphi_4\left(x_k, z_{k-1}, y_{k-1}, w_{k-1}, x_{k-1}\right) \longrightarrow M_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.
\end{aligned}
\tag{20}
$$

Hence, it is easy to derive

$$
M^{(N4)} = M_1 M_2 M_3 M_4 = \begin{bmatrix} 8 & 4 & 4 & 4 & 4 \\ 4 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{21}
$$

with the eigenvalues $\{12, 0, 0, 0, 0\}$. Consequently, the order of the method with memory (18)-($N4$) is at least twelve. The proof of Theorem 2 is finished. □

Clearly, the proposed with memory scheme possesses a high computational efficiency index $12^{1/4} \approx 1.861$, which makes it interesting for practical problems.

4. Numerical Examples

In this section, we test our proposed methods and compare their results with some other methods of the same order of convergence. The errors $|x_k - \alpha|$ denote approximations to the sought zeros, and $a(-b)$ stands for $a \times 10^{-b}$. Moreover, coc indicates computational order of convergence and is computed by

$$
\text{coc} = \frac{\log\left(\left|f\left(x_k\right)/f\left(x_{k-1}\right)\right|\right)}{\log\left(\left|f\left(x_{k-1}\right)/f\left(x_{k-2}\right)\right|\right)}. \tag{22}
$$

The calculated value coc estimates the theoretical order of convergence well when "pathological behavior" of the iterative method (for instance, slow convergence at the beginning of the implemented iterative method, "oscillating" behavior of approximations, etc.) does not exist.

We have used 1000-fixed floating point arithmetic so as to minimize the effect of round-off errors.

By using weight functions (15), we introduce some concrete methods based on the proposed class. Note that it is assumed that the initial estimate β_0 should be chosen before starting the iterative process and also x_0 is given suitably.

Concrete method 1:

$$
\begin{aligned}
y_k &= x_k - \frac{f\left(x_k\right)}{f\left[x_k, w_k\right]}, \quad w_k = x_k + \beta_k f\left(x_k\right), \\
z_k &= y_k - \left(1 + u_k + 2u_k v_k + u_k^2\right) \frac{f\left(y_k\right)}{f\left[y_k, w_k\right]}, \\
& \qquad u_k = \frac{f\left(y_k\right)}{f\left(x_k\right)}, \quad v_k = \frac{f\left(y_k\right)}{f\left(w_k\right)}, \\
x_{k+1} &= z_k - \left(\cos\left(s_k\right) + \sin\left(s_k\right)\right) \\
& \quad \times \frac{f\left(z_k\right)}{f\left[z_k, y_k\right] + f\left[w_k, y_k, z_k\right]\left(z_k - y_k\right)}, \\
& \qquad s_k = \frac{f\left(z_k\right)}{f\left(x_k\right)}.
\end{aligned}
\tag{23}
$$

Concrete method 2:

$$
\begin{aligned}
y_k &= x_k - \frac{f\left(x_k\right)}{f\left[x_k, w_k\right]}, \quad w_k = x_k + \beta_k f\left(x_k\right), \\
z_k &= y_k - \left(1 + u_k + 2u_k v_k + u_k^2\right) \frac{f\left(y_k\right)}{f\left[y_k, w_k\right]}, \\
& \qquad u_k = \frac{f\left(y_k\right)}{f\left(x_k\right)}, \quad v_k = \frac{f\left(y_k\right)}{f\left(w_k\right)}, \\
x_{k+1} &= z_k - \left(\frac{1}{1 - s_k}\right) \\
& \quad \times \frac{f\left(z_k\right)}{f\left[z_k, y_k\right] + f\left[w_k, y_k, z_k\right]\left(z_k - y_k\right)}, \\
& \qquad s_k = \frac{f\left(z_k\right)}{f\left(x_k\right)}.
\end{aligned}
\tag{24}
$$

Concrete method 3:

$$
y_k = x_k - \frac{f\left(x_k\right)}{f\left[x_k, w_k\right]}, \quad w_k = x_k + \beta_k f\left(x_k\right),
$$

$$\begin{aligned}
z_k &= y_k - \left(1 + u_k + 2u_k v_k + u_k^2\right) \frac{f(y_k)}{f[y_k, w_k]}, \\
u_k &= \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \\
x_{k+1} &= z_k - (1 + s_k) \\
&\quad \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)}, \\
s_k &= \frac{f(z_k)}{f(x_k)}.
\end{aligned} \tag{25}$$

Concrete method 4:

$$\begin{aligned}
y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k), \\
z_k &= y_k - \left(1 + u_k + 2u_k v_k + u_k^2\right) \frac{f(y_k)}{f[y_k, w_k]}, \\
u_k &= \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \\
x_{k+1} &= z_k - \left(e^{s_k}\right) \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)}, \\
s_k &= \frac{f(z_k)}{f(x_k)}.
\end{aligned} \tag{26}$$

Concrete method 5:

$$\begin{aligned}
y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k), \\
z_k &= y_k - \left(\frac{1}{1 - u_k - 2u_k v_k}\right) \frac{f(y_k)}{f[y_k, w_k]}, \\
u_k &= \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \\
x_{k+1} &= z_k - (\cos(s_k) + \sin(s_k)) \\
&\quad \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)} \\
s_k &= \frac{f(z_k)}{f(x_k)}.
\end{aligned} \tag{27}$$

Concrete method 6:

$$\begin{aligned}
y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k), \\
z_k &= y_k - \left(\frac{1}{1 - u_k - 2u_k v_k}\right) \frac{f(y_k)}{f[y_k, w_k]}, \\
u_k &= \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \\
x_{k+1} &= z_k - \left(\frac{1}{1 - s_k}\right) \\
&\quad \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)}, \\
w_k &= \frac{f(z_k)}{f(x_k)}.
\end{aligned} \tag{28}$$

Concrete method 7:

$$\begin{aligned}
y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k), \\
z_k &= y_k - \left(\frac{1}{1 - u_k - 2u_k v_k}\right) \frac{f(y_k)}{f[y_k, w_k]}, \\
u_k &= \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \\
x_{k+1} &= z_k - (1 + s_k) \\
&\quad \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)}, \\
w_k &= \frac{f(z_k)}{f(x_k)}.
\end{aligned} \tag{29}$$

Concrete method 8:

$$\begin{aligned}
y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k), \\
z_k &= y_k - \left(\frac{1}{1 - u_k - 2u_k v_k}\right) \frac{f(y_k)}{f[y_k, w_k]}, \\
u_k &= \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \\
x_{k+1} &= z_k - \left(e^{s_k}\right) \\
&\quad \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)}, \\
w_k &= \frac{f(z_k)}{f(x_k)}.
\end{aligned} \tag{30}$$

Several iterative methods (IM) of optimal order eight, which also require four function evaluations, for comparisons with our proposed methods have been chosen.

Table 1: Consider $f_1(x) = \sin(\pi x)e^{(x^2+x\cos(x)-1)} + x\log(x\sin(x)+1)$, $\alpha = 0$, $x_0 = 0.6$, $\beta = -0.01$.

Methods without memory	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	coc
Our method (23)	5.3810(−3)	5.5392(−25)	6.9091(−201)	8.0003
Our method (24)	2.1802(−3)	3.9973(−28)	5.0816(−226)	8.0001
Our method (25)	4.2865(−3)	8.9612(−26)	3.2420(−207)	8.0002
Our method (26)	3.1912(−3)	8.4384(−27)	2.0042(−215)	8.0002
Our method (27)	5.7030(−3)	4.9625(−25)	1.6203(−201)	8.0002
Our method (28)	2.3293(−3)	3.8278(−28)	2.0304(−226)	8.0001
Our method (29)	5.7030(−3)	4.9625(−25)	1.6203(−201)	8.0002
Our method (30)	4.5488(−3)	8.1186(−26)	8.3149(−208)	8.0002
(34), $\beta = -0.01$	9.7732(−3)	4.3543(−24)	6.6317(−195)	8.0002
(35), $\beta = -0.01$	1.7342(−2)	1.1724(−21)	3.5692(−175)	8.0085

Three-step method by Wang et al. [8]:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k,w_k]}, \quad w_k = x_k + \beta f(x_k), \\ z_k &= y_k - \frac{f(y_k)}{f[y_k,w_k]}H(t_k), \quad t_k = \frac{f(y_k)}{f(x_k)}, \\ x_{k+1} &= z_k - \frac{f(z_k)}{f[z_k,w_k]}G(t_k,s_k), \quad s_k = \frac{f(z_k)}{f(y_k)}, \end{aligned} \tag{31}$$

where functions H and G are following $H_1(t) = 1+t$, $G_1(t,s) = 1+t+s+2ts-(1+\lambda)t^3$, and $\lambda = 1/(1+\beta f[x,w])$.

Three-step method by Lotfi and Tavakoli [9]:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n,w_n]}, \\ z_n &= y_n - H(t_n,u_n)\frac{f(y_n)}{f[y_n,w_n]}, \quad t_n = \frac{f(y)}{f(x)}, \\ & \qquad u_n = \frac{f(w)}{f(x)}, \\ x_{k+1} &= z_n - G(t_n,s_n)W(v_n,s_n)\frac{f(z_n)}{f[z_n,w_n]}, \\ & \qquad s_n = \frac{f(z)}{f(y)}, \quad v_n = \frac{f(z)}{f(x)}, \end{aligned} \tag{32}$$

where

$$\begin{aligned} W(s_n,v_n) &= 1+s_n^2+v_n^2, \\ G(t_n,s_n) &= 1+t_n+s_n+2t_ns_n+(-1-\phi_n)t_n^3, \\ & \left(\phi_n = \frac{1}{1+\beta_n f[x_n,w_n]}\right), \\ H(t_n,u_n) &= 1+t_n. \end{aligned} \tag{33}$$

Derivative-free Kung-Traub's family [2]:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k,w_k]}, \quad w_k = x_k + \beta f(x_k), \\ z_k &= y_k - \frac{f(y_k)f(w_k)}{[f(w_k)-f(y_k)]f[x_k,y_k]}, \\ x_{k+1} &= z_k - \frac{f(y_k)f(w_k)(y_k - x_k + f(x_k)/f[x_k,z_k])}{[f(y_k)-f(z_k)][f(w_k)-f(z_k)]} \\ &\quad + \frac{f(y_k)}{f[y_k,z_k]}. \end{aligned} \tag{34}$$

Three-step methods made by Zheng et al. [10]:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k,w_k]}, \quad w_k = x_k + \beta f(x_k), \\ z_k &= y_k - \frac{f(y_k)}{f[y_k,w_k] + f[y_k,x_k,w_k](y_k-x_k)}, \\ x_{k+1} &= z_k - \Big(f(z_k)\times\big(f[z_k,y_k] + f[z_k,y_k,x_k](z_k-y_k) \\ &\quad + f[z_k,y_k,x_k,w_k](z_k-y_k) \\ &\quad \times (z_k-x_k)\big)^{-1}\Big). \end{aligned} \tag{35}$$

In Tables 1, 3, and 5 our without memory proposed method by different weight functions (23)–(30) has been compared with optimal three-point methods (34) and (35), and we observe that all these methods behave very well practically and confirm their theoretical results.

Also Tables 2, 4, and 6 present numerical results for our with memory classes (23)–(30). It is also clear that all these methods behave very well practically and confirm their relevant theories. They all provide wherea bout twelve of convergence order.

Table 2: Consider $f_1(x) = \sin(\pi x)e^{(x^2+x\cos(x)-1)} + x\log(x\sin(x)+1)$, $\alpha = 0$, $x_0 = 0.6$, $\beta_0 = -0.01$.

Methods with memory	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	coc
Our method (23)	5.3810(−3)	2.7788(−35)	1.8668(−426)	12.1156
Our method (24)	2.1802(−3)	3.0336(−38)	5.3469(−462)	12.1571
Our method (25)	4.2865(−3)	5.2041(−36)	3.4739(−435)	12.1273
Our method (26)	3.1912(−3)	5.6445(−37)	9.2065(−447)	12.1411
Our method (27)	5.7030(−3)	4.6138(−35)	8.1591(−424)	12.1138
Our method (28)	2.3293(−3)	5.4580(−38)	6.1228(−459)	12.1556
Our method (29)	4.5488(−3)	8.7828(−36)	1.8467(−432)	12.1256
Our method (30)	3.3939(−3)	9.7470(−37)	6.4438(−444)	12.1395
(31), $\beta = -1$	2.1170(−3)	1.1077(−41)	3.4407(−501)	12.0035
(32), $\beta = -1$	5.7578(−3)	2.7125(−39)	1.4859(−474)	11.9819

Table 3: Consider $f_2(x) = e^{-5x}(x-2)(x^{10}+x+2)$, $\alpha = 2$, $x_0 = 2.2$, $\beta = -1$.

Methods without memory	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	coc
Our method (23)	5.4211(−6)	7.6321(−54)	1.1776(−436)	8.0000
Our method (24)	5.4981(−6)	8.5429(−54)	2.9020(−436)	8.0000
Our method (25)	5.4468(−6)	7.9261(−54)	1.5935(−436)	8.0001
Our method (26)	5.4725(−6)	8.2301(−54)	2.1533(−436)	8.0000
Our method (27)	3.2606(−6)	1.3070(−55)	8.7125(−451)	8.0000
Our method (28)	3.3757(−6)	1.7249(−55)	8.0153(−450)	8.0000
Our method (29)	3.2991(−6)	1.4354(−55)	1.8434(−450)	8.0000
Our method (30)	3.3375(−6)	1.5747(−55)	3.8666(−450)	8.0000
(34), $\beta = -1$	8.0552(−6)	1.4131(−52)	1.2677(−456)	8.0000
(35), $\beta = -0.01$	3.9453(−6)	1.2705(−59)	5.0021(−491)	8.0085

Table 4: Consider $f_2(x) = e^{-5x}(x-2)(x^{10}+x+2)$, $\alpha = 2$, $x_0 = 2.2$, $\beta_0 = -1$.

Methods with memory	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	coc
Our method (23)	5.4211(−6)	2.1857(−80)	4.4911(−970)	11.9590
Our method (24)	5.4981(−6)	2.4529(−80)	1.7931(−969)	11.9587
Our method (25)	5.4468(−6)	2.2719(−80)	7.1443(−970)	11.9589
Our method (26)	5.4725(−6)	2.3611(−80)	9.2065(−969)	11.9588
Our method (27)	5.4981(−6)	2.4529(−80)	1.7931(−969)	11.9587
Our method (28)	3.3757(−6)	7.7624(−82)	1.8084(−987)	11.9732
Our method (29)	3.2991(−6)	6.4445(−82)	1.9393(−988)	11.9737
Our method (30)	3.3375(−6)	7.0781(−82)	5.9748(−988)	11.9735
(31), $\beta = -1$	4.9425(−5)	1.5275(−69)	4.1256(−874)	12.0535
(32), $\beta = -1$	4.0518(−6)	8.2730(−77)	2.4267(−861)	11.0982

Table 5: Consider $f_3(x) = e^{x^3-x} - \cos(x^2-1) + x^3 + 1$, $\alpha = -1$, $x_0 = -1.65$, $\beta = -1$.

Methods without memory	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	coc
Our method (23)	6.0256(−3)	4.0898(−24)	1.8272(−193)	8.0001
Our method (24)	5.9797(−3)	3.8471(−24)	1.1202(−193)	8.0001
Our method (25)	6.0104(−3)	4.0081(−24)	1.5548(−193)	8.0001
Our method (26)	5.9952(−3)	3.9278(−24)	1.3225(−193)	8.0001
Our method (27)	5.3628(−3)	1.6071(−24)	1.0388(−196)	8.0001
Our method (28)	5.3357(−3)	1.5431(−24)	7.5042(−197)	8.0001
Our method (29)	5.3537(−3)	1.5856(−24)	9.3292(−197)	8.0001
Our method (30)	5.3448(−3)	1.5644(−24)	8.3766(−197)	8.0001
(34), $\beta = -1$	2.9152(−4)	5.3239(−34)	6.5827(−272)	8.0000
(35), $\beta = -0.01$	6.4036(−5)	4.4335(−44)	2.4457(−357)	7.9995

Table 6: Consider $f_3(x) = e^{x^3-x} - \cos(x^2-1) + x^3 + 1, \alpha = -1, x_0 = -1.65, \beta_0 = -1$.

Methods with memory	$\|x_1 - \alpha\|$	$\|x_2 - \alpha\|$	$\|x_3 - \alpha\|$	coc
Our method (23)	6.0256(−3)	4.7294(−38)	3.8067(−459)	11.9952
Our method (24)	5.9797(−3)	4.4345(−38)	1.7580(−459)	11.9955
Our method (25)	6.0104(−3)	4.6300(−38)	2.9502(−459)	11.9953
Our method (26)	5.9959(−3)	4.5324(−38)	2.2850(−459)	11.9954
Our method (27)	5.3628(−3)	9.5497(−39)	1.7425(−467)	11.9929
Our method (28)	5.3357(−3)	9.1483(−39)	1.0410(−467)	11.9931
Our method (29)	5.3538(−3)	9.4151(−39)	1.4696(−467)	11.9929
Our method (30)	5.3448(−3)	9.2821(−39)	1.2391(−467)	11.9930
(31), $\beta = -1$	1.3000(−2)	5.5214(−33)	1.2114(−401)	12.1381
(32), $\beta = -1$	1.0937(−2)	3.5221(−34)	1.7174(−415)	12.1081

5. Concluding Remarks

We have constructed a class of methods without and with memory. Our proposed methods do not need any derivative and therefore are applicable to nonsmooth functions too. Another advantage of the proposed methods is that their without memory versions are optimal in the sense of K-T conjecture. In addition, it contains an accelerator parameter which rises convergence order from eight to twelve without any new functional evaluations. In other words, the efficiency index of the with memory class is $12^{1/4} \approx 1.861$.

We finalize this work by suggesting some outlines for future research: first developing the proposed methods for computing multiple roots and second exploring its dynamic or basins of attractions, and finally we wonder why not to use an adaptive arithmetic in each step of the iterative method instead of using a fixed precision, since this higher precision is only necessary in the last step of the iterative process.

Conflict of Interests

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

Acknowledgments

The first three authors thank Islamic Azad University, Hamedan Branch, for the financial support of the present research. Also, the authors would like to express their sincere appreciations to Dr. Soleymani and the respected referees for their constructive comments regarding improving this work.

References

[1] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, New York, NY, USA, 1964.

[2] 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.

[3] S. Artidiello, A. Cordero, J. R. Torregrosa, and M. P. Vassileva, “Optimal high-order methods for solving nonlinear equations,” *Journal of Applied Mathematics*, vol. 2014, Article ID 591638, 9 pages, 2014.

[4] B. Ignatova, N. Kyurkchiev, and A. Iliev, “Multipoint algorithms arising from optimal in the sense of Kung-Traub iterative procedures for numerical solution of nonlinear equations,” *General Mathematics Notes*, vol. 6, pp. 45–79, 2011.

[5] N. Kyurkchiev and A. Iliev, “On some multipoint methods arising from optimal in the sense of Kung-Traub algorithms,” *Biomath*, vol. 2, no. 1, pp. 1–7, 2013.

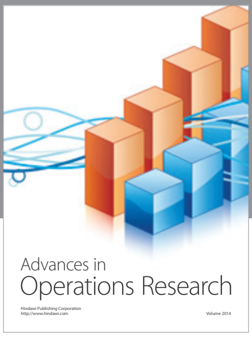
[6] X. Wang and T. Zhang, “A new family of Newton-type iterative methods with and without memory for solving nonlinear equations,” *Calcolo*, vol. 51, no. 1, pp. 1–15, 2014.

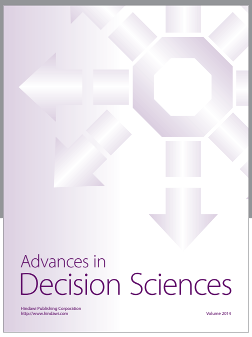
[7] J. Herzberger, “Über Matrixdarstellungen für Iterationverfahren bei nichtlinearen Gleichungen,” *Computing*, vol. 12, no. 3, pp. 215–222, 1974.

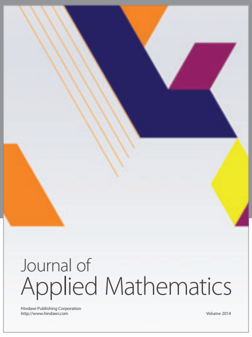
[8] X. Wang, J. Džunić, and T. Zhang, “On an efficient family of derivative free three-point methods for solving nonlinear equations,” *Applied Mathematics and Computation*, vol. 219, no. 4, pp. 1749–1760, 2012.

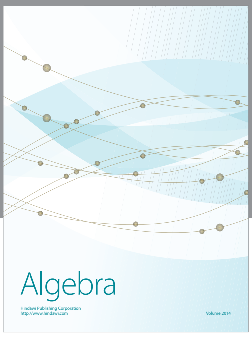
[9] T. Lotfi and E. Tavakoli, “On a new efficient Steffensen-like iterative class by applying a suitable self-accelerator parameter,” *The Scientific World Journal*, vol. 2014, Article ID 769758, 9 pages, 2014.

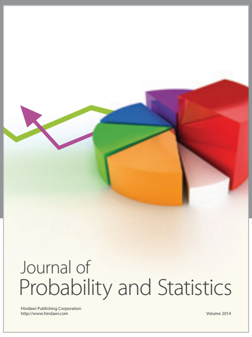
[10] Q. Zheng, J. Li, and F. Huang, “An optimal Steffensen-type family for solving nonlinear equations,” *Applied Mathematics and Computation*, vol. 217, no. 23, pp. 9592–9597, 2011.


Advances in
Operations Research

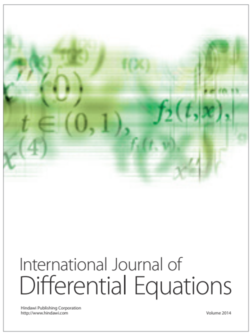

Advances in
Decision Sciences


Journal of
Applied Mathematics

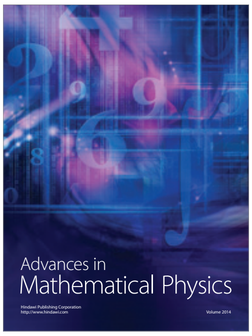

Algebra

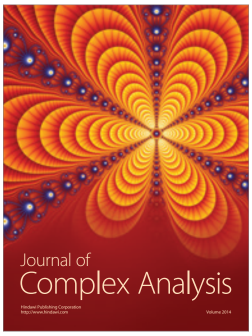

Journal of
Probability and Statistics

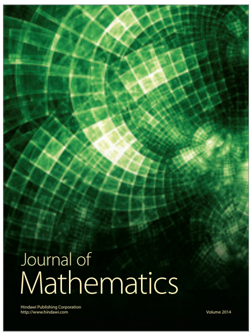
The Scientific
World Journal

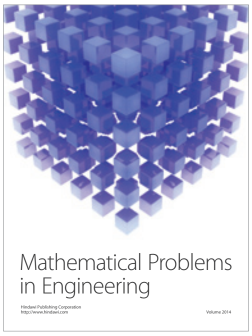

International Journal of
Differential Equations

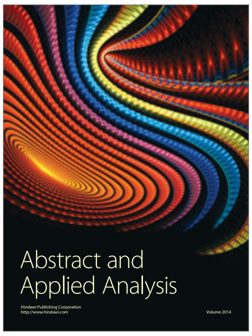
International Journal of
Combinatorics

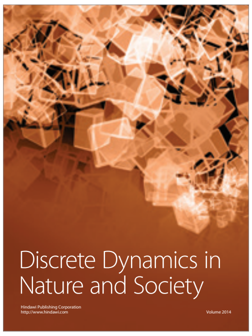

Advances in
Mathematical Physics

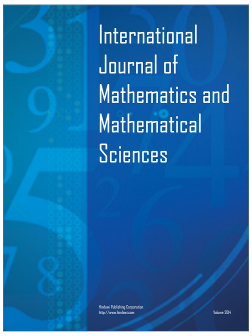

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

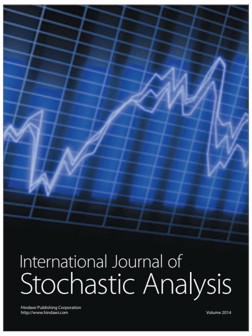

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization