

## *Research Article*

# **A General Three-Step Class of Optimal Iterations for Nonlinear Equations**

**Fazlollah Soleymani,<sup>1</sup> Solat Karimi Vanani,<sup>2</sup> and Abtin Afghani<sup>2</sup>**

<sup>1</sup> *Young Researchers Club, Islamic Azad University, Zahedan Branch, Zahedan 98168, Iran*

<sup>2</sup> *Department of Mathematics, Islamic Azad University, Zahedan Branch, Zahedan 98168, Iran*

Correspondence should be addressed to Fazlollah Soleymani, fazlollah.soleymani@gmail.com

Received 7 August 2011; Accepted 16 September 2011

Academic Editor: Hung Nguyen-Xuan

Copyright © 2011 Fazlollah Soleymani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many of the engineering problems are reduced to solve a nonlinear equation numerically, and as a result, an especial attention to suggest efficient and accurate root solvers is given in literature. Inspired and motivated by the research going on in this area, this paper establishes an efficient general class of root solvers, where per computing step, three evaluations of the function and one evaluation of the first-order derivative are used to achieve the optimal order of convergence eight. The without-memory methods from the developed class possess the optimal efficiency index 1.682. In order to show the applicability and validity of the class, some numerical examples are discussed.

## **1. Introduction**

Numerical solution of nonlinear scalar equations plays a crucial role in many optimization and engineering problems. For example, many engineering systems can be modeled as neutral delay differential equations (NDDEs) that involve a time delay in the derivative of the highest order, which are different from retarded delay differential equations (RDDEs) that do not involve a time delay in the derivative of the highest order. To illustrate more, a system, which consists of a mass mounted on a linear spring to which a pendulum is attached via a hinged massless rod, is used to predict the dynamic response of structures to external forces using a set of actuators, and it is modeled as an NDDE if the delay in actuators is taken into consideration [1]. On the other hand, the stability of a delay differential equation can be investigated on the basis of the root location of the characteristic function. This simple example shows the importance of numerical root solvers in engineering problems.

There are numerical methods, which find one root at a time, such as Newton's iteration or its variant, and the schemes, which find all the roots at a time, namely,

simultaneous methods, such as Weierstrass method. Recently many journals such as Numerical Algorithms, Mathematical Problems in Engineering, Applied Mathematics and Computation, etc., have published new findings; see, for example, [2–5] and the references therein in this active topic of study. To shortly provide some of the newest findings in this field, we mention the following.

Noor et al. in [3] developed the follow-up quartically iterative scheme consisting of three steps and eight numbers of evaluation per full iteration as comes next

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{4f(y_n)}{f'(x_n) + 2f'((x_n + y_n)/2) + f'(y_n)}, \\ x_{n+1} &= z_n - \frac{4f(z_n)}{f'(x_n) + 2f'((x_n + z_n)/2) + f'(z_n)}. \end{aligned} \quad (1.1)$$

In 2010, an eighth-order method is provided in [6] using Ostrowski's method in the first two steps of a three-step cycle as follows:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(x_n)}{f(x_n) - 2f(y_n)} \frac{f(y_n)}{f'(x_n)}, \\ x_{n+1} &= z_n - \left[ 1 + \frac{f(z_n)}{f(x_n)} + \left( \frac{f(z_n)}{f(x_n)} \right)^2 \right] \frac{f[x_n, y_n]f(z_n)}{f[x_n, z_n]f[y_n, z_n]}, \end{aligned} \quad (1.2)$$

wherein  $f[x_0, x_1, \dots, x_k]$  are the divided differences of the function  $f$ .

Soleymani and Mousavi in [7] suggested an iteration without memory scheme including three steps and only four functional evaluations per iteration in what follows:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= x_n + \frac{f(x_n) + f(y_n)}{f'(x_n)} - 2 \frac{f(x_n)}{f'(x_n)} \frac{f(x_n)}{f(x_n) - f(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n) \left\{ \left( 1 + (f(z_n)/f(y_n))^2 \right) \left( 1 + 2(f(z_n)/f(x_n)) \right) \left( 1 - 6(f(y_n)/f(x_n))^3 - \mathcal{A} \right) \times \mathcal{B} \right\}}{f[z_n, y_n] + f[z_n, x_n](z_n - y_n)}, \end{aligned} \quad (1.3)$$

where  $\mathcal{A}$  denotes  $9(f(y_n)/f(x_n))^4$ , and  $\mathcal{B}$  denotes  $(1 + (f(z_n)/f'(x_n))^2)(1 + (f(y_n)/f'(x_n))^3)$ .

For further reading, one may consult [8], where a complete review of the methods given in literature from 2000 to 2010 was furnished, and also [9] for obtaining a background on the application use of such root solvers. We here remark that the efficiency of different

methods could be assessed by the measure of efficiency index, which could be defined as  $\sqrt[p]{p}$ , wherein  $p$  is the order of convergence and  $n$  is the total number of evaluations per iteration. Now, we should remark that Kung and Traub in [10] conjectured that an iterative scheme without memory by using  $n$  evaluations per cycle can arrive at the maximum order of convergence  $2^{n-1}$ . Any without memory iteration, which satisfies this hypothesis, is named as an optimal method in literature.

After providing a short background of this research in this section, we give the main contribution in Section 2. The convergence study of our general three-step class is also furnished therein. We will also produce different optimal three-step iterations from the contributed class. Section 3 discusses some numerical comparisons with the existing methods in literature, and finally Section 4 draws a conclusion of this research paper.

## 2. New Class of Iteration Methods

In order to contribute and give a general class of methods consistent with the optimality conjecture of Kung-Traub, an iteration eighth-order scheme without memory in this section should be constructed such that four evaluations per computing step are used. Such schemes are also known as predictor-corrector methods in which the first step is (Newton's step) predictor, while the other two steps correct the obtained solution. To achieve our goal, we consider the following three-step scheme on which the first two steps are the King's fourth-order family with one free parameter in real numbers,  $\beta \in \mathbb{R}$ ,

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \quad z_n = y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n) + \beta f(y_n)}{f(x_n) + (\beta - 2)f(y_n)}, \quad x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}. \quad (2.1)$$

Clearly in (2.1),  $f'(z_n)$  should be annihilated as the order of convergence remains at the highest level by the smallest use of number of evaluations per iteration. Toward this end, we approximate it by a polynomial of degree two that fits in  $f'(x_n)$ ,  $f(y_n)$ , and  $f(z_n)$ . Therefore, we take into account  $f(t) \approx A(t) = a_0 + a_1(t - y_n) + a_2(t - y_n)^2$  where  $A'(t) = a_1 + 2a_2(t - y_n)$ . Subsequently, by considering  $f'(x_n) = A'(x_n)$ ,  $f(y_n) = A(y_n)$ , and  $f(z_n) = A(z_n)$ , we attain  $a_0 = f(y_n)$ ,

$$\begin{aligned} a_1 + 2a_2(x_n - y_n) &= f'(x_n), \\ a_1 + a_2(z_n - y_n) &= \frac{f(z_n) - f(y_n)}{z_n - y_n} = f[z_n, y_n]. \end{aligned} \quad (2.2)$$

Solving the system of two linear equations with two unknowns, (2.2) gives us  $a_1$  and  $a_2$ . Using the obtained relations for the unknowns in the approximation  $f'(z_n) \approx A'(z_n) = a_1 + 2a_2(z_n - y_n)$  and simplifying, we have

$$f'(z_n) \approx \frac{2f[z_n, y_n](x_n - z_n) + (z_n - y_n)f'(x_n)}{2x_n - z_n - y_n}. \quad (2.3)$$

Considering (2.3) in (2.1) and using weight function approach, we have the following general class of three-step without-memory iteration:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n) + \beta f(y_n)}{f(x_n) + (\beta - 2)f(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)(2x_n - z_n - y_n)}{2f[z_n, y_n](x_n - z_n) + (z_n - y_n)f'(x_n)} \{G(t) + H(\tau) + Q(\gamma)\}, \end{aligned} \quad (2.4)$$

wherein  $G(t)$ ,  $H(\tau)$ , and  $Q(\gamma)$  are three real-valued weight functions with  $t = f(z)/f(y)$ ,  $\tau = f(z)/f(x)$ , and  $\gamma = f(y)/f(x)$  (without the index  $n$ ) which should be chosen such that the order of convergence arrives at the optimal level eight. We summarize this in the following theorem.

**Theorem 2.1.** *Let  $\alpha \in D$  be a simple zero of a sufficiently differentiable function  $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$  for an open interval  $D$ , which contains  $x_0$  as the initial approximation of  $\alpha$ , then the three-step iteration (2.4), which includes four evaluations per full cycle, has the optimal convergence rate eight, when*

$$\begin{aligned} G(0) &= 1, & G'(0) &= 0, & |G''(0)| &< \infty, \\ H(0) &= 0, & H'(0) &= \frac{9}{6}, & |H''(0)| &< \infty, \\ Q(0) &= Q'(0) = Q''(0) = 0, & Q^{(3)}(0) &= -(9 + 18\beta), & |Q^{(4)}(0)| &< \infty, \end{aligned} \quad (2.5)$$

and satisfies the error equation below

$$\begin{aligned} e_{n+1} &= -\frac{1}{24} \left( c_2 \left( -c_3 + c_2^2(1 + 2\beta) \right) \left( 6c_2 \left( 9c_2c_3 - 4c_4 + 4c_2^3(5 + (4 - 3\beta)\beta) \right) \right. \right. \\ &\quad \left. \left. + 12 \left( c_3 - c_2^2(1 + 2\beta) \right)^2 G''(0) + c_2^4 Q^{(4)}(0) \right) \right) e_n^8 + O(e_n^9). \end{aligned} \quad (2.6)$$

*Proof.* By defining  $e_n = x_n - \alpha$  as the error of the iterative scheme in the  $n$ th iterate, applying the Taylor's series expansion for (2.4), and taking into account  $f(\alpha) = 0$ , we have

$$f(x_n) = f'(\alpha) \left[ e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + c_6 e_n^6 + c_7 e_n^7 + c_8 e_n^8 + O(e_n^9) \right], \quad (2.7)$$

where  $c_k = (1/k!)(f^{(k)}(\alpha)/f'(\alpha))$ ,  $k \geq 2$ . Furthermore, we have

$$f'(x_n) = f'(\alpha) \left[ 1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + 6c_6 e_n^5 + 7c_7 e_n^6 + 8c_8 e_n^7 + O(e_n^8) \right]. \quad (2.8)$$

Dividing (2.7) by (2.8) gives us  $f(x_n)/f'(x_n) = e_n - c_2 e_n^2 + 2(c_2^2 - c_3)e_n^3 + (7c_2 c_3 - 4c_2^3 - 3c_4)e_n^4 + \dots + O(e_n^8)$ . Again by substituting this relation in the first step of (2.4) and writing the Taylor's series expansion for  $f(y_n)$ , we obtain, respectively,

$$\begin{aligned} y_n &= \alpha + c_2 e_n^2 + 2(-c_2^2 + c_3)e_n^3 + (-7c_2 c_3 + 4c_2^3 + 3c_4)e_n^4 + \dots + O(e_n^8), \\ f(y_n) &= f'(\alpha) \left[ c_2 e_n^2 + 2(-c_2^2 + c_3)e_n^3 + (-7c_2 c_3 + 4c_2^3 + 3c_4)e_n^4 + \dots + O(e_n^8) \right]. \end{aligned} \quad (2.9)$$

Furthermore, we find

$$z_n - \alpha = (-c_2 c_3 + c_2^3(1 + 2\beta))e_n^4 - 2(c_3^2 + c_2 c_4 - 2c_2^2 c_3(2 + 3\beta) + c_2^4(2 + \beta(6 + \beta)))e_n^5 + \dots + O(e_n^8). \quad (2.10)$$

Similarly, we have

$$\begin{aligned} & \frac{f(z_n)(2x_n - z_n - y_n)}{2f[z_n, y_n](x_n - z_n) + (z_n - y_n)f'(x_n)} \\ &= (-c_2 c_3 + c_2^3(1 + 2\beta))e_n^4 - 2(c_3^2 + c_2 c_4 - 2c_2^2 c_3(2 + 3\beta) + c_2^4(2 + \beta(6 + \beta)))e_n^5 \\ & \quad + (-7c_3 c_4 + 6c_2^2 c_4(2 + 3\beta) - 2c_2^3 c_3(15 + 42\beta + 8\beta^2) + 3c_2(-c_5 + c_2^2(6 + 8\beta))) \\ & \quad + 2c_2^5(5 + \beta(22 + \beta(7 + \beta)))e_n^6 + \dots + O(e_n^8), \end{aligned} \quad (2.11)$$

$$\begin{aligned} z_n - \frac{f(z_n)(2x_n - z_n - y_n)}{2f[z_n, y_n](x_n - z_n) + (z_n - y_n)f'(x_n)} \\ &= \alpha - \frac{3}{2}(c_2 c_3(-c_2 c_3 + c_2^3(1 + 2\beta)))e_n^7 + (6c_2 c_3^3 + 5c_2^2 c_3 c_4 + c_2^7(1 + 2\beta)^2 \\ & \quad - \frac{3}{4}c_2^3 c_3^2(23 + 32\beta) - 2c_2^4(c_4 + 2c_4\beta) + \frac{1}{4}c_2^5 c_3(29 + 2\beta(41 + 6\beta)))e_n^8 + O(e_n^9). \end{aligned} \quad (2.12)$$

We moreover by using (2.11) and (2.5) attain that  $(f(z_n)(2x_n - z_n - y_n)/2f[z_n, y_n](x_n - z_n) + (z_n - y_n)f'(x_n))\{G(f(z_n)/f(y_n)) + H(f(z_n)/f(x_n)) + Q(f(y_n)/f(x_n))\} = (-c_2 c_3 + c_2^3(1 + 2\beta))e_n^4 - 2(c_3^2 + c_2 c_4 - 2c_2^2 c_3(2 + 3\beta) + c_2^4(2 + \beta(6 + \beta)))e_n^5 + (-7c_3 c_4 + 6c_2^2 c_4(2 + 3\beta) - 2c_2^3 c_3(15 + 42\beta + 8\beta^2) + 3c_2(-c_5 + c_2^2(6 + 8\beta)) + 2c_2^5(5 + \beta(22 + \beta(7 + \beta))))e_n^6 + \dots + O(e_n^8)$ . Considering this new relation, (2.12) and (2.5) in the last step of (2.4) will end in

$$\begin{aligned} e_{n+1} = x_{n+1} - \alpha &= -\frac{1}{24} \left( c_2(-c_3 + c_2^2(1 + 2\beta)) \right. \\ & \quad \times \left( 6c_2(9c_2 c_3 - 4c_4 + 4c_2^3(5 + (4 - 3\beta)\beta)) \right. \\ & \quad \left. \left. + 12(c_3 - c_2^2(1 + 2\beta))^2 G''(0) + c_2^4 Q^{(4)}(0) \right) \right) e_n^8 + O(e_n^9). \end{aligned} \quad (2.13)$$

**Table 1:** Typical forms of the weight functions satisfying (2.5).

Weight function	Form 1	Form 2	Form 3
$G(t)$	$1 + t^2$	$1 + \mu t^3, \mu \neq 0$	$(1 + \mu t^2)^{1/\mu}, \mu \neq 0$
$H(\tau)$	$\frac{9}{6}\tau$	$\frac{9}{6}\tau + \mu\tau^2$	$\frac{\tau}{(9/6) + \mu\tau}$
$Q(\gamma)$	$-\frac{(9+18\beta)}{6}\gamma^3$	$-\frac{(9+18\beta)}{6}\gamma^3 + \mu\gamma^4$	$-\frac{(9+18\beta)}{6}\gamma^3 + \mu\gamma^4 + \mu\gamma^5$

$t = f(z)/f(y), \tau = f(z)/f(x), \gamma = f(y)/f(x)$  (without the index  $n$ ), and  $\beta \in \mathbb{R}$ .

**Table 2:** Interesting choices of  $\beta$ ,  $G''(0)$ , and  $Q^{(4)}(0)$  in (2.13), which provide very efficient optimal root solvers.

Method	$\beta$	$G''(0)$	$Q^{(4)}(0)$	Error equation
1	$\frac{4}{3}$	$\frac{27}{44}$	-219	$e_{n+1} = \frac{1}{264}c_2(11c_2^2 - 3c_3)(-27c_3^2 + 88c_2c_4)e_n^8 + O(e_n^9)$
2	$\frac{4}{3}$	0	-120	$e_{n+1} = -\frac{1}{12}(c_2^2(11c_2^2 - 3c_3)(9c_2c_3 - 4c_4))e_n^8 + O(e_n^9)$
3	$-\frac{1}{2}$	-1	-54	$e_{n+1} = \frac{1}{4}c_2c_3(9c_2^2c_3 - 2c_3^2 - 4c_2c_4)e_n^8 + O(e_n^9)$
4	$-\frac{1}{2}$	0	0	$e_{n+1} = \frac{1}{4}c_2^2c_3(9c_2(c_2^2 + c_3) - 4c_4)e_n^8 + O(e_n^9)$
5	0	-2	-96	$e_{n+1} = -\frac{1}{4}(c_2(c_2^2 - c_3)(17c_2^2c_3 - 4c_3^2 - 4c_2c_4))e_n^8 + O(e_n^9)$

This concludes the proof. And it shows that our suggested general class of three-step without-memory methods (2.4)-(2.5) possesses the eighth order of convergence.  $\square$

*Remark 2.2.* The class of three-step methods (2.4)-(2.5) requires four evaluations and has the order of convergence eight. Therefore, this class is of optimal order and supports the Kung-Traub conjecture [10]. Hence, the efficiency index of the eighth-order derivative-involved methods from the class is  $\sqrt[4]{8} \approx 1.682$ .

Some efficient methods from the contributed optimal three-step class are given below. Per computing step, these methods are free from second or higher order derivative computations. The new contributed methods are

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
 z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{2f(x_n) - f(y_n)}{2f(x_n) - 5f(y_n)}, \\
 x_{n+1} &= z_n - \frac{f(z_n)(2x_n - z_n - y_n)}{2f[z_n, y_n](x_n - z_n) + (z_n - y_n)f'(x_n)} \left\{ 1 + \left( \frac{f(z_n)}{f(y_n)} \right)^3 + \frac{9}{6} \frac{f(z_n)}{f(x_n)} - \frac{9}{4} \left( \frac{f(y_n)}{f(x_n)} \right)^4 \right\},
 \end{aligned} \tag{2.14}$$

**Table 3:** Test function, their roots, and the starting points.

Test functions	Roots	Guess
$f_1 = \sin(\tan(x) + x) - \frac{1}{2}$	0.2588298273352688443917065956960	0.4
$f_2 = \sin(x) - x + 2$	2.55419595283704303782966617379187	2
$f_3 = x^5 - 2x + 2 \cos(x)$	-1.25644837202179610636347166071080	-1.3
$f_4 = 2 \cos(x) + \sin(x) - x$	1.37318816571803393434603174397956	1.5
$f_5 = 2x^3 - x \cos(x) + \frac{1}{x^2}$	-0.94316510784112785413792698729205	-0.8
$f_6 = x^9 + x^4 - x^3 + x^2 + 14$	-1.41582823891434741754690479611753	-1.4
$f_7 = x^9 - 14 \sin(x)$	-1.33663345183152516454641803558884	-1.31
$f_8 = \sin(x^2 + x - 3) + x$	0.931908190324955905662501231761206	1.5
$f_9 = \cos(x^2 + x - 3) + x^3$	0.671280744928817559650708950129257	1
$f_{10} = \frac{x}{\tan x} + x$	-0.785398163397448309615660845819875	-0.6
$f_{11} = x^4 + 13x - 1 + \tan x$	0.714180223545189655580041854721814	1
$f_{12} = x^{30} - x - 1 + \sin x$	1.004983564135382106859677582107315	1

where  $e_{n+1} = (1/4)c_2^2c_3(9c_2c_3 - 4c_4)e_n^8 + O(e_n^9)$ ,

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
 z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n)}{f(x_n) - 2f(y_n)}, \\
 x_{n+1} &= z_n - \frac{f(z_n)(2x_n - z_n - y_n)}{2f[z_n, y_n](x_n - z_n) + (z_n - y_n)f'(x_n)} \\
 &\quad \times \left\{ 1 + \left( \frac{f(z_n)}{f(y_n)} \right)^3 + \frac{9}{6} \frac{f(z_n)}{f(x_n)} - \frac{9}{6} \left( \frac{f(y_n)}{f(x_n)} \right)^3 - 5 \left( \frac{f(y_n)}{f(x_n)} \right)^4 \right\},
 \end{aligned} \tag{2.15}$$

where  $e_{n+1} = -(1/4)(c_2^2(c_2^2 - c_3)(9c_2c_3 - 4c_4))e_n^8 + O(e_n^9)$  is its error equation. We also here mention some of the typical forms of the weight functions  $G(t)$ ,  $H(\tau)$ , and  $Q(\gamma)$  in iteration (2.4), which satisfy (2.5) for making the order optimal. These forms are listed in Table 1.

Other than the very efficient methods (2.14) and (2.15) of optimal order eight, many more three-step without-memory iterations can be constructed using Table 1, that is, (2.5) in (2.4) and also with different values for the free parameter  $\beta$ . Thus, now, in order to save the space and also giving some of the other such optimal eighth-order methods according to (2.4) and (2.5), we list the interesting ones in Table 2. Note that we consider first that the weight functions satisfy (2.5), and then we try to make new error equations based on the available data in (2.13).

Future researches in this field of study can now be turned to finding optimal sixteenth-order four-step without-memory iterations based on the general class (2.4)-(2.5). Furthermore, producing with-memory iterations according to this class can also be of researcher's interest for future studies.

**Table 4:** Comparison of different methods for finding the simple roots of test functions.

Test functions	Methods	NM	(1.1)	(1.2)	(1.3)	(2.14)	(2.15)
$f_1$	IN	8	4	3	3	3	3
	TNE	16	32	12	12	12	12
	$ f(x) $	$0.2e-273$	$0.4e-271$	$0.5e-415$	$0.7e-369$	$0.2e-448$	$0.1e-429$
$f_2$	IN	9	4	3	3	3	3
	TNE	18	32	12	12	12	12
	$ f(x) $	$0.1e-485$	$0.8e-238$	$0.6e-372$	$0.6e-301$	$0.4e-409$	$0.1e-378$
$f_3$	IN	9	4	3	3	3	3
	TNE	18	32	12	12	12	12
	$ f(x) $	$0.2e-595$	$0.6e-297$	$0.3e-608$	$0.6e-511$	$0.2e-610$	$0.2e-603$
$f_4$	IN	8	4	3	3	3	3
	TNI	16	32	12	12	12	12
	$ f(x) $	$0.5e-393$	$0.2e-393$	$0.3e-683$	$0.2e-645$	$0.3e-729$	$0.2e-717$
$f_5$	IN	8	5	3	3	3	3
	TNE	16	40	12	12	12	12
	$ f(x) $	$0.1e-798$	$0.1e-431$	$0.3e-600$	$0.3e-511$	$0.8e-641$	$0.3e-641$
$f_6$	IN	9	4	3	3	3	3
	TNE	18	32	12	12	12	12
	$ f(x) $	$0.2e-660$	$0.2e-329$	$0.2e-685$	$0.5e-567$	$0.1e-708$	$0.7e-729$
$f_7$	IN	9	4	3	3	3	3
	TNE	18	32	12	12	12	12
	$ f(x) $	$0.1e-538$	$0.8e-269$	$0.1e-562$	$0.3e-444$	$0.1e-580$	$0.1e-572$
$f_8$	IN	8	4	3	3	3	3
	TNE	16	32	12	12	12	12
	$ f(x) $	$0.2e-204$	$0.3e-148$	$0.3e-239$	$0.7e-158$	$0.6e-205$	$0.5e-198$
$f_9$	IN	9	4	3	3	3	3
	TNE	18	32	12	12	12	12
	$ f(x) $	$0.1e-332$	$0.1e-164$	$0.1e-273$	$0.5e-254$	$0.7e-336$	$0.1e-315$
$f_{10}$	IN	8	4	3	3	3	3
	TNE	16	32	12	12	12	12
	$ f(x) $	$0.4e-329$	$0.1e-329$	$0.5e-689$	$0.2e-579$	$0.4e-696$	$0.5e-721$
$f_{11}$	IN	8	4	3	3	3	3
	TNE	16	32	12	12	12	12
	$ f(x) $	$0.1e-309$	$0.6e-285$	$0.2e-291$	$0.7e-312$	$0.3e-219$	$0.2e-255$
$f_{12}$	IN	9	4	3	3	3	3
	TNE	18	32	12	12	12	12
	$ f(x) $	$0.4e-576$	$0.4e-288$	$0.3e-619$	$0.4e-486$	$0.1e-692$	$0.5e-651$

### 3. Computational Examples

The contribution given in Section 2 is supported here through numerical works. We check the effectiveness of the novel methods (2.14) and (2.15) from our class of methods. For this reason, we have compared our new methods with Newton's method (NM), (1.1), (1.2), and (1.3). The nonlinear test functions are furnished in Table 3. The results of comparisons are given in Table 4 in terms of the number significant digits for each test function after some specified iterations.



All computations in this paper were performed in MATLAB 7.6 using variable precision arithmetic (VPA) to increase the number of significant digits. We have considered the following stopping criterion  $|f(x_n)| \leq 10^{-800}$ . In Table 4,  $0.2e - 448$  shows that the absolute value of the given nonlinear function after three iterations is zero up to 448 decimal places. In Table 4, IN and TNE stand for iteration number and total number of evaluation. As shown in Table 4, the proposed method (2.14) is preferable to Newton's method and some methods with fourth- and eighth-order of convergences. It is evident that (2.14) is more robust than the other competent from various orders. We also recall an important concern in using multi point iterations, which indicates that the high-order root solvers are very sensitive for initial guesses far from the root. And they are so powerful for starting points in the vicinity of the sought zero and so close.

*Remark 3.1.* If we need to solve a lot of equations from a large system of boundary-value problems, then the cost of function evaluations becomes important. Therefore, the proposed class (2.4)-(2.5) is valuable for solving such problems.

#### 4. Concluding Remarks

In recent years, numerous works have been focusing on the development of more advanced and efficient methods for nonlinear scalar equations. Many methods have been developed, which improve the convergence rate of the Newton's method. One practical drawback of so many methods is their slow rate of convergence. This paper has developed and established a rapid class of eighth-order iteration methods. Per iteration, the methods from our class require three evaluations of the function and one of its first derivatives; and therefore, the efficiency of the methods is equal to  $\sqrt[4]{8} \approx 1.682$ , which is better than that of the classical Newton's method. Kung and Traub [10] conjectured that a multipoint iteration without memory based on  $n$  evaluations of  $f$  or its derivatives could achieve optimal convergence order  $2^{n-1}$ . Newton's method is an example, which agrees with Kung-Traub's conjecture for  $n = 2$ , and the class of methods (2.4)-(2.5) is another example, which agrees with Kung-Traub's hypothesis for  $n = 4$ . Thus, the suggested class (2.4)-(2.5) is effective and attracts the attention of researchers.

#### References

- [1] Z. H. Wang, "Numerical stability test of neutral delay differential equations," *Mathematical Problems in Engineering*, vol. 2008, Article ID 698043, 10 pages, 2008.
- [2] N. A. Mir, R. Muneer, and I. Jabeen, "Some families of two-step simultaneous methods for determining zeros of nonlinear equations," *ISRN Applied Mathematics*, Article ID 817174, 11 pages, 2011.
- [3] M. A. Noor, K. I. Noor, E. Al-Said, and M. Waseem, "Some new iterative methods for nonlinear equations," *Mathematical Problems in Engineering*, Article ID 198943, 12 pages, 2010.
- [4] P. Sargolzaei and F. Soleymani, "Accurate fourteenth-order methods for solving nonlinear equations," *Numerical Algorithms*, vol. 58, pp. 513–527, 2011.
- [5] 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*. In press.
- [6] J. R. Sharma and R. Sharma, "A new family of modified Ostrowski's methods with accelerated eighth order convergence," *Numerical Algorithms*, vol. 54, no. 4, pp. 445–458, 2010.
- [7] F. Soleymani and B. S. Mousavi, "A novel computational technique for finding simple roots of nonlinear equations," *International Journal of Mathematical Analysis*, vol. 5, pp. 1813–1819, 2011.
- [8] A. Iliev and N. Kyurkchiev, *Nontrivial Methods in Numerical Analysis (Selected Topics in Numerical Analysis)*, Lambeth Academy, 2010.
- [9] G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method*, Kluwer Academic, Boston, Mass, USA, 1994.

- [10] 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.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

