MATHEMATICAL COMMUNICATIONS
Math. Commun. **21**(2016), 189–197

A new optimal family of three-step methods for efficient finding of a simple root of a nonlinear equation

Nebojša Ralević* and Dejan Ćebić

Faculty of Engineering, University of Novi Sad, Trg Dositeja Obradovića 6, 21 000 Novi Sad, Serbia

Received October 16, 2015; accepted February 3, 2016

Abstract. This study presents a new efficient family of eighth-order methods for finding the simple root of a nonlinear equation. The new family consists of three steps: the Newton step, any optimal fourth-order iteration scheme and a simply structured third step which improves the convergence order up to at least eight and ensures the efficiency index 1.6818. For several relevant numerical test functions, the numerical performances confirm the theoretical results.
AMS subject classifications: 65H05, 65B99

Key words: nonlinear equation, iterative methods, eighth-order convergence, optimal methods, divided differences

1. Introduction

One of the most frequent problems in engineering, scientific computing and applied mathematics in general, is the problem of solving a nonlinear equation $f(x) = 0$. In this research, we are interested in finding simple roots of the nonlinear function $f : D \subseteq \mathbb{R} \to \mathbb{R}$, where D is some open interval. The best known iterative method for determining the solution of this problem is Newton's method given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \tag{1}$$

which produces a sequence $\{x_n\}$ quadratically convergent to the simple root α, if the initial approximation x_0 is sufficiently close to α.

There are many studies which have been developed with the aim to create multi-step iterative methods with the improved convergence order. Some two-step methods with third or fourth order of convergence are considered in [2, 5, 6, 8, 12, 14, 20], and some three-step methods with sixth, seventh and eighth convergence order are given in [1, 4, 7, 9, 11, 15, 17, 18, 19]. Higher convergence order is achieved by the higher cost in the sense of the additional function or derivative evaluations. The coefficient $p^{1/m}$ is introduced by Ostrowski [13], where p is the convergence order and m is the number of function or derivative evaluations per iteration, as a measure of methods efficiency (*the efficiency index*). According to the Kung-Traub conjecture [10],

*Corresponding author. *Email addresses:* `nralevic@uns.ac.rs` (N. Ralević), `cebicd@gmail.com` (D. Ćebić)

if a multipoint iterative method without memory requires one first-order derivative evaluation and $n-1$ function evaluations per iteration, it can reach the convergence order of at most 2^{n-1}. In the literature, those methods are known as optimal methods. The survey and certain generalizations of optimal methods can be found in [16].

This paper is reduced only to methods with optimal properties, especially to the eighth-order methods. Recently, Sharma and Arora [17] have proposed an efficient family of three-step methods based on the following basic requirements: (i) high convergence speed, (ii) minimum computational cost, and (iii) a simple structure. The iteration scheme is given by

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = M_4(x_n, w_n), \\ x_{n+1} = z_n - \dfrac{f(z_n)}{f'(x_n)} \dfrac{f'(x_n) - f[w_n, x_n] + f[z_n, w_n]}{2f[z_n, w_n] - f[z_n, x_n]} \end{cases}, \tag{2}$$

where the first step is Newton's method, $M_4(\cdot,\cdot)$ is any optimal fourth-order iterative scheme and $f[\cdot,\cdot]$ represents the first order divided difference. For every optimal scheme $M_4(\cdot,\cdot)$, method (2) reaches the eighth convergence order, requires one derivative and three function evaluations, and therefore, its efficiency index is $8^{1/4} \approx 1.6818$.

In Section 2, we suggest a new optimal iteration scheme satisfying the same requirements as method (2), but with an even more simpler structure. Section 2 also includes the convergence analysis of the new method and the proof of its optimal behavior. In Section 3, we compare its numerical performance with method (2) and other well known eighth-order methods. Concluding remarks are given in Section 4.

2. The new method and covergence analysis

Preserving the first and the second step of the Sharma and Arora's scheme (2), we present a new efficient family of three-step methods for locating a simple root α of nonlinear function $f(x)$. The new iteration scheme has a form

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = M_4(x_n, w_n), \\ x_{n+1} = z_n + \dfrac{f(z_n)}{f[z_n, x_n]} \dfrac{f[z_n, w_n]}{f[z_n, x_n] - 2f[z_n, w_n]}, \end{cases} \tag{3}$$

where $M_4(\cdot,\cdot)$ is any optimal fourth-order method based on the Newton step, which satisfies

$$z_n - \alpha = A_4 e_n^4 + A_5 e_n^5 + A_6 e_n^6 + A_7 e_n^7 + A_8 e_n^8 + O(e_n^9), \tag{4}$$

where $e_n = x_n - \alpha$. The convergence order of method (3) remains eight, and consequently, the efficiency index is $8^{1/4} \approx 1.6818$ as well. It is easy to note that the third step of the new method requires only two different divided differences while method (2) requires three. The theoretical properties and the conditions for optimal behavior of the new method are summarized in the next theorem.

Theorem 1. *Let α be a simple root of sufficiently differentable function $f(x)$ and $M_4(\cdot,\cdot)$ is any optimal fourth-order iteration scheme satisfying (4). Then for any initial approximation x_0 chosen close enough to α, method (3) is at least of eighth order.*

Proof. Let $c_i = (1/i!)f^{(i)}(\alpha)/f'(\alpha)$ for $i = 2, 3, ...$ From Taylor's expansion of $f(x_n)$ and $f'(x_n)$ about α, it is well known that

$$f(x_n) = f'(\alpha)\cdot e_n\big(1 + c_2e_n + c_3e_n^2 + c_4e_n^3 + ... + c_8e_n^7\big) + O(e_n^9), \tag{5}$$

$$f'(x_n) = f'(\alpha)\cdot\big(1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + ... + 8c_8e_n^7\big) + O(e_n^8), \tag{6}$$

and that the error of Newton's iteration $w_n - \alpha$, denoted by $\hat{e}_n$, can be written as

$$\begin{aligned}\hat{e}_n =\ & c_2e_n^2 + \big(-2c_2^2 + 2c_3\big)e_n^3 + \big(4c_2^3 - 7c_2c_3 + 3c_4\big)e_n^4\\ &+\big(-8c_2^4 + 20c_2^2c_3 - 6c_3^2 - 10c_2c_4 + 4c_5\big)e_n^5\\ &+\big(16c_2^5 - 52c_2^3c_3 + 28c_2^2c_4 - 17c_3c_4 + c_2(33c_3^2 - 13c_5) + 5c_6\big)e_n^6\\ &-2\Big(16c_2^6 - 64c_2^4c_3 - 9c_3^3 + 36c_2^3c_4 + 6c_4^2 + 9c_2^2(7c_3^2 - 2c_5) + 11c_3c_5\\ &+c_2(-46c_3c_4 + 8c_6) - 3c_7\Big)e_n^7\\ &+\Big(64c_2^7 - 304c_2^5c_3 + 176c_2^4c_4 + 75c_3^2c_4 + c_2^3(408c_3^2 - 92c_5) - 31c_4c_5 - 27c_3c_6\\ &+c_2^2(-348c_3c_4 + 44c_6) + c_2(-135c_3^3 + 64c_4^2 + 118c_3c_5 - 19c_7) + 7c_8\Big)e_n^8\\ &+O(e_n^9).\end{aligned} \tag{7}$$

Since equation (4) holds, substituting separately (7) and (4) into (5), we get

$$\begin{aligned}f(w_n) =\ & f'(\alpha)\cdot e_n^2\Big[c_2 - 2\big(c_2^2 - c_3\big)e_n + \big(5c_2^3 - 7c_2c_3 + 3c_4\big)e_n^2\\ &-2\big(6c_2^4 - 12c_2^2c_3 + 3c_3^2 + 5c_2c_4 - 2c_5\big)e_n^3\\ &+\big(28c_2^5 - 73c_2^3c_3 + 34c_2^2c_4 - 17c_3c_4 + c_2(37c_3^2 - 13c_5) + 5c_6\big)e_n^4\\ &-2\Big(32c_2^6 - 103c_2^4c_3 - 9c_3^3 + 52c_2^3c_4 + 6c_4^2 + c_2^2(80c_3^2 - 22c_5)\\ &+11c_3c_5 + c_2(-52c_3c_4 + 8c_6) - 3c_7\Big)e_n^5\\ &+\Big(144c_2^7 - 552c_2^5c_3 + 297c_2^4c_4 + 75c_3^2c_4 + 2c_2^3(291c_3^2 - 67c_5)\\ &-31c_4c_5 - 27c_3c_6 + c_2^2(-455c_3c_4 + 54c_6)\\ &+c_2(-147c_3^3 + 73c_4^2 + 134c_3c_5 - 19c_7) + 7c_8\Big)e_n^6\Big] + O(e_n^9)\end{aligned} \tag{8}$$

and

$$f(z_n) = f'(\alpha) \cdot e_n^4 \left(A_4 + A_5 e_n + A_6 e_n^2 + A_7 e_n^3 + (A_8 + A_4^2 c_2) e_n^4 \right) + O(e_n^9), \qquad (9)$$

respectively, required for calculating divided differences $f[z_n, w_n]$ and $f[z_n, x_n]$. After substituting equations (5), (8) and (9) in the third step of method (3), and simplifying with the help of *Mathematica*'s symbolic computation, we have the error equation

$$e_{n+1} = A_4 \big(c_2 c_4 - c_3^2 \big) e_n^8 + O(e_n^9). \qquad (10)$$

□

For the purpose of comparing the new family of methods with other recently developed methods, we choose the optimal fourth-order iteration schemes for the second step of (3), as they suggested in Sharma and Arora's research [17]. Namely, method (3) is denoted by NM_1, NM_2 and NM_3, if the second step $z_n = M_4(x_n, z_n)$ has a form

- $z_n = w_n - \dfrac{f(w_n)}{2f[w_n, x_n] - f'(x_n)}$, [13],
- $z_n = w_n - \Big(\dfrac{2}{f[w_n, x_n]} - \dfrac{1}{f'(x_n)} \Big) f(w_n)$, [6],
- $z_n = w_n - \Big(3 - \dfrac{2f[w_n, x_n]}{f'(x_n)} \Big) \dfrac{f(w_n)}{f'(x_n)}$, [17].

Analogously, when those fourth-order schemes are used for constructing the members of Sharma and Arora's family (2), they are denoted by SA_1, SA_2 and SA_3, respectively.

3. Numerical results

First, we list other relevant eighth-order methods that will be numerically compared with the members of families (2) and (3) described in the previous section:

Bi, Wu and Ren's method [1] (denoted by BWR):

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = w_n - \dfrac{2f(x_n) - f(w_n)}{2f(x_n) - 5f(w_n)} \dfrac{f(w_n)}{f'(x_n)}, \\ x_{n+1} = z_n - \dfrac{f'(x_n) + (\beta + 2) f(z_n)}{f(x_n) + \beta f(z_n)} \dfrac{f(z_n)}{f[z_n, w_n] + f[z_n, x_n, x_n](z_n - w_n)}, \quad \beta \in \mathbb{R}, \end{cases}$$

where $f[z_n, x_n, x_n] = \dfrac{f[z_n, x_n] - f'(x_n)}{z_n - x_n}$.

Thukral and Petković's method [19] (TP):

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = w_n - \dfrac{f(x_n)+\beta_1 f(w_n)}{f(x_n)+(\beta_1-2)f(w_n)}\dfrac{f(w_n)}{f'(x_n)}, \quad \beta_1 \in \mathbb{R}, \\ x_{n+1} = z_n - \Big(\phi(t) + \dfrac{f(z_n)}{f(w_n)-\beta_2 f(z_n)} + \dfrac{4f(z_n)}{f(x_n)}\Big)\dfrac{f(z_n)}{f'(x_n)}, \quad \beta_2 \in \mathbb{R}, \end{cases}$$

where $\phi(t) = 1 + 2t + (5-2\beta_1)t^2 + (12 - 12\beta_1 + 2\beta_1^2)t^3$ and $t = f(w_n)/f(x_n)$.

Liu and Wang's method [11] (LW):

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = w_n - \dfrac{f(x_n)}{f(x_n)-2f(w_n)}\dfrac{f(w_n)}{f'(x_n)}, \quad \beta_1, \beta_2 \in \mathbb{R}. \\ x_{n+1} = z_n - \Big[\Big(\dfrac{f(x_n)-f(w_n)}{f(x_n)-2f(w_n)}\Big)^2 + \dfrac{f(z_n)}{f(w_n)-\beta_1 f(z_n)} + \dfrac{4f(z_n)}{f(x_n)+\beta_2 f(z_n)}\Big]\dfrac{f(z_n)}{f'(x_n)}. \end{cases}$$

Cordero, Torregrosa and Vassileva's method [4] (CTV):

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \dfrac{f(x_n)-f(w_n)}{f(x_n)-2f(w_n)}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = u_n - \dfrac{3(\beta_2+\beta_3)(u_n-z_n)}{\beta_1(u_n-z_n)+\beta_2(w_n-x_n)+\beta_3(z_n-x_n)}\dfrac{f(z_n)}{f'(x_n)}, \quad \beta_1, \beta_2, \beta_3 \in \mathbb{R}, \end{cases}$$

where $\beta_2 + \beta_3 \neq 0$ and $u_n = z_n - \dfrac{f(z_n)}{f'(x_n)}\big(\dfrac{f(x_n)-f(w_n)}{f(x_n)-2f(w_n)} + \dfrac{1}{2}\dfrac{f(z_n)}{f(w_n)-2f(z_n)}\big)^2$.

Khan, Fardi and Sayevand's method [7] (KFS):

$$\begin{cases} w_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = w_n - \dfrac{f^2(x_n)}{f^2(x_n)-2f(x_n)f(w_n)+\beta_1 f^2(w_n)}\dfrac{f(w_n)}{f'(x_n)}, \quad \beta_1 \in \mathbb{R}, \\ x_{n+1} = z_n - \dfrac{1}{1+\beta_2 q_n^2}\dfrac{f(z_n)}{K - C(w_n-z_n) - D(w_n-z_n)^2}, \quad \beta_2 \in \mathbb{R}, \end{cases}$$

where $q_n = \dfrac{f(z_n)}{f(x_n)}$, $D = \dfrac{f'(x_n)-H}{(x_n-w_n)(x_n-z_n)} - \dfrac{H-K}{(x_n-z_n)^2}$, $C = \dfrac{H-K}{x_n-z_n} - D(x_n + w_n - 2z_n)$, $H = \dfrac{f(x_n)-f(w_n)}{x_n-w_n}$ and $K = \dfrac{f(w_n)-f(z_n)}{w_n-z_n}$.

The values of real parameters used for numerical calculations are suggested by authors of the original papers ($\beta = 1$ for BWR; $\beta_1 = 0, \beta_2 = 0$ for TP; $\beta_1 = 5, \beta_2 = -7$ for LW; $\beta_1 = 0, \beta_2 = 1, \beta_3 = 0$ for CTV; $\beta_1 = 1, \beta_2 = 1$ for KFS).

All numerical computations were carried out by *Mathematica* software package using its *SetPrecision* function with 10000 significant digits, on the computer with the Windows Vista 32-bit operating system and the Intel(R) Pentium(R) Dual CPU @ 1.73 GHz processor.

Numerical properties of the methods are checked through several test examples taken from [3, 4, 16]. They are listed with the corresponding roots as follows:

$$f_1(x) = x^5 + x^4 + 4x^2 - 15; \qquad \alpha \approx 1.347428099$$
$$f_2(x) = x^3 + 4x^2 - 15; \qquad \alpha \approx 1.631980806$$
$$f_3(x) = e^{-x^2+x+2} - 1; \qquad \alpha = -1$$
$$f_4(x) = (x-2)(x^{10} + x + 1)e^{-x-1}; \qquad \alpha = 2$$
$$f_5(x) = \log x + \sqrt{x} - 5; \qquad \alpha \approx 8.309432694$$
$$f_6(x) = \sin x - x/2; \qquad \alpha \approx 1.895494267$$

Tables 1-6 show the numerical performances of the methods. The number of iterations (it) required to satisfy the stopping criterion $|x_{n+1} - x_n| + |f(x_n)| < 10^{-200}$ is displayed in the second column. The errors $|x_{k+1} - x_k|$ for $k = 1, 2, 3$ are given in the third, fourth and fifth columns. The order of convergence (COC), calculated using the last three iterations by the formula COC $= \frac{\log|f(x_k)/f(x_{k-1})|}{\log|f(x_{k-1})/f(x_{k-2})|}$ is displayed in the sixth column with the aim to verify the theoretically derived order of convergence. The last column shows CPU time considered as the average of 50 performances of each method.

Due to the fact that every COC value is approximately 8, it is clear that the eighth convergence order of method (3) and the underlying theory are numerically confirmed. Comparison with other optimal methods also verifies the relatively good numerical performance.

method	it	$\|x_2 - x_1\|$	$\|x_3 - x_2\|$	$\|x_4 - x_3\|$	COC	CPU
BWR	5	0.06793	$1.072 \cdot 10^{-9}$	$5.928 \cdot 10^{-72}$	8.0000	0.2134
TP	5	0.07501	$1.227 \cdot 10^{-8}$	$1.133 \cdot 10^{-62}$	8.0000	0.2789
LW	5	0.004868	$8.790 \cdot 10^{-19}$	$9.730 \cdot 10^{-145}$	8.0000	0.2502
CTV	5	0.03221	$7.815 \cdot 10^{-13}$	$1.040 \cdot 10^{-97}$	8.0000	0.1672
KFS	5	0.05476	$2.248 \cdot 10^{-10}$	$2.306 \cdot 10^{-77}$	8.0000	0.2602
SA_1	5	0.009520	$2.696 \cdot 10^{-17}$	$1.080 \cdot 10^{-133}$	8.0000	0.1716
SA_2	5	0.01217	$5.108 \cdot 10^{-15}$	$4.452 \cdot 10^{-114}$	8.0000	0.1797
SA_3	5	0.01331	$3.509 \cdot 10^{-14}$	$7.133 \cdot 10^{-107}$	8.0000	0.1870
NM_1	5	0.003659	$3.088 \cdot 10^{-21}$	$7.892 \cdot 10^{-166}$	8.0000	0.1703
NM_2	5	0.004992	$2.007 \cdot 10^{-19}$	$1.402 \cdot 10^{-150}$	8.0000	0.1791
NM_3	5	0.01002	$9.275 \cdot 10^{-17}$	$5.305 \cdot 10^{-129}$	8.0000	0.1872

Table 1: *Numerical results for function* $f_1(x)$, $x_0 = 2.4$

method	it	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	COC	CPU
BWR	4	$1.169 \cdot 10^{-7}$	$7.913 \cdot 10^{-59}$	$3.482 \cdot 10^{-468}$	8.0000	0.1466
TP	4	$4.631 \cdot 10^{-6}$	$1.388 \cdot 10^{-44}$	$9.051 \cdot 10^{-353}$	8.0000	0.1791
LW	4	$1.146 \cdot 10^{-6}$	$3.368 \cdot 10^{-50}$	$1.870 \cdot 10^{-398}$	8.0000	0.1573
CTV	4	$4.604 \cdot 10^{-7}$	$7.137 \cdot 10^{-54}$	$2.378 \cdot 10^{-428}$	8.0000	0.1267
KFS	4	$8.335 \cdot 10^{-7}$	$1.688 \cdot 10^{-51}$	$4.776 \cdot 10^{-409}$	8.0000	0.1878
SA_1	4	$1.666 \cdot 10^{-7}$	$8.463 \cdot 10^{-58}$	$3.749 \cdot 10^{-460}$	8.0000	0.1092
SA_2	4	$1.277 \cdot 10^{-6}$	$1.309 \cdot 10^{-49}$	$1.597 \cdot 10^{-393}$	8.0000	0.1154
SA_3	4	$2.861 \cdot 10^{-6}$	$2.461 \cdot 10^{-46}$	$7.368 \cdot 10^{-367}$	8.0000	0.1250
NA_1	4	$1.807 \cdot 10^{-8}$	$1.424 \cdot 10^{-66}$	$2.122 \cdot 10^{-531}$	8.0000	0.1098
NA_2	4	$3.675 \cdot 10^{-8}$	$1.551 \cdot 10^{-63}$	$1.565 \cdot 10^{-506}$	8.0000	0.1167
NA_3	4	$3.732 \cdot 10^{-8}$	$3.035 \cdot 10^{-63}$	$5.804 \cdot 10^{-504}$	8.0000	0.1217

Table 2: *Numerical results for function* $f_2(x)$, $x_0 = 2$

method	it	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	COC	CPU
BWR	4	$1.309 \cdot 10^{-7}$	$3.607 \cdot 10^{-55}$	$1.197 \cdot 10^{-435}$	8.0000	0.5098
TP	4	$3.698 \cdot 10^{-6}$	$2.309 \cdot 10^{-42}$	$5.335 \cdot 10^{-332}$	8.0000	0.6945
LW	4	$8.888 \cdot 10^{-7}$	$3.981 \cdot 10^{-48}$	$6.454 \cdot 10^{-379}$	8.0000	0.6546
CTV	4	$3.304 \cdot 10^{-7}$	$4.044 \cdot 10^{-52}$	$2.038 \cdot 10^{-411}$	8.0000	0.3713
KFS	4	$6.043 \cdot 10^{-7}$	$1.283 \cdot 10^{-49}$	$5.290 \cdot 10^{-391}$	8.0000	0.5129
SA_1	4	$2.003 \cdot 10^{-7}$	$4.387 \cdot 10^{-54}$	$2.095 \cdot 10^{-427}$	8.0000	0.4387
SA_2	4	$1.231 \cdot 10^{-6}$	$1.183 \cdot 10^{-46}$	$8.616 \cdot 10^{-367}$	8.0000	0.4555
SA_3	4	$2.577 \cdot 10^{-6}$	$1.299 \cdot 10^{-43}$	$5.423 \cdot 10^{-342}$	8.0000	0.4680
NA_1	4	$7.661 \cdot 10^{-8}$	$5.877 \cdot 10^{-58}$	$7.045 \cdot 10^{-459}$	8.0000	0.4324
NA_2	4	$2.045 \cdot 10^{-7}$	$6.310 \cdot 10^{-54}$	$5.183 \cdot 10^{-426}$	8.0000	0.4449
NA_3	4	$2.873 \cdot 10^{-7}$	$1.683 \cdot 10^{-52}$	$2.337 \cdot 10^{-414}$	8.0000	0.4561

Table 3: *Numerical results for function* $f_3(x)$, $x_0 = -0.85$

method	it	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	COC	CPU
BWR	5	0.0003841	$1.191 \cdot 10^{-23}$	$1.031 \cdot 10^{-179}$	8.0000	1.287
TP	5	0.004192	$2.623 \cdot 10^{-14}$	$7.217 \cdot 10^{-104}$	8.0000	1.755
LW	5	0.001381	$6.188 \cdot 10^{-19}$	$1.042 \cdot 10^{-141}$	8.0000	1.670
CTV	5	0.001309	$1.207 \cdot 10^{-19}$	$6.501 \cdot 10^{-148}$	8.0000	0.8487
KFS	5	0.002338	$3.569 \cdot 10^{-17}$	$1.111 \cdot 10^{-127}$	8.0000	1.121
SA_1	5	0.0003173	$6.294 \cdot 10^{-25}$	$1.499 \cdot 10^{-190}$	8.0000	0.9860
SA_2	5	0.0008464	$2.980 \cdot 10^{-20}$	$6.812 \cdot 10^{-152}$	8.0000	1.024
SA_3	5	0.001185	$1.403 \cdot 10^{-18}$	$5.120 \cdot 10^{-138}$	8.0000	1.045
NA_1	4	0.00005326	$5.001 \cdot 10^{-32}$	$3.020 \cdot 10^{-248}$	8.0000	0.8324
NA_2	4	0.0001893	$5.667 \cdot 10^{-27}$	$3.669 \cdot 10^{-207}$	7.9998	0.8655
NA_3	5	0.0003838	$2.853 \cdot 10^{-24}$	$2.692 \cdot 10^{-185}$	8.0000	1.016

Table 4: *Numerical results for function* $f_4(x)$, $x_0 = 2.2$

method	it	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	COC	CPU
BWR	4	$1.442 \cdot 10^{-11}$	$2.336 \cdot 10^{-96}$	$1.109 \cdot 10^{-774}$	8.0000	0.7706
TP	4	$4.848 \cdot 10^{-11}$	$1.142 \cdot 10^{-91}$	$1.079 \cdot 10^{-736}$	8.0000	0.8149
LW	4	$1.780 \cdot 10^{-11}$	$1.473 \cdot 10^{-95}$	$3.245 \cdot 10^{-768}$	8.0000	0.7931
CTV	4	$3.413 \cdot 10^{-12}$	$5.284 \cdot 10^{-102}$	$1.744 \cdot 10^{-820}$	8.0000	0.6146
KFS	4	$3.292 \cdot 10^{-12}$	$3.858 \cdot 10^{-102}$	$1.373 \cdot 10^{-821}$	8.0000	0.6708
SA_1	4	$2.520 \cdot 10^{-12}$	$3.396 \cdot 10^{-103}$	$3.694 \cdot 10^{-830}$	8.0000	0.5984
SA_2	4	$3.429 \cdot 10^{-12}$	$4.809 \cdot 10^{-102}$	$7.206 \cdot 10^{-821}$	8.0000	0.6047
SA_3	4	$3.158 \cdot 10^{-11}$	$2.247 \cdot 10^{-93}$	$1.474 \cdot 10^{-750}$	8.0000	0.6240
NA_1	4	$1.081 \cdot 10^{-12}$	$1.679 \cdot 10^{-106}$	$5.673 \cdot 10^{-857}$	8.0000	0.6009
NA_2	4	$2.120 \cdot 10^{-12}$	$6.897 \cdot 10^{-104}$	$8.665 \cdot 10^{-836}$	8.0000	0.6040
NA_3	4	$5.468 \cdot 10^{-12}$	$3.426 \cdot 10^{-100}$	$8.130 \cdot 10^{-806}$	8.0000	0.6103

Table 5: *Numerical results for function* $f_5(x)$, $x_0 = 8.9$

method	it	$\lvert x_2 - x_1\rvert$	$\lvert x_3 - x_2\rvert$	$\lvert x_4 - x_3\rvert$	COC	CPU
BWR	4	$7.514 \cdot 10^{-18}$	$1.224 \cdot 10^{-139}$	$6.081 \cdot 10^{-1114}$	8.0000	3.417
TP	4	$1.849 \cdot 10^{-19}$	$1.545 \cdot 10^{-150}$	$3.670 \cdot 10^{-1199}$	8.0000	3.369
LW	4	$3.655 \cdot 10^{-20}$	$7.069 \cdot 10^{-157}$	$1.385 \cdot 10^{-1250}$	8.0000	3.370
CTV	4	$7.514 \cdot 10^{-18}$	$7.743 \cdot 10^{-139}$	$9.838 \cdot 10^{-1107}$	8.0000	2.371
KFS	4	$1.527 \cdot 10^{-20}$	$2.733 \cdot 10^{-160}$	$2.880 \cdot 10^{-1278}$	8.0000	2.418
SA_1	4	$6.350 \cdot 10^{-21}$	$1.014 \cdot 10^{-163}$	$4.280 \cdot 10^{-1306}$	8.0000	2.356
SA_2	4	$3.942 \cdot 10^{-20}$	$1.406 \cdot 10^{-156}$	$3.680 \cdot 10^{-1248}$	8.0000	2.355
SA_3	4	$1.002 \cdot 10^{-19}$	$6.273 \cdot 10^{-153}$	$1.482 \cdot 10^{-1218}$	8.0000	2.371
NA_1	4	$1.241 \cdot 10^{-21}$	$4.186 \cdot 10^{-170}$	$6.997 \cdot 10^{-1358}$	8.0000	2.356
NA_2	4	$3.300 \cdot 10^{-21}$	$2.792 \cdot 10^{-166}$	$7.332 \cdot 10^{-1327}$	8.0000	2.356
NA_3	4	$5.347 \cdot 10^{-21}$	$2.159 \cdot 10^{-164}$	$1.525 \cdot 10^{-1311}$	8.0000	2.371

Table 6: *Numerical results for function* $f_6(x)$, $x_0 = 1.9$

For some test functions (for instance, see f_2, f_5 and f_6), it can be seen that all the members of the new family converge faster to the root α than corresponding members of family (2). CPU time for those families does not have significantly different values for every considered test example. Numerical examples suggest that the new family is very competitive with the existing optimal methods.

4. Conclusion

In this paper, we have proposed a new three-step iterative scheme for solving nonlinear equations. If the first two steps are any optimal fourth-order methods based on Newton's iteration (1), then the third step provides the eighth-order of convergence and preserves the optimal properties of the new method with the efficiency index $8^{1/4} \approx 1.6818$. In addition to the high efficiency, the new method has a simpler structure than some recently developed methods, which is also one of the basic requirements for producing a numerical algorithm. Several test examples confirm the theoretical results and good numerical properties.

References

[1] W. Bi, H. Ren, Q. Wu, *Three-step iterative methods with eighth-order convergence for solving nonlinear equations*, J. Comput. Appl. Math. **225**(2009), 105–112.

[2] C. Chun, M. Y. Lee, B. Neta, J. Džunić, *On optimal fourth-order iterative methods free from second derivative and their dynamics*, Appl. Math. Comput. **218**(2012), 6427–6438.

[3] C. Chun, B. Neta, *An analysis of a new family of eighth-order optimal methods*, Appl. Math. Comput. **245**(2014), 86–107.

[4] A. Cordero, J. R. Torregrosa, M. P. Vassileva, *Three-step iterative methods with optimal eighth-order convergence*, J. Comput. Appl. Math. **235**(2011), 3189–3194.

[5] L. Fang, G. He, Z. Hu, *A cubically convergent Newton-type method under weak conditions*, J. Comput. Appl. Math. **220**(2008), 409–412.

[6] M. Grau, J. L. Diaz-Barrero, *A technique to composite a modified Newton's method for solving nonlinear equations*, Ann. Univ. Buchar. Math. Ser. **2**(**LX**)(2011), 53–61.

[7] Y. Khan, M. Fardi, K. Sayevand, *A new general eighth-order family of iterative methods for solving nonlinear equations*, Appl. Math. Lett. **25**(2012), 2262–2266.

[8] R. F. King, *A family of fourth order methods for nonlinear equations*, SIAM J. Numer. Math. **10**(1973), 876–879.

[9] J. Kou, Y. Li, X. Wang, *Some variants of Ostrowski's method with seventh-order convergence*, J. Comput. Appl. Math. **209**(2007), 153–159.

[10] H. T. Kung, J. F. Traub, *Optimal order of one-point and multipoint iteration*, J. Assoc. Comput. Mach. **21**(1974), 643–651.

[11] L. Liu, X. Wang, *Eighth-order methods with high efficiency index for solving nonlinear equations*, Appl. Math. Comput. **215**(2010), 3449–3454.

[12] T. Lukić, N. M. Ralević, *Geometric mean Newton's method for simple and multiple roots*, Appl. Math. Lett. **21**(2008), 30–36.

[13] A. M. Ostrowski, *Solutions of Equations and System of Equations*, Academic Press, New York-London, 1960.

[14] A. Y. Özban, *Some new variants of Newton's method*, Appl. Math. Lett. **17**(2004), 677–682.

[15] S. K. Parhi, D. K. Gupta, *A sixth order method for nonlinear equations*, Appl. Math. Comput. **203**(2008), 50–55.

[16] M. S. Petković, B. Neta, Lj. D. Petković, J. Džunić, *Multipoint Methods for Solving Nonlinear Equations*, Elsevier, Boston, 2013.

[17] J. R. Sharma, H. Arora, *An efficient family of weighted-Newton methods with optimal eighth order convergence*, Appl. Math. Lett. **29**(2014), 1–6.

[18] J. R. Sharma, R. K. Guha, *A family of modified Ostrowski methods with accelerated sixth order convergence*, Appl. Math. Comput. **190**(2007), 111–115.

[19] R. Thukral, M. S. Petković, *A family of three-point methods of optimal order for solving nonlinear equations*, J. Comput. Appl. Math. **233**(2010), 2278–2284.

[20] S. Weerakoon, T. G. I. Fernando, *A variant of Newton's method with accelerated third-order convergence*, Appl. Math. Lett. **13**(2000), 87–93.