*Research Article*

# Two Optimal Eighth-Order Derivative-Free Classes of Iterative Methods

## F. Soleymani[1] and S. Shateyi[2]

[1] *Department of Mathematics, Islamic Azad University, Zahedan Branch, Zahedan, Iran*
[2] *Department of Mathematics, University of Venda, Private Bag X5050, Thohoyandou 0950, South Africa*

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

Optimization problems defined by (objective) functions for which derivatives are unavailable or available at an expensive cost are emerging in computational science. Due to this, the main aim of this paper is to attain as high as possible of local convergence order by using fixed number of (functional) evaluations to find efficient solvers for one-variable nonlinear equations, while the procedure to achieve this goal is totally free from derivative. To this end, we consider the fourth-order uniparametric family of Kung and Traub to suggest and demonstrate two classes of three-step derivative-free methods using only four pieces of information per full iteration to reach the optimal order eight and the optimal efficiency index 1.682. Moreover, a large number of numerical tests are considered to confirm the applicability and efficiency of the produced methods from the new classes.

## 1. Introduction

This paper focuses on finding approximate solutions to nonlinear scalar and sufficiently smooth equations by derivative-free methods. Techniques such as the false position method in the root-finding of a nonlinear equation $f(x) = 0$, requires bracketing of the zero by two guesses. Such schemes are called bracketing methods. These methods are almost always convergent since they are based on reducing the interval between the two guesses so as to zero in on the root of the equation. In the Newton method, the zero is not bracketed. In fact, only one initial guess of the solution is needed to get the iterative process started in order to find the zero. The method hence falls in the category of open methods.

Convergence in open methods is not guaranteed but if the method does converge, it does so much faster than the bracketing methods [1]. Although Newton's iteration has widely been discussed and improved in the literature, see for example, [2, 3], one of the

main drawback, which is the matter of first derivative evaluation, occasionally restricts the application of this method or its variants. On the other hand, when we discuss improving the iterative without memory methods for solving one variable nonlinear equations, the conjecture of Kung and Traub [4] is taken automatically into consideration. It should be remarked that according to the unproved conjecture of Kung and Traub [4], the index of efficiency will not pass the maximum level $2^{(n-1)/n}$, wherein $n$ is the total number of (functional) evaluations.

Apart from being involved in the first derivative, Newton's iteration will use the second derivative of the function, when it is applied in *optimization problems* to final the local minima. Therefore, its use will be limited especially when the cost of first and second derivatives of the functions is expensive. And subsequently, derivative-free algorithms come to attention.

For the first time, Steffensen in [5] gave the following derivative-free form of Newton's iteration

$$x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n + f(x_n), \ n = 0, 1, 2, \ldots \tag{1.1}$$

which possesses the same rate of convergence and efficiency index as Newton's.

In this work, we suggest novel classes of three-step four-point iterative methods, which are without memory, derivative-free, optimal, and therefore consistent with hard problems. For this reason, the contents of the paper unfold as comes next. Section 2 reveals our main contribution as some generalizations to the fourth-order uniparametric family of Kung and Traub [4]. Subsequently, Section 3 gives a very short discussion on the available derivative-free methods in the literature. Section 4 discusses the implementation of the new produced methods from our classes on large number of numerical examples. And finally, a short conclusion will be given in Section 5.

## 2. Main Outcome

In order to contribute and provide a class of methods, which are derivative-free with high efficiency index, we take into consideration the optimal two-step fourth-order uniparametric family of Kung and Traub [4] in the first two steps of a three-step cycle in which the Newton's iteration has been performed in the last step as comes next

$$y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n + \beta f(x_n), \ \beta \in \mathbb{R} \setminus \{0\},$$

$$z_n = y_n - \frac{f(y_n)f(w_n)}{(f(w_n) - f(y_n))f[x_n, y_n]}, \tag{2.1}$$

$$x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}.$$

This scheme includes four evaluations of the function and one of its first-order derivative to reach the order eight and 1.516 as its efficiency index. To improve the efficiency

index, we first consider the same approximation as used in the second step of (2.1) to annihilate $f'(z_n)$ and then take advantage of weight function approach as follows:

$$y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n + \beta f(x_n),$$

$$z_n = y_n - \frac{f(y_n)f(w_n)}{(f(w_n) - f(y_n))f[x_n, y_n]},$$

$$x_{n+1} = z_n - \frac{f(z_n)f(w_n)}{(f(w_n) - f(y_n))f[x_n, y_n]} \{G(\varphi) \times H(\tau) \times Q(\sigma) \times L(\rho)\},$$

(2.2)

where $\beta \in \mathbb{R} \setminus \{0\}$, $\varphi = f(z)/f(y)$, $\tau = f(z)/f(w)$, $\sigma = f(z)/f(x)$, $\rho = f(y)/f(w)$, without the index $n$, and $G(\varphi)$, $H(\tau)$, $Q(\sigma)$, $L(\rho)$ are four real-valued weight functions, which force the order to arrive at the maximum level eight by using the fixed number of evaluations per cycle. Theorem 2.1 shows that (2.2) will arrive at local eighth-order convergence by using only four function evaluations per full cycle. This reveals that any method from our proposed class possesses 1.682 as the efficiency index, which is optimal according to the Kung and Traub conjecture, while it is fully free from derivative evaluations.

**Theorem 2.1.** *Let $\alpha \in D$ be a simple zero of a sufficiently differentiable function $f : D \subseteq \mathbb{R} \to \mathbb{R}$ and let that $c_j = f^{(j)}(\alpha)/j!$, $j \geq 1$. If $x_0$ is sufficiently close to $\alpha$, then, (i) the local order of convergence of the solution by the class of methods without memory defined in (2.2) is eight, when*

$$G(0) = G'(0) = 1, \quad |G''(0)| < \infty,$$

$$H(0) = H'(0) = 1,$$

$$Q(0) = 1, \quad Q'(0) = 1,$$

$$L(0) = 1, \quad L'(0) = L'''(0) = 0, \quad L''(0) = 2 + 2\beta f[x_n, w_n], \quad |L^{(4)}(0)| < \infty,$$

(2.3)

*and (ii) this solution reads the error equation*

$$e_{n+1} = x_{n+1} - \alpha$$

$$= -\frac{1}{24c_1^7}\left(c_2\left(2c_2^2 - c_1c_3\right)(1 + c_1\beta)^2\right.$$

$$\times \left(12(1 + c_1\beta)^2 \times \left(-2\left(7c_2^4 - 8c_1c_2^2c_3 + c_1^2c_3^2 + c_1^2c_2c_4\right)\right.\right.$$

$$\left.\left.\left. + \left(-2c_2^2 + c_1c_3\right)^2 G''(0)\right) + c_2^4 L^{(4)}(0)\right)\right)e_n^8 + O\left(e_n^9\right).$$

(2.4)

*Proof.* We expand any terms of (2.2) around the simple root $\alpha$ in the $n$th iterate by taking into account $e_n = x_n - \alpha$ and $f(\alpha) = 0$. Thus, we write $f(x_n) = c_1 e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + \cdots + O(e_n^9)$. Accordingly, we attain

$$
y_n - \alpha = c_2 \left( \frac{1}{c_1} + \beta \right) e_n^2 + \cdots + O\left(e_n^9\right).
\tag{2.5}
$$

In the same vein, we obtain

$$
z_n - \alpha = \frac{\left(2c_2^3 - c_1 c_2 c_3\right)\left(1 + c_1 \beta\right)^2}{c_1^3} e_n^4 - \frac{1}{c_1^4}\left(1 + c_1 \beta\right)
$$
$$
\times \left( c_1^2 c_3^2 (1 + c_1 \beta)(2 + c_1 \beta) + c_1^2 c_2 c_4 (1 + c_1 \beta)(2 + c_1 \beta) + c_2^4 (10 + c_1 \beta (11 + 5 c_1 \beta)) \right.
$$
$$
\left. - c_1 c_2^2 c_3 (14 + c_1 \beta(19 + 7 c_1 \beta)) \right) e_n^5 + \cdots + O\left(e_n^9\right).
\tag{2.6}
$$

We also have $f(z_n) = (1/c_1^2)(2c_2^3 - c_1 c_2 c_3)(1 + c_1 \beta)^2 e_n^4 + \cdots + O(e_n^9)$. Now using symbolic computation in the last step of (2.2), we attain

$$
z_n - \frac{f(z_n)f(w_n)}{\left(f(w_n) - f(y_n)\right)f\left[x_n, y_n\right]} - \alpha = \frac{\left(6c_2^5 - 5c_1 c_2^3 c_3 + c_1^2 c_2 c_3^2\right)\left(1 + c_1 \beta\right)^3}{c_1^5} e_n^6 + \cdots + O\left(e_n^9\right).
\tag{2.7}
$$

Furthermore, by considering (2.7) and the real-valued weight functions as in (2.3), we will obtain the error equation (2.4). This shows that the proposed class of derivative-free methods (2.2)-(2.3) reaches the optimal eighth-order convergence by using only four function evaluations per full iteration. This ends the proof. $\qquad\square$

Now, any optimal three-step four-point derivative-free without memory method can be produced by using (2.2)-(2.3). As an instance, we can have ($\beta = 0.01$)

$$
y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n + 0.01 f(x_n),
$$
$$
z_n = y_n - \frac{f(y_n)f(w_n)}{\left(f(w_n) - f(y_n)\right)f\left[x_n, y_n\right]},
$$

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

$$\times \left\{ \left( 1 + \frac{f(z_n)}{f(y_n)} \right) \left( 1 + \frac{f(z_n)}{f(w_n)} \right) \left( 1 + \frac{f(z_n)}{f(x_n)} \right) \right.$$

$$\left. \times \left( 1 + (1 + 0.01 f[x_n, w_n]) \left( \frac{f(y_n)}{f(w_n)} \right)^2 \right) \right\},$$

$$(2.8)$$

where its error equation reads

$$e_{n+1} = \frac{(1 + 0.01 c_1)^4 c_2 (2c_2^2 - c_1 c_3)(7c_2^4 - 8c_1 c_2^2 c_3 + c_1^2 c_3^2 + c_1^2 c_2 c_4)}{c_1^7} e_n^8 + O\left( e_n^9 \right). \qquad (2.9)$$

We should here recall that per computing step for any of the methods from the new class, the values of $f(w_n)/((f(w_n) - f(y_n))f[x_n, y_n])$ and $f[x_n, w_n]$ should be computed only once, and then their values will be considered in the rest of the cycle wherever required.

Many of the nonlinear functions are arising from solving complex environmental engineering problems, where the objective depends on the output of a numerical simulation of a physical process. These simulators are expensive to evaluate because they involve numerically solving systems of partial differential equations governing the underlying physical phenomena. However, function evaluation remains the dominant expense in optimization problems since the savings in time are often offset by increased accuracy of the simulation. For these reasons, algorithms like (2.2)-(2.3) in which there is no need of derivative evaluation, the order is optimal as well as the efficiency index is high are important in hard problems.

Before going to the next sections, it is required to have a discussion on another similar class of derivative-free methods that is attainable by choosing a different approximation in the first step of (2.1) and also with a somehow different weight functions in the last step. Considering these in (2.1), we can have the following without memory derivative-free class

$$y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n - \beta f(x_n),$$

$$z_n = y_n - \frac{f(y_n)f(w_n)}{(f(w_n) - f(y_n))f[x_n, y_n]}, \qquad (2.10)$$

$$x_{n+1} = z_n - \frac{f(z_n)f(w_n)}{(f(w_n) - f(y_n))f[x_n, y_n]} \{ G(\varphi) \times H(\tau) \times Q(\sigma) \times L(\rho) \},$$

which its order is given in Theorem 2.2 to be eight by choosing appropriate weight functions. Noting that $\beta \in \mathbb{R} \setminus \{0\}$, $\varphi = f(z)/f(y)$, $\tau = f(z)/f(w)$, $\sigma = f(z)/f(x)$, $\rho = f(y)/f(w)$, without the index $n$, $G(\varphi)$, $H(\tau)$, $Q(\sigma)$, and $L(\rho)$ are four real-valued weight functions.

**Theorem 2.2.** *Let $\alpha \in D$ be a simple zero of a sufficiently differentiable function $f : D \subseteq \mathbb{R} \to \mathbb{R}$ and let that $c_j = f^{(j)}(\alpha)/j!, j \geq 1$. If $x_0$ is sufficiently close to $\alpha$, then, (i) the local order of convergence of the solution by the class of methods without memory defined in* (2.10) *is eight, when*

$$G(0) = G'(0) = 1, \qquad |G''(0)| < \infty,$$
$$H(0) = 1, \quad H'(0) = 2 - \beta f[x_n, w_n],$$
$$Q(0) = 1, \qquad Q'(0) = 0, \tag{2.11}$$
$$L(0) = 1, \qquad L'(0) = L'''(0) = 0, \qquad L''(0) = 2 - 2\beta f[x_n, w_n], \qquad \left|L^{(4)}(0)\right| < \infty,$$

*and (ii) this solution reads the error equation*

$$e_{n+1} = x_{n+1} - \alpha$$

$$= -\frac{1}{24c_1^7}\left(c_2\left(2c_2^2 - c_1 c_3\right)(-1 + c_1\beta)^2\right.$$

$$\times \left(12(-1 + c_1\beta)^2\left(-2\left(7c_2^4 - 8c_1 c_2^2 c_3 + c_1^2 c_3^2 + c_1^2 c_2 c_4\right)\right.\right.$$

$$\left.\left.\left. + \left(-2c_2^2 + c_1 c_3\right)^2 G''(0)\right) + c_2^4 L^{(4)}(0)\right)\right)e_n^8 + O\left(e_n^9\right). \tag{2.12}$$

*Proof.* The proof of this theorem is similar to the proof of Theorem 2.1. It is hence omitted. $\square$

Now, by using (2.10)-(2.11), we can have some other efficient optimal eighth-order without memory derivative-free methods, such as

$$y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n - f(x_n),$$

$$z_n = y_n - \frac{f(y_n)f(w_n)}{(f(w_n) - f(y_n))f[x_n, y_n]},$$

$$x_{n+1} = z_n - \frac{f(z_n)f(w_n)}{(f(w_n) - f(y_n))f[x_n, y_n]} \tag{2.13}$$

$$\times \left\{\left(1 + \frac{f(z_n)}{f(y_n)}\right)\left(1 + (2 - f[x_n, w_n])\left(\frac{f(z_n)}{f(w_n)}\right)\right)\right.$$

$$\left.\times \left(1 + \left(\frac{f(z_n)}{f(x_n)}\right)^2\right)\left(1 + (1 - f[x_n, w_n])\left(\frac{f(y_n)}{f(w_n)}\right)^2\right)\right\},$$

where its error equation is as follows:

$$e_{n+1} = \frac{(-1 + c_1)^4 c_2 (2c_2^2 - c_1 c_3)(7c_2^4 - 8c_1 c_2^2 c_3 + c_1^2 c_3^2 + c_1^2 c_2 c_4)}{c_1^7} e_n^8 + O(e_n^9). \tag{2.14}$$

Another example from the new class of optimal iterations (2.10)-(2.11) can be

$$y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n - f(x_n),$$

$$z_n = y_n - \frac{f(y_n)f(w_n)}{(f(w_n) - f(y_n))f[x_n, y_n]}, \tag{2.15}$$

$$x_{n+1} = z_n - \frac{f(z_n)f(w_n)}{(f(w_n) - f(y_n))f[x_n, y_n]}\{W_1\},$$

where

$$W_1 = \left(1 + \frac{f(z_n)}{f(y_n)} + \left(\frac{f(z_n)}{f(y_n)}\right)^2\right)$$

$$\times \left(1 + (2 - f[x_n, w_n])\left(\frac{f(z_n)}{f(w_n)}\right)\right) \tag{2.16}$$

$$\times \left(1 + \left(\frac{f(z_n)}{f(x_n)}\right)^2\right)\left(1 + (1 - f[x_n, w_n])\left(\frac{f(y_n)}{f(w_n)}\right)^2\right),$$

by the following error equation

$$e_{n+1} = \frac{(-1 + c_1)^4 c_2^2 (2c_2^2 - c_1 c_3)(3c_2^3 - 4c_1 c_2 c_3 + c_1^2 c_4)}{c_1^7} e_n^8 + O(e_n^9). \tag{2.17}$$

## 3. A Brief Look at the Literature

In this section, we shortly present some of the well-known high-order derivative-free techniques to find the simple zeros of nonlinear equations for the sake of comparison. Kung and Traub in [4] introduced the following without memory two-step iteration

$$y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n + \beta f(x_n),$$

$$x_{n+1} = y_n - \frac{f(y_n)f(w_n)}{(f(w_n) - f(y_n))f[x_n, y_n]}, \tag{3.1}$$

where $\beta \in \mathbb{R} \setminus \{0\}$, and it was in fact the first two steps of our novel class (2.2)-(2.3) in this paper. They moreover gave the following four-point eighth-order iterative scheme

$$y_n = x_n + \beta f(x_n),$$

$$z_n = y_n - \beta \frac{f(x_n)f(y_n)}{f(y_n) - f(x_n)},$$

$$w_n = z_n - \frac{f(x_n)f(y_n)}{f(z_n) - f(x_n)}\left[\frac{1}{f[y_n, x_n]} - \frac{1}{f[z_n, y_n]}\right],$$

$$x_{n+1} = w_n - \frac{f(x_n)f(y_n)f(z_n)}{f(w_n) - f(x_n)} \tag{3.2}$$

$$\times \left[\frac{1}{f(w_n) - f(y_n)}\left\{\frac{1}{f[w_n, z_n]} - \frac{1}{f[z_n, y_n]}\right\}\right.$$

$$\left. - \frac{1}{f(z_n) - f(x_n)}\left\{\frac{1}{f[z_n, y_n]} - \frac{1}{f[y_n, x_n]}\right\}\right],$$

wherein $\beta \in \mathbb{R} \setminus \{0\}$. Soleymani in [6] suggested the following three-step derivative-free seventh-order method

$$y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n + f(x_n),$$

$$z_n = y_n - \frac{f(y_n)}{f[x_n, w_n]}\left\{1 + \frac{2 + f[x_n, w_n]}{1 + f[x_n, w_n]}\frac{f(y_n)}{f(x_n)}\right\}, \tag{3.3}$$

$$x_{n+1} = z_n - \frac{f(z_n)}{f[y_n, z_n]}\left\{1 + \frac{1}{1 + f[x_n, w_n]}\left(\frac{f(y_n)}{f(x_n)}\right)^2\right\}.$$

For further reading, one should consult the papers [7–10] and the references therein.

*Remark 3.1.* In terms of computational point of view, the efficiency index of our classes of derivative-free without memory methods (2.2)-(2.3) (and the class (2.10)-(2.11)) is greater than that of Newton's and Steffensen's, that is, 1.414, 1.565 of the sixth-order derivative-free technique given in [11], 1.587 of (3.1), 1.626 of (3.3), and is equal to 1.682 of the family (3.2).

## 4. Numerical Experiments

We now check the effectiveness of the novel derivative-free classes of iterative methods. In order to do this, we choose (2.8) as the representative from our class (2.2)-(2.3). Note that the derived methods from the class (2.10)-(2.11) can also be used.

We have compared (2.8) with Steffensen's method (1.1), the fourth-order family of Kung and Traub (3.1) with $\beta = 0.01$, the seventh-order technique of Soleymani (3.3), and the optimal eighth-order family of Kung and Traub (3.2) with $\beta = 1$, using the examples listed

Table 1: The examples considered in this study.

| Test functions | Zeros |
|---|---|
| $f_1(x) = (\sin x)^2 + x$ | $\alpha_1 = 0$ |
| $f_2(x) = (1 + x) + \cos\left(\dfrac{\pi x}{2}\right) - \sqrt{1 - x^2}$ | $\alpha_2 \approx -0.728584046444826 \cdots$ |
| $f_3(x) = (\sin x)^2 - x^2 + 1$ | $\alpha_3 \approx 1.404491648215341 \cdots$ |
| $f_4(x) = e^{-x} + \sin(x) - 2$ | $\alpha_4 \approx -1.0541271240912128 \cdots$ |
| $f_5(x) = xe^{-x} - 0.1$ | $\alpha_5 \approx 0.111832559158963 \cdots$ |
| $f_6(x) = \sqrt{x^4 + 8}\sin\left(\dfrac{\pi}{x^2 + 2}\right) + \dfrac{x^3}{x^4 + 1} - \sqrt{6} + \dfrac{8}{17}$ | $\alpha_6 = -2$ |
| $f_7(x) = \sqrt{x^2 + 2x + 5} - 2\sin(x) - x^2 + 3$ | $\alpha_7 \approx 2.331967655883964 \cdots$ |
| $f_8(x) = \sin^{-1}(x^2 - 1) - \dfrac{x}{2} + 1$ | $\alpha_8 \approx 0.594810968398369 \cdots$ |
| $f_9(x) = \left(\sin(x) - \dfrac{\sqrt{2}}{2}\right)(x + 1)$ | $\alpha_9 \approx 0.785398163397448 \cdots$ |
| $f_{10}(x) = x^5 + x^4 + 4x^2 - 15$ | $\alpha_{10} \approx 1.347428098968304 \cdots$ |
| $f_{11}(x) = x - \sin(\cos(x)) + 1$ | $\alpha_{11} \approx -0.1660390510510295 \cdots$ |
| $f_{12}(x) = x^5 - 10\cos(\tan^{-1}(x))$ | $\alpha_{12} \approx 1.419315939421532 \cdots$ |
| $f_{13}(x) = x^5 - 10\cos(\tan^{-1}(x^4)) + 17x$ | $\alpha_{13} \approx 0.580593457809920 \cdots$ |
| $f_{14}(x) = \sin(\cos(\tan^{-1}(2x^{\sin(x)}))) - 2x$ | $\alpha_{14} \approx 0.273402731005321 \cdots$ |
| $f_{15}(x) = x^3 - x^2 - 2x - \cos(x) + 2$ | $\alpha_{15} \approx 0.498542523582153 \cdots$ |
| $f_{16}(x) = \sqrt{x^3} + \sin(x) - 30$ | $\alpha_{16} \approx 9.716501993365200 \cdots$ |
| $f_{17}(x) = \tan^{-1}(x^2 - x)$ | $\alpha_{17} = 1$ |
| $f_{18}(x) = \sin^{-1}(x^2) - 2x$ | $\alpha_{18} = 0$ |
| $f_{19}(x) = \sin(-3x\cos(\sqrt{x}))$ | $\alpha_{19} \approx -0.74886062246711 \cdots$ |
| $f_{20}(x) = x^2 - \sqrt{\cos(x)}$ | $\alpha_{20} \approx 0.89055270972295 \cdots$ |

in Table 1. The results of comparisons are given in Table 2 in terms of the number significant digits for each test function after the specified number of iterations, that is, for example, $0.1e-207$ shows that the absolute value of the given nonlinear function ($f_1$) after nine iterations (corresponding to iteration (1.1)) is zero up to 207 decimal places.

It is important to review the proof of convergence for our proposed classes of methods (or the compared methods in Table 2) before implementing them. Specifically, one should review the assumptions made in the proofs, when the result of the iterations become divergent. For situations, where the method fails to converge, it is because the assumptions made in the proofs are not met. For this reason in the long run, we give a Mathematica 8. code to extract initial approximations for all the zeros of a nonlinear function in an interval.

It can be observed from Table 2, that almost in most cases our contributed method from the class (2.2)-(2.3) is superior in solving nonlinear equations. Generally speaking, iterative methods of the same order and efficiency, and also with the same background, that is, Newton-type or Steffensen-type methods, have similar numerical outputs due to their similar characters. Note also that experimental results show that whatever the value of $\beta \neq 0$ is smaller, then the output results of solving nonlinear equations will be more accurate.

We have completed Table 2 for two different initial guesses. In Table 2, IT and TNE stand for number of iterations, and total number of (functional) evaluations. To have a fair comparison, we used high number of iterations for (1.1). The numerical results for the test functions support the theoretical discussions given in Section 2. We also point out that the procedure of applying the new iterative methods for nonsmooth functions is similar to those recently illustrated in [12]. And for multiple zeros, we must first take into account

Table 2: Results of convergence for different derivative-free methods.

| $f$ | Guess | | (1.1) | (3.1) | (3.3) | (3.2) | (2.8) |
|---|---|---|---|---|---|---|---|
| $f_1$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | 0.3 | $|f|$ | $0.1e-207$ | $0.3e-143$ | $0.1e-126$ | $0.1e-210$ | $0.4e-257$ |
| | 0.2 | $|f|$ | $0.3e-264$ | $0.5e-177$ | $0.7e-161$ | $0.1e-267$ | $0.1e-323$ |
| $f_2$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | $-0.9$ | $|f|$ | $0.3e-29$ | $0.8e-61$ | $0.5e-29$ | $0.8e-60$ | $0.7e-54$ |
| | $-0.4$ | $|f|$ | $0.6e-156$ | $0.1e-216$ | $0.6e-107$ | $0.1e-178$ | $0.1e-341$ |
| $f_3$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | 1.25 | $|f|$ | $0.7e-411$ | $0.2e-199$ | $0.2e-322$ | $0.3e-381$ | $0.2e-373$ |
| | 1.6 | $|f|$ | $0.1e-298$ | $0.1e-203$ | $0.5e-229$ | $0.4e-258$ | $0.3e-378$ |
| $f_4$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | $-1.3$ | $|f|$ | $0.6e-259$ | $0.4e-182$ | $0.2e-211$ | $0.5e-110$ | $0.4e-339$ |
| | $-1$ | $|f|$ | $0.9e-640$ | $0.2e-330$ | $0.6e-483$ | $0.4e-618$ | $0.1e-636$ |
| $f_5$ | | IT | 10 | 5 | 3 | 3 | 3 |
| | | TNE | 20 | 15 | 12 | 12 | 12 |
| | 0 | $|f|$ | $0.1e-714$ | $0.4e-917$ | $0.5e-220$ | $0.2e-375$ | $0.3e-435$ |
| | $-0.7$ | $|f|$ | $0.1e-64$ | $0.4e-220$ | $0.7e-20$ | $0.4e-53$ | $0.1e-95$ |
| $f_6$ | | IT | 8 | 4 | 3 | 3 | 3 |
| | | TNE | 16 | 12 | 12 | 12 | 12 |
| | $-2.2$ | $|f|$ | $0.9e-362$ | $0.2e-327$ | $0.4e-348$ | $0.8e-518$ | $0.3e-449$ |
| | $-1.7$ | $|f|$ | $0.4e-221$ | $0.3e-186$ | $0.2e-119$ | $0.5e-231$ | $0.1e-156$ |
| $f_7$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | 3 | $|f|$ | $0.2e-528$ | $0.2e-232$ | $0.3e-294$ | $0.8e-429$ | $0.3e-411$ |
| | 0.6 | $|f|$ | $0.1e-259$ | $0.1e-232$ | $0.2e-154$ | $0.3e-232$ | $0.5e-362$ |
| $f_8$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | 1.3 | $|f|$ | Div. | $0.1e-172$ | $0.3e-52$ | $0.1e-38$ | $0.7e-300$ |
| | 0.3 | $|f|$ | $0.2e-487$ | $0.6e-275$ | $0.1e-316$ | $0.3e-466$ | $0.5e-555$ |
| $f_9$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | 0.6 | $|f|$ | $0.1e-682$ | $0.4e-277$ | $0.1e-236$ | $0.9e-396$ | $0.4e-483$ |
| | 0.4 | $|f|$ | $0.1e-169$ | $0.5e-168$ | Div. | $0.1e-144$ | $0.2e-271$ |
| $f_{10}$ | | IT | 8 | 4 | 3 | 3 | 3 |
| | | TNE | 16 | 12 | 12 | 12 | 12 |
| | 1.36 | $|f|$ | $0.3e-90$ | $0.4e-442$ | $0.1e-206$ | $0.5e-490$ | $0.8e-873$ |
| | 1.32 | $|f|$ | $0.8e-3$ | $0.5e-353$ | $0.8e-5$ | $0.6e-229$ | $0.5e-696$ |
| $f_{11}$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | $-1$ | $|f|$ | $0.1e-5$ | $0.1e-44$ | $0.2e-34$ | Div. | $0.1e-53$ |
| | 0.3 | $|f|$ | $0.3e-313$ | $0.9e-209$ | $0.1e-192$ | $0.5e-314$ | $0.4e-376$ |
| $f_{12}$ | | IT | 9 | 4 | 3 | 5 | 3 |
| | | TNE | 18 | 12 | 12 | 20 | 12 |
| | 1.4 | $|f|$ | $0.2e-122$ | $0.2e-388$ | $0.3e-133$ | $0.1e-378$ | $0.2e-766$ |
| | 1.3 | $|f|$ | Div. | $0.4e-181$ | Div. | $0.2e-221$ | $0.1e-357$ |

**Table 2:** Continued.

| $f$ | Guess | | (1.1) | (3.1) | (3.3) | (3.2) | (2.8) |
|---|---|---|---|---|---|---|---|
| $f_{13}$ | | IT | 18 | 4 | 3 | 3 | 3 |
| | | TNE | 36 | 12 | 12 | 12 | 12 |
| | 0.3 | $\lvert f \rvert$ | $0.5e-750$ | $0.9e-138$ | $0.2e-63$ | $0.9e-3$ | $0.3e-411$ |
| | 0.8 | $\lvert f \rvert$ | 0.1 | $0.5e-245$ | $0.6e-35$ | $0.1e-194$ | $0.1e-384$ |
| $f_{14}$ | | IT | 8 | 4 | 3 | 3 | 3 |
| | | TNE | 16 | 12 | 12 | 12 | 12 |
| | 0.2 | $\lvert f \rvert$ | $0.2e-42$ | $0.9e-334$ | $0.1e-519$ | $0.1e-214$ | $0.3e-604$ |
| | 0.5 | $\lvert f \rvert$ | $0.2e-283$ | $0.5e-263$ | $0.1e-306$ | $0.8e-364$ | $0.3e-468$ |
| $f_{15}$ | | IT | 8 | 4 | 3 | 3 | 3 |
| | | TNE | 16 | 12 | 12 | 12 | 12 |
| | 0.3 | $\lvert f \rvert$ | $0.1e-269$ | $0.2e-332$ | $0.1e-386$ | $0.6e-666$ | $0.2e-417$ |
| | 0.4 | $\lvert f \rvert$ | $0.6e-352$ | $0.7e-295$ | $0.8e-446$ | $0.3e-552$ | $0.6e-544$ |
| $f_{16}$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | 9 | $\lvert f \rvert$ | $0.9e-391$ | $0.1e-310$ | $0.1e-246$ | $0.4e-346$ | $0.6e-528$ |
| | 9.2 | $\lvert f \rvert$ | $0.1e-474$ | $0.3e-314$ | $0.5e-281$ | $0.2e-448$ | $0.6e-573$ |
| $f_{17}$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | 0.1 | $\lvert f \rvert$ | Div. | $0.1e-209$ | $0.2e-652$ | $0.9e-850$ | $0.1e-388$ |
| | 0.8 | $\lvert f \rvert$ | $0.3e-166$ | $0.3e-115$ | $0.3e-36$ | $0.6e-106$ | $0.2e-200$ |
| $f_{18}$ | | IT | 9 | 4 | 3 | 3 | 3 |
| | | TNE | 18 | 12 | 12 | 12 | 12 |
| | 0.3 | $\lvert f \rvert$ | $0.2e-457$ | $0.1e-162$ | $0.6e-406$ | $0.3e-377$ | $0.2e-294$ |
| | 0.2 | $\lvert f \rvert$ | $0.1e-534$ | $0.4e-216$ | $0.7e-490$ | $0.2e-458$ | $0.4e-401$ |
| $f_{19}$ | | IT | 10 | 4 | 3 | 3 | 3 |
| | | TNE | 20 | 12 | 12 | 12 | 12 |
| | $-0.7$ | $\lvert f \rvert$ | $0.4e-294$ | $0.2e-282$ | $0.2e-56$ | $0.1e-202$ | $0.1e-504$ |
| | $-0.9$ | $\lvert f \rvert$ | Div. | $0.2e-116$ | Div. | Div. | $0.3e-182$ |
| $f_{20}$ | | IT | 8 | 4 | 3 | 3 | 3 |
| | | TNE | 16 | 12 | 12 | 12 | 12 |
| | 0.9 | $\lvert f \rvert$ | $0.5e-445$ | $0.5e-552$ | $0.1e-588$ | $0.2e-937$ | $0.4e-1074$ |
| | 0.7 | $\lvert f \rvert$ | $0.1e-89$ | $0.5e-202$ | $0.4e-72$ | $0.2e-203$ | $0.1e-375$ |

a transformation to convert the multiple zero into a simple one and then employ our iterative methods on the transformed equation. Moreover, for applying on functions with inexact coefficients, one remedy is to rationalize the coefficients first and then apply the desired scheme for obtaining the solutions with high accuracy.

An important aspect of implementing high-order nonlinear solvers is in finding very robust initial guesses to start the process, when high precision computing is needed. As discussed in Section 1, the convergence of our iterative methods is locally. To resolve this shortcoming, the best way is to rely on hybrid algorithms, in which the first item produces a robust initial point and the second item employs the new iterative methods when high precision is required. There are some ways in the literature to find robust starting points, mostly based on interval mathematics see for example, [13]. But herein we take into consideration the programming package Mathematica 8 [14] which could be efficiently

```
f[x_]:=Sin[x]Cot[x]+Cos[x^2]+0.5;
a=-2.; b=12.;
zeros=Reap[soln=y[x]/.First[NDSolve[{y'[x]==Evaluate[D[f[x],x]],y[b]==
(f[b])},y[x],{x,a,b},
         Method->{"EventLocator","Event"->y[x],
"EventAction":>Sow[{x,y[x]}]}]]]][[2,1]];
initialPoints=Sort[Flatten[Take[zeros,Length[zeros],1]]]
```

**Algorithm** 1
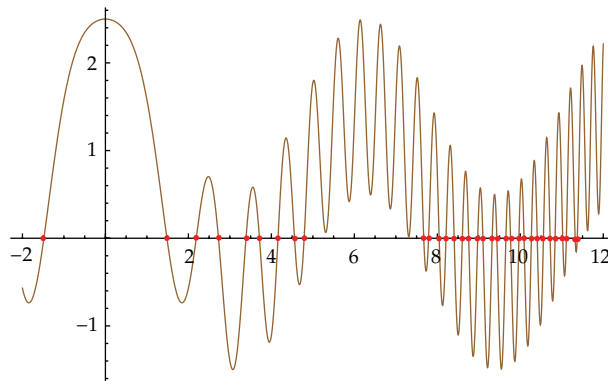


**Figure 1:** The graph of the function $f$ with finitely many zeros in an interval.

applied on lists for high precision computing. In fact using [15], we could build a list of initial guesses enough close with good accuracy to start the procedure of our optimal derivative-free eighth-order methods. The procedure of finding such a robust list is based on the powerful command of NDSolve for the nonlinear function on the interval $D = [a, b]$. Such a way can be written in the following piece of Mathematica code by considering an oscillatory function as the input test function on the domain $D = [-2., 12.]$, see Algorithm 1.

Thus now, we have an efficient list of initial approximations for the zeros of a nonlinear once differentiable function with finitely many zeros in an interval. The number of zeros and the graph of the function including the positions of the zeros can be given by the following commands (see Figure 1), see Algorithm 2.

For this test, there are 33 zeros in the considered interval which can easily be used as the starting points for our proposed high-order derivative-free methods. Note that the output of the vector "initialPoints" contains the initial approximations. In this test problem, the list of zeros are $\{-1.48293, 1.48293, 2.18909, 2.72517, 3.38886, 3.71339, 4.15977, 4.55782, 4.7902, 7.67254, 7.78846, 8.04476, 8.20923, 8.40603, 8.60286, 8.75671, 8.9734, 9.09881, 9.32263, 9.43461, 9.65251, 9.76525, 9.9662, 10.0901, 10.2669, 10.4082, 10.557, 10.719, 10.8379, 11.023, 11.11, 11.3224, 11.3721\}$. Note that we end this section by mentioning that for very oscillatory functions, it is better to first divide the interval into some smaller subintervals and then obtain the solutions. The command NDSolve uses Maximum number of 10000 steps, if it is needed this could be changed. In cases when NDSolve fails, this algorithm might fail too.

```
Length[initialPoints]
Plot[f[x],{x,a,b},  Epilog->{PointSize[Medium], Red, Point[zeros]},
PlotRange->All, PerformanceGoal->"Quality", PlotStyle->{Thick, Brown}]
```

**Algorithm** 2

## 5. Concluding Remarks and Future Works

The importance and application of nonlinear solvers made the construction of new methods by the beginning of the new century. On the other hand, when the cost of derivative evaluation is expensive or in some cases not available, the need for higher-order methods with high efficiency index, which do not require derivative evaluations per full cycle, are more and more felt in the scientific computing.
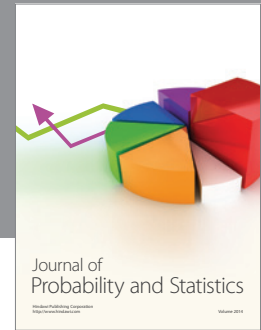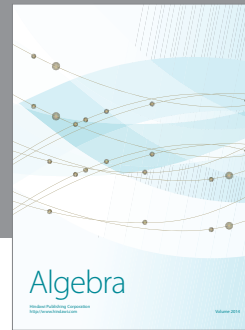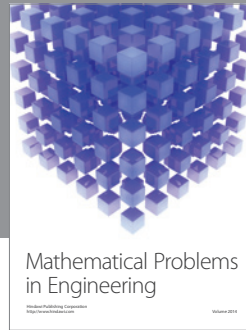
Hence, this paper has recommended two wide classes of optimal eighth-order methods without memory to solve nonlinear scalar equations numerically. The merits of the produced methods from our classes were high efficiency index, being totally free from derivative, high accuracy in numerical examples and also consistency with the conjecture of Kung and Traub. The analytical proof of the main contribution was given in Section 2 and its stability and efficiency were tested in Section 4. The numerical results attested the theoretical aspects as well as the high efficiency index by as small as possible of number of evaluations to achieve the highest possible order. We have also pointed out some notes regarding multiple zeros and how to extract initial approximations for the nonlinear functions with finitely many zeros in an interval.

Providing with memory iterations using the classes (2.2)-(2.3) and (2.10)-(2.11) can be of researchers' interests for future works.

## References

[1] A. Iliev and N. Kyurkchiev, *Nontrivial Methods in Numerical Analysis (Selected Topics in Numerical Analysis)*, Lambert Academy, 2010.

[2] 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.

[3] 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.

[4] 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.

[5] J. F. Steffensen, "Remarks on iteration," *Scandinavian Aktuarietidskr*, vol. 16, pp. 64–72, 1933.

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

[7] M. Sharifi, D. K. R. Babajee, and F. Soleymani, "Finding the solution of nonlinear equations by a class of optimal methods," *Computers & Mathematics with Applications*, vol. 63, no. 4, pp. 764–774, 2012.

[8] 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.

[9] 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.

[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] S. K. Khattri and I. K. Argyros, "Sixth order derivative free family of iterative methods," *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5500–5507, 2011.

[12] A. Cordero, J. L. Hueso, E. Martínez, and J. R. Torregrosa, "A family of derivative-free methods with high order of convergence and its application to nonsmooth equations," *Abstract and Applied Analysis*, vol. 2012, Article ID Article number836901, 15 pages, 2012.

[13] G. Alefeld, "Verified numerical computation for nonlinear equations," *Japan Journal of Industrial and Applied Mathematics*, vol. 26, no. 2-3, pp. 297–315, 2009.

[14] M. Trott, *The Mathematica Guidebook for Numerics*, Springer, New York, NY, USA, 2006.

[15] http://www.mathematica.stackexchange.com/questions/5663/about-multi-root-search-inmathe-matica-for-transcendental-equations?lq=1.