

## Research Article

# New Families of Third-Order Iterative Methods for Finding Multiple Roots

R. F. Lin,<sup>1</sup> H. M. Ren,<sup>2</sup> Z. Šmarda,<sup>3</sup> Q. B. Wu,<sup>4</sup> Y. Khan,<sup>4</sup> and J. L. Hu<sup>5</sup>

<sup>1</sup> Department of Mathematics, Taizhou University, Linhai, Zhejiang 317000, China

<sup>2</sup> College of Information and Engineering, Hangzhou Polytechnic, Hangzhou, Zhejiang 311402, China

<sup>3</sup> Department of Mathematics, Brno University of Technology, Brno, Czech Republic

<sup>4</sup> Department of Mathematics, Zhejiang University, Hangzhou, Zhejiang 310027, China

<sup>5</sup> Department of Mathematics, Zhejiang Sci-Tech University, Hangzhou, Zhejiang 310018, China

Correspondence should be addressed to Y. Khan; yasirmath@yahoo.com

Received 5 February 2014; Revised 19 May 2014; Accepted 20 May 2014; Published 15 June 2014

Academic Editor: Alicia Cordero

Copyright © 2014 R. F. Lin 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.

Two families of third-order iterative methods for finding multiple roots of nonlinear equations are developed in this paper. Mild conditions are given to assure the cubic convergence of two iteration schemes (I) and (II). The presented families include many third-order methods for finding multiple roots, such as the known Dong's methods and Neta's method. Some new concrete iterative methods are provided. Each member of the two families requires two evaluations of the function and one of its first derivative per iteration. All these methods require the knowledge of the multiplicity. The obtained methods are also compared in their performance with various other iteration methods via numerical examples, and it is observed that these have better performance than the modified Newton method, and demonstrate at least equal performance to iterative methods of the same order.

## 1. Introduction

Finding the roots of nonlinear equations is one of the most important problems in numerical analysis. In this study, we use iterative methods to find a multiple root  $x^*$  of multiplicity  $m$  ( $m > 1$ ); that is,  $f^{(j)}(x^*) = 0$ ,  $j = 0, 1, \dots, m - 1$ , and  $f^{(m)}(x^*) \neq 0$ , of a nonlinear equation  $f(x) = 0$ .

It is known that the modified Newton method for multiple roots is given by

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

which converges quadratically [1].

There exists a cubically convergent method for multiple roots, presented by Hansen and Patrick [2]. Consider

$$x_{n+1} = x_n - (f(x_n)) \times \left( \frac{m+1}{2m} f'(x_n) - \frac{f(x_n)f''(x_n)}{2f'(x_n)^2} \right)^{-1}, \quad (2)$$

which is an extension of the classical Halley method of the third order.

Another cubically convergent method for multiple roots is proposed by Traub [3]. Consider

$$x_{n+1} = x_n - \frac{m(3-m)}{2} \frac{f(x_n)}{f'(x_n)} - \frac{m^2}{2} \frac{f(x_n)^2 f''(x_n)}{f'(x_n)^3}, \quad (3)$$

which is an extension of the well-known Chebyshev method of the third order.

In recent years, a lot of methods for multiple roots have been presented and analyzed, which require the knowledge of the multiplicity  $m$ ; see [4–24] and references therein.

Based on King's fourth-order method (for simple roots) [25], Dong [4] has developed two third-order methods for

multiple roots, requiring two evaluations of the function and one of its first derivative. Consider

$$\begin{aligned}
 y_n &= x_n - \sqrt{m} \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n - m \left(1 - \frac{1}{\sqrt{m}}\right)^{(1-m)} \frac{f(y_n)}{f'(x_n)}, \\
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n - \frac{f(y_n)}{((m-1)/m)^{m-1} f(x_n) - f(y_n) f'(x_n)}.
 \end{aligned}
 \tag{4}$$

Using the same information, Victory Jr. and Neta [5] have developed a third method. Consider

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n - \frac{f(y_n) f(x_n) + Af(y_n)}{f'(x_n) f(x_n) + Bf(y_n)},
 \end{aligned}
 \tag{6}$$

where

$$\begin{aligned}
 A &= \left(\frac{m}{m-1}\right)^{2m} - \left(\frac{m}{m-1}\right)^{m+1}, \\
 B &= -\frac{(m/(m-1))^m (m-2)(m-1) + 1}{(m-1)^2}.
 \end{aligned}
 \tag{7}$$

Neta [9] has developed another third-order method requiring the same information

$$\begin{aligned}
 y_n &= x_n - \alpha \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n - \frac{(\beta - \alpha) f(x_n) + \gamma f(y_n)}{f'(x_n)},
 \end{aligned}
 \tag{8}$$

where

$$\begin{aligned}
 \alpha &= \frac{1}{2} \frac{m(m+3)}{m+1}, \quad \beta = \frac{m^3 + 4m^2 + 9m + 2}{(m+3)^2}, \\
 \gamma &= \frac{2^{m+1}(m^2 - 1)}{(m+3)^2((m-1)/(m+1))^m}.
 \end{aligned}
 \tag{9}$$

Based on Halley's method, Li et al. [15] have proposed a family of third methods using the same information. Consider

$$\begin{aligned}
 y_n &= x_n - \alpha \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= x_n - \frac{m\alpha^2 \mu^m f(x_n)}{(m - \alpha + \alpha^2) \mu^m f(x_n) - (m - \alpha) f(y_n) f'(x_n)},
 \end{aligned}
 \tag{10}$$

where  $\alpha$  is a real parameter and  $\alpha \neq 0, m$ , and  $\mu = (m - \alpha)/m$ .

Note also that, based on Traub's method [2], Homeier [16] has suggested a family of third methods using the same information. Consider

$$\begin{aligned}
 y_n &= x_n - \alpha \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= x_n - \beta \frac{f(x_n)}{f'(x_n)} - \frac{f(y_n)}{\gamma f'(x_n)},
 \end{aligned}
 \tag{11}$$

where  $\alpha \neq 0, m$  is a real parameter,  $\beta = (m/\alpha^2)(\alpha^2 + \alpha - m)$ , and  $\gamma = (1/m)(1 - \alpha/m)^m(\alpha^2/(m - \alpha))$ .

In this paper, we propose two new families of third-order methods for multiple roots; each of the methods requires two-function and one-derivative evaluation per iteration, respectively. The presented methods are obtained by investigating the following two iteration schemes:

$$\text{(I)} \begin{cases} y_n = x_n - \alpha \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \frac{af(x_n) + bf(y_n) f(y_n)}{cf(x_n) + df(y_n) f'(x_n)}, \end{cases}
 \tag{12}$$

$$\text{(II)} \begin{cases} y_n = x_n - \alpha \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \frac{af(x_n) + bf(y_n) f(x_n)}{cf(x_n) + df(y_n) f'(x_n)}, \end{cases}
 \tag{13}$$

where  $\alpha, a, b, c,$  and  $d$  are parameters to be determined. By specially choosing the parameters in (12) and (13), we get two new families of third-order methods, which include methods (4)–(6), (8), (10), and (11). In fact, the mild conditions to assure the cubic convergence of (I)-type iteration (12) or (II)-type iteration (13) are given. Divided differences are adopted successfully in developing our methods, which will be useful in developing more new methods. Finally, we use some numerical examples to compare the presented methods with the modified Newton method and some known third-order methods.

## 2. Preliminaries

We need the definitions of divided differences and their properties.

*Definition 1* (see [26]). The divided differences  $f[a_0, a_1, \dots, a_k]$  on  $k+1$  distinct points  $a_0, a_1, \dots, a_k$  of a function  $f(x)$  are defined by

$$\begin{aligned}
 f[a_0] &= f(a_0), \\
 f[a_0, a_1] &= \frac{f[a_0] - f[a_1]}{a_0 - a_1}, \\
 &\vdots \\
 f[a_0, a_1, \dots, a_k] &= \frac{f[a_0, a_1, \dots, a_{k-1}] - f[a_1, a_2, \dots, a_k]}{a_0 - a_k}.
 \end{aligned}
 \tag{14}$$

If the function  $f$  is sufficiently differentiable, then its divided differences  $f[a_0, a_1, \dots, a_k]$  can be defined if some of the arguments  $a_i$  coincide. For instance, if  $f(x)$  has a derivative of the  $k$ th order at  $a_0$ , then it makes sense to define

$$f \left[ \underbrace{a_0, a_0, \dots, a_0}_{k+1} \right] = \frac{f^{(k)}(a_0)}{k!}. \tag{15}$$

**Lemma 2** (see [26]). *The divided differences  $f[a_0, a_1, \dots, a_k]$  are symmetric functions of their arguments; that is, they are invariant to permutations of the  $a_0, a_1, \dots, a_k$ .*

**Lemma 3** (see [26]). *If the function  $f$  has  $(k + 1)$ st derivative, then, for every argument  $x$ , the following interpolation formula holds:*

$$f(x) = f[a_0] + \sum_{i=1}^k f[a_0, a_1, \dots, a_i] \prod_{j=0}^{i-1} (x - a_j) + f[a_0, a_1, \dots, a_k, x] \prod_{i=0}^k (x - a_i). \tag{16}$$

**Lemma 4.** *If the function  $f$  has a derivative of the  $(m + 1)$ th order, and  $x^*$  is a multiple root of multiplicity  $m$ , then, for every argument  $x$ , the following formulae hold:*

$$f(x) = f \left[ \underbrace{x^*, x^*, \dots, x^*}_m, x \right] (x - x^*)^m, \tag{17}$$

$$f'(x) = f \left[ \underbrace{x^*, x^*, \dots, x^*}_m, x, x \right] (x - x^*)^m + mf \left[ \underbrace{x^*, x^*, \dots, x^*}_m, x \right] (x - x^*)^{m-1}. \tag{18}$$

*Proof.* Applying Lemma 3 to the case of the multiple zero  $x^*$  of multiplicity  $m$  and using (15), we get (17). Differentiating both sides of (17) gives (18).  $\square$

### 3. Development of New Families of Third-Order Methods

We would like to find the five parameters  $\alpha, a, b, c$ , and  $d$  in I-type iteration (12) and II-type iteration (13) so as to maximize its order of convergence to a root  $x^*$  of multiplicity  $m$ , respectively. Let  $e_n, d_n$  be the errors at the  $n$ th step; that is,

$$e_n = x_n - x^*, \quad d_n = y_n - x^*. \tag{19}$$

Define functions  $g(x)$  and  $h(x)$  as follows:

$$g(x) = f \left[ \underbrace{x^*, x^*, \dots, x^*}_m, x \right], \tag{20}$$

$$h(x) = f \left[ \underbrace{x^*, x^*, \dots, x^*}_m, x, x \right].$$

Write

$$g_n = g(x_n), \quad h_n = h(x_n), \quad g_* = g(x^*), \tag{21}$$

$$h_* = h(x^*).$$

In view of (17) and (18), we get the following:

$$f(x_n) = g_n e_n^m, \tag{22}$$

$$f'(x_n) = h_n e_n^m + m g_n e_n^{m-1}, \tag{23}$$

$$f(y_n) = g(y_n) d_n^m. \tag{24}$$

Using the definitions of divided differences, we get the following:

$$g(y_n) = g(y_n) - g(x_n) + g(x_n) = f \left[ \underbrace{x^*, x^*, \dots, x^*}_m, y_n, x_n \right] (d_n - e_n) + g_n = \left( f \left[ \underbrace{x^*, x^*, \dots, x^*}_m, y_n, x_n \right] - h(x_n) + h(x_n) \right) \times (d_n - e_n) + g_n = p_n (d_n - e_n)^2 + h_n (d_n - e_n) + g_n, \tag{25}$$

where

$$p_n = f \left[ \underbrace{x^*, x^*, \dots, x^*}_m, y_n, x_n, x_n \right]. \tag{26}$$

In view of (22), (23), (12), and (13), we get in turn

$$\frac{f(x_n)}{f'(x_n)} = \frac{g_n e_n}{h_n e_n + m g_n}, \tag{27}$$

$$d_n = e_n - \alpha \frac{g_n e_n}{h_n e_n + m g_n} = \frac{h_n e_n + (m - \alpha) g_n}{h_n e_n + m g_n} e_n, \tag{28}$$

$$d_n - e_n = -\frac{\alpha g_n e_n}{h_n e_n + m g_n}. \tag{29}$$

Substituting (29) into (25) yields

$$g(y_n) = p_n \left( -\frac{\alpha g_n e_n}{h_n e_n + m g_n} \right)^2 + h_n \left( -\frac{\alpha g_n e_n}{h_n e_n + m g_n} \right) + g_n = \frac{g_n}{(h_n e_n + m g_n)^2} \left( [\alpha^2 p_n g_n + (1 - \alpha) h_n^2] e_n^2 + (2 - \alpha) m h_n g_n e_n + m^2 g_n^2 \right). \tag{30}$$

Substituting (30) and (28) into (24) leads to

$$f(y_n) = \frac{g_n [h_n e_n + (m - \alpha) g_n]^m}{(h_n e_n + m g_n)^{m+2}} \times \left( [\alpha^2 p_n g_n + (1 - \alpha) h_n^2] e_n^2 + (2 - \alpha) m h_n g_n e_n + m^2 g_n^2 \right) e_n^m. \tag{31}$$

Write

$$A_n = h_n e_n + m g_n,$$

$$B_n = h_n e_n + (m - \alpha) g_n,$$

$$C_n = [\alpha^2 p_n g_n + (1 - \alpha) h_n^2] e_n^2 + (2 - \alpha) m h_n g_n e_n + m^2 g_n^2. \quad (32)$$

Using (23), (24), and (32), we get

$$\begin{aligned} f'(x_n) &= A_n e_n^{m-1}, \\ f(y_n) &= \frac{g_n B_n^m C_n}{A_n^{m+2}} e_n^m. \end{aligned} \quad (33)$$

Then, we can get the error equations as follows:

$$\begin{aligned} e_{n+1} &= \frac{B_n}{A_n} e_n - \frac{a g_n e_n^m + b (g_n B_n^m C_n / A_n^{m+2}) e_n^m}{c g_n e_n^m + d (g_n B_n^m C_n / A_n^{m+2}) e_n^m} \\ &\quad \times \frac{(g_n B_n^m C_n / A_n^{m+2}) e_n^m}{A_n e_n^{m-1}} \\ &= - \left( (a A_n^{m+2} + b B_n^m C_n) g_n B_n^{m-1} C_n \right. \\ &\quad \left. - (c A_n^{m+2} + d B_n^m C_n) A_n^{m+2} \right] B_n e_n \\ &\quad \times \left( (c A_n^{m+2} + d B_n^m C_n) A_n^{m+3} \right)^{-1}, \end{aligned} \quad (34)$$

for I-type iteration (12), and

$$\begin{aligned} e_{n+1} &= \frac{B_n}{A_n} e_n - \frac{a g_n e_n^m + b (g_n B_n^m C_n / A_n^{m+2}) e_n^m}{c g_n e_n^m + d (g_n B_n^m C_n / A_n^{m+2}) e_n^m} \\ &\quad \times \frac{g_n e_n^m}{A_n e_n^{m-1}} \\ &= - \left( (a A_n^{m+2} + b B_n^m C_n) g_n - (c A_n^{m+2} + d B_n^m C_n) B_n \right) e_n \\ &\quad \times \left( (c A_n^{m+2} + d B_n^m C_n) A_n \right)^{-1}, \end{aligned} \quad (35)$$

for II-type iteration (13).

In view of (34) and (35), the order of convergence for I-type or II-type iteration will arrive at three provided that

$$\frac{\Phi_n}{(c A_n^{m+2} + d B_n^m C_n) A_n^{m+3}} = O(e_n^2), \quad (36)$$

or

$$\frac{\Psi_n}{(c A_n^{m+2} + d B_n^m C_n) A_n} = O(e_n^2), \quad (37)$$

holds true, respectively. Here

$$\begin{aligned} \Phi_n &= \left[ (a A_n^{m+2} + b B_n^m C_n) g_n B_n^{m-1} C_n \right. \\ &\quad \left. - (c A_n^{m+2} + d B_n^m C_n) A_n^{m+2} \right] B_n, \end{aligned} \quad (38)$$

$$\Psi_n = (a A_n^{m+2} + b B_n^m C_n) g_n - (c A_n^{m+2} + d B_n^m C_n) B_n.$$

Write

$$\lambda = m - \alpha. \quad (39)$$

In view of (32), we can get, as  $n \rightarrow \infty$ ,

$$\begin{aligned} A_n &\rightarrow m g_* = m f \left[ \underbrace{x^*, x^*, \dots, x^*}_{m+1} \right] = m \frac{f^{(m)}(x^*)}{m!} \neq 0, \\ B_n &\rightarrow \lambda g_*, \\ C_n &\rightarrow (m g_*)^2, \end{aligned} \quad (40)$$

and then

$$\begin{aligned} (c A_n^{m+2} + d B_n^m C_n) A_n^{m+3} &\rightarrow (c m^m + d \lambda^m) m^{m+5} g_*^{2m+5}, \\ &\quad (n \rightarrow \infty), \\ (c A_n^{m+2} + d B_n^m C_n) A_n &\rightarrow (c m^m + d \lambda^m) m^3 g_*^{m+3}, \\ &\quad (n \rightarrow \infty), \end{aligned} \quad (41)$$

which show that, in order to assure the denominators in (36) and (37) are not equal to zero, we demand naturally that

$$c m^m + d \lambda^m \neq 0. \quad (42)$$

It is obvious that, under the condition (42), the error relations (36) and (37) are equivalent to

$$\Phi_n = O(e_n^2), \quad (43)$$

$$\Psi_n = O(e_n^2), \quad (44)$$

respectively.

Next we will find conditions to assure (43) and (44). Note that the factor  $B_n$  of  $\Phi_n$  plays an important role on the order of  $\Phi_n$ . In fact, using the Taylor formula, we get from (32) the following:

$$B_n = \lambda g_* + O(e_n). \quad (45)$$

Then, in the case  $\lambda = 0$ , to assure the relation (43) holds true, the following estimate is needed:

$$\begin{aligned} \Delta_n &= (a A_n^{m+2} + b B_n^m C_n) g_n B_n^{m-1} C_n \\ &\quad - (c A_n^{m+2} + d B_n^m C_n) A_n^{m+2} = O(e_n), \end{aligned} \quad (46)$$

which demands

$$\Delta_n \rightarrow -c (m g_*)^{2m+4} = 0, \quad (n \rightarrow \infty); \quad (47)$$

that is,  $c = 0$ . This is a contradiction to (42). Hence, in the case  $\lambda = 0$ , the relation (43) cannot be satisfied; that is, we cannot choose parameters so that the order of convergence of (I)-type iteration (12) arrives at three.

In what follows, we suppose  $\lambda \neq 0$ . In this case,  $B_n \rightarrow \lambda g_* \neq 0 (n \rightarrow \infty)$ , and then (43) is equivalent to

$$\Delta_n = O(e_n^2). \tag{48}$$

In view of (32), and by a straight computation, we can get

$$\Delta_n = u_n + v_n e_n + O(e_n^2), \tag{49}$$

where

$$u_n = (m^m \lambda^{m-1} a + \lambda^{2m-1} b - m^{2m} c - m^m \lambda^m d) m^4 g_n^{2m+4}, \tag{50}$$

$$\begin{aligned} v_n = & (m^m \lambda^{m-2} [\lambda^2 + 4\lambda + m(m-1)] a \\ & + \lambda^{2m-2} [2\lambda^2 + (4-2m)\lambda + m(2m-1)] b \\ & - (2m+4) m^{2m} c \\ & - m^m \lambda^{m-1} (\lambda^2 + 4\lambda + m^2) d) m^3 g_n^{2m+3} h_n. \end{aligned} \tag{51}$$

In view of (49) and in order to assure the relation (48) holds, we should choose parameters  $\lambda, a, b, c,$  and  $d$  such that

$$\begin{aligned} m^m \lambda^{m-1} a + \lambda^{2m-1} b - m^{2m} c - m^m \lambda^m d = 0, \\ m^m \lambda^{m-2} [\lambda^2 + 4\lambda + m(m-1)] a \\ + \lambda^{2m-2} [2\lambda^2 + (4-2m)\lambda + m(2m-1)] b \\ - (2m+4) m^{2m} c - m^m \lambda^{m-1} (\lambda^2 + 4\lambda + m^2) d = 0. \end{aligned} \tag{52}$$

By a straight computation, we deduce that

$$c = \frac{m^{m+1} \lambda^{m-1} a + [m - (\lambda - m)^2] \lambda^{2m-1} b}{(\lambda - m)^2 m^{2m}}, \tag{53}$$

$$d = \frac{[(\lambda - m)^2 - m] m^m a + [2(\lambda - m)^2 - m] \lambda^m b}{\lambda(\lambda - m)^2 m^m}. \tag{54}$$

Substituting (53) and (54) into the left side of (42), we get

$$\begin{aligned} & cm^m + d\lambda^m \\ &= \frac{m^{m+1} \lambda^{m-1} a + [m - (\lambda - m)^2] \lambda^{2m-1} b}{(\lambda - m)^2 m^m} \\ &+ \frac{[(\lambda - m)^2 - m] m^m \lambda^{m-1} a + [2(\lambda - m)^2 - m] \lambda^{2m-1} b}{(\lambda - m)^2 m^m} \\ &= \frac{m^m a + \lambda^m b}{m^m} \lambda^{m-1}, \end{aligned} \tag{55}$$

which shows the condition (42) is equivalent to

$$am^m + b(m - \alpha)^m \neq 0. \tag{56}$$

We summarize our development of new methods done so far in the following theorem.

**Theorem 5.** Let  $x^* \in I$  be a multiple root of multiplicity  $m (m > 1)$  of a sufficiently differentiable function  $f : I \rightarrow \mathbb{R}$  for an open interval  $I$ . If  $x_0$  is sufficiently close to  $x^*$ , then the methods defined by (I)-type iteration (12) are cubically convergent for any parameters  $\alpha, a, b, c,$  and  $d$  such that  $\alpha \neq 0$  and (53), (54), and (56) hold.

Next, we turn to find the proper conditions to establish the relation (44). In view of (32), and by a straight computation, we can get

$$\Psi_n = r_n + s_n e_n + O(e_n^2), \tag{57}$$

where

$$\begin{aligned} r_n = & (m^m a + \lambda^m b - m^m \lambda c - \lambda^{m+1} d) m^2 g_n^{m+3}, \\ s_n = & ((m+2) m^{m+1} a + [(2-\alpha) m \lambda^m + m^3 \lambda^{m-1}] b \\ & - [m^{m+2} + m^{m+1} (m+2) \lambda] c \\ & - [(2-\alpha) m \lambda^{m+1} + m^2 (m+1) \lambda^m] d) g_n^{m+2} h_n. \end{aligned} \tag{58}$$

In view of (57) and in order to assure the relation (44) holds, we should choose parameters  $\lambda, a, b, c,$  and  $d$  such that

$$\begin{aligned} m^m a + \lambda^m b - m^m \lambda c - \lambda^{m+1} d = 0, \\ (m+2) m^m a + [(2-\alpha) \lambda + m^2] \lambda^{m-1} b \\ - [m + (m+2) \lambda] m^m c \\ - [(2-\alpha) \lambda + m(m+1)] \lambda^m d = 0. \end{aligned} \tag{59}$$

By a straight computation, we deduce that

$$\begin{aligned} c = & \frac{[(\lambda - m)^2 + m] m^{m-1} a + \lambda^m b}{\lambda(\lambda - m)^2 m^{m-1}}, \\ d = & \frac{-m^{m+1} a + [(\lambda - m)^2 - m] \lambda^m b}{\lambda^{m+1} (\lambda - m)^2}. \end{aligned} \tag{60}$$

Substituting (60) into the left side of (42), we get

$$cm^m + d\lambda^m = \frac{m^m a + \lambda^m b}{\lambda}, \tag{61}$$

which shows the condition (42) is also equivalent to the condition given by (56).

We can summarize the development of new methods involving (II)-type iteration (13) done so far in the following theorem.

**Theorem 6.** Let  $x^* \in I$  be a multiple root of multiplicity  $m (m > 1)$  of a sufficiently differentiable function  $f : I \rightarrow \mathbb{R}$  for an open interval  $I$ . If  $x_0$  is sufficiently close to  $x^*$ , then the methods defined by (II)-type iteration (13) are cubically convergent for any parameters  $\alpha, a, b, c,$  and  $d$  such that  $\alpha \neq 0$  and (60) and (56) hold.

Choosing  $\alpha = \sqrt{m}$ ,  $a = 1$ , and  $b = 0$ , we can deduce from (53), (54), and (56) that

$$c = \frac{(m - \sqrt{m})^{m-1}}{m^m}, \quad (62)$$

$$d = 0, \quad (63)$$

$$am^m + b(m - \alpha)^m = m^m \neq 0. \quad (64)$$

Using the parameters  $\alpha$ ,  $a$ ,  $b$ ,  $c$ , and  $d$  given above in (I)-type iteration (12), we can obtain Dong's method (4), and its order of convergence arrives at three by Theorem 5.

Choosing  $\alpha = 1$ ,  $a = 1$ , and  $b = 0$ , we can deduce from (53), (54), and (56) that

$$c = \frac{(m - 1)^{m-1}}{m^{m-1}},$$

$$d = -1, \quad (65)$$

$$am^m + b(m - \alpha)^m = m^m \neq 0.$$

Using the parameters  $\alpha$ ,  $a$ ,  $b$ ,  $c$ , and  $d$  given above in (I)-type iteration (12), we can obtain Dong's method (5), and its order of convergence arrives at three by Theorem 5.

Choosing  $\alpha = 1$ ,  $a = 1$ , and  $b = (m/(m - 1))^{2m} - (m/(m - 1))^{m+1}$ , we can deduce from (53), (54), and (56) that

$$c = 1, \quad (66)$$

$$d = -\frac{(m/(m - 1))^m (m - 2) (m - 1) + 1}{(m - 1)^2}, \quad (67)$$

$$am^m + b(m - \alpha)^m = m^m \left[ 1 + \left( \frac{m}{m - 1} \right)^m - \frac{m}{m - 1} \right] \neq 0. \quad (68)$$

Using the parameters  $\alpha$ ,  $a$ ,  $b$ ,  $c$ , and  $d$  given above in (I)-type iteration (12), we can obtain Victory and Neta's method (6), and its order of convergence arrives at three by Theorem 5.

Let  $\alpha = (1/2)(m(m + 3)/(m + 1))$ ,  $a = (m^3 + 4m^2 + 9m + 2)/(m + 3)^2 - (1/2)(m(m + 3)/(m + 1))$ ,  $b = 2^{m+1}(m^2 - 1)/(m + 3)^2((m - 1)/(m + 1))^m$ ,  $c = 0$ , and  $d = 1$ . Using the definition of  $\lambda$ , we get

$$\lambda = \frac{m(m - 1)}{2(m + 1)}. \quad (69)$$

We can verify that the parameters given above satisfy (52), and thus they also satisfy (53) and (54) (as  $c$  given in (53) and  $d$  given in (54) are solved from (52)). Furthermore, it is easy to verify that the condition (42) holds:

$$cm^m + d\lambda^m = \left( \frac{m(m - 1)}{2(m + 1)} \right)^m \neq 0. \quad (70)$$

This means that the condition (56) is also true, since (56) is equivalent to (42). Using the parameters  $\alpha$ ,  $a$ ,  $b$ ,  $c$ , and  $d$  given in (I)-type iteration (12), we can obtain Neta's method (8), and its order of convergence arrives at three by Theorem 5.

We can verify that the family of methods (11) given by Homeier [16] satisfies all conditions in Theorem 5. First, we can rewrite (11) as follows:

$$y_n = x_n - \alpha \frac{f(x_n)}{f'(x_n)},$$

$$x_{n+1} = y_n - \frac{(\beta - \alpha) f(x_n) + (1/\gamma) f(y_n)}{f(y_n)} \frac{f(y_n)}{f'(x_n)}, \quad (71)$$

where  $\alpha \neq 0$ ,  $m$  is a real parameter,  $\beta = (m/\alpha^2)(\alpha^2 + (\alpha - m)) = (m/\alpha^2)(\alpha^2 - \lambda)$ , and  $\gamma = (1/m)(1 - \alpha/m)^m(\alpha^2/(m - \alpha)) = (\lambda^{m-1}\alpha^2/m^{m+1})$ . Choosing  $a = \beta - \alpha$  and  $b = 1/\gamma$ , we can deduce from (53), (54), and (56) that  $c = 0$ ,  $d = 1$ , and

$$cm^m + d\lambda^m = (m - \alpha)^m \neq 0. \quad (72)$$

This means that the condition (56) is also true. Using the parameters  $\alpha$ ,  $a$ ,  $b$ ,  $c$ , and  $d$  given in (I)-type iteration (12), we can obtain Homeier's family of methods (11), which has cubic convergence by Theorem 5.

We can verify that the family of methods (10) given by Homeier [16] satisfies all conditions in Theorem 6. First, we can rewrite (10) as follows:

$$y_n = x_n - \alpha \frac{f(x_n)}{f'(x_n)}, \quad (73)$$

$$x_{n+1} = y_n - \left( (m\alpha^2 - m\alpha + \alpha^2 - \alpha^3) \right. \\ \left. \times \mu^m f(x_n) + \alpha(m - \alpha) f(y_n) \right) \\ \times \left( (m - \alpha + \alpha^2) \mu^m f(x_n) - (m - \alpha) f(y_n) \right)^{-1} \\ \times \frac{f(x_n)}{f'(x_n)}, \quad (74)$$

where  $\alpha$  is a real parameter and  $\alpha \neq 0$ ,  $m$ , and  $\mu = (m - \alpha)/m$ .

Let  $a = \lambda^{m+1}\alpha(\alpha - 1)/m^m$ ,  $b = \alpha\lambda$ ,  $c = (\lambda + \alpha^2)\lambda^m/m^m$ , and  $d = -\lambda$ . We can verify that the parameters given above satisfy (60) and (56). Using the parameters  $\alpha$ ,  $a$ ,  $b$ ,  $c$ , and  $d$  given in (II)-type iteration (13), we can obtain Shengguo et al.'s family of methods (10), which has cubic convergence by Theorem 6.

#### 4. Some Concrete Methods

In this section, we give some concrete iterative forms of (I)-type iteration (12) and (II)-type iteration (13).

*Method 1.* Choosing  $\alpha = 1$ ,  $a = 0$ , and  $b = 1$ , we obtain from (63) and (64) that  $c = ((m - 1)/m)^{2m}$ ,  $d = (2 - m)(m - 1)^{m-1}/m^m$ , and  $am^m + b(m - \alpha)^m = (m - 1)^m \neq 0$ .

Using these parameters in (12), we get a new method. Consider

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n - (f(y_n)) \times \left( \left( \frac{m-1}{m} \right)^{2m} f(x_n) \right. \\
 &\quad \left. + \frac{(2-m)(m-1)^{m-1}}{m^m} f(y_n) \right)^{-1} \\
 &\quad \times \frac{f(y_n)}{f'(x_n)},
 \end{aligned} \tag{75}$$

which has cubic convergence by Theorem 5.

*Method 2.* Choosing  $\alpha = 1$ ,  $c = 1$ , and  $d = 0$ , we can obtain from (61) and (62) that  $a = (2-m)m^m/(m-1)^{m-1}$ ,  $b = m^{2m}/(m-1)^{2m-2}$ , and  $cm^m + d(m-\alpha)^m = m^m \neq 0$ . Using these parameters in (12), we get a new method. Consider

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n - \left( \frac{(2-m)m^m}{(m-1)^{m-1}} f(x_n) \right. \\
 &\quad \left. + \frac{m^{2m}}{(m-1)^{2m-2}} f(y_n) \right) \\
 &\quad \times (f(x_n))^{-1} \\
 &\quad \times \frac{f(y_n)}{f'(x_n)},
 \end{aligned} \tag{76}$$

which has cubic convergence by Theorem 5.

*Method 3.* Choosing  $\alpha = 1$ ,  $b = 1$ , and  $d = 1$ , we can obtain from (61) and (62) that  $a = ((m-1)^{m-1}(2-m) - m^m)/m^m$ ,  $c = ((m-1)^{2m-2} - m^{m+1}(m-1)^{m-1})/m^{2m}$ , and  $cm^m + d(m-\alpha)^m = (m-1)^{m-1}[(m-1)^{m-1} - m^m]/m^m \neq 0$ . Using these parameters in (12), we get a new method. Consider

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n - \left( \frac{(m-1)^{m-1}(2-m) - m^m}{m^m} f(x_n) + f(y_n) \right) \\
 &\quad \times \left( \frac{(m-1)^{2m-2} - m^{m+1}(m-1)^{m-1}}{m^{2m}} \right. \\
 &\quad \left. \times f(x_n) + f(y_n) \right)^{-1} \\
 &\quad \times \frac{f(y_n)}{f'(x_n)},
 \end{aligned} \tag{77}$$

which has cubic convergence by Theorem 5.

*Method 4.* Choosing  $\alpha = 1$ ,  $a = 1$ , and  $b = 0$ , we can obtain from (70) and (71) that  $c = (m+1)/(m-1)$ ,  $d =$

$-(m/(m-1))^{m+1}$ , and  $am^m + b(m-\alpha)^m = m^m \neq 0$ . Using these parameters in (13), we get a new method. Consider

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n - (f(x_n)) \\
 &\quad \times \left( \frac{m+1}{m-1} f(x_n) - \left( \frac{m}{m-1} \right)^{m+1} f(y_n) \right)^{-1} \\
 &\quad \times \frac{f(x_n)}{f'(x_n)},
 \end{aligned} \tag{78}$$

which has cubic convergence by Theorem 6.

*Method 5.* Choosing  $\alpha = 1$ ,  $a = 1$ , and  $c = 0$ , we can obtain from (68) and (69) that  $b = -(m+1)m^{m-1}/(m-1)^m$ ,  $d = -m^{m-1}/(m-1)^{m+1}$ , and  $cm^m + d(m-\alpha)^m = -m^{m-1}/(m-1) \neq 0$ . Using these parameters in (13), we get a new method. Consider

$$\begin{aligned}
 y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n + \left( f(x_n) - \frac{(m+1)m^{m-1}}{(m-1)^m} f(y_n) \right) \\
 &\quad \times \left( \frac{m^{m-1}}{(m-1)^{m+1}} f(y_n) \right)^{-1} \\
 &\quad \times \frac{f(x_n)}{f'(x_n)},
 \end{aligned} \tag{79}$$

which has cubic convergence by Theorem 6.

*Method 6.* Choosing  $\alpha = \sqrt{m}$ ,  $a = 1$ , and  $b = 0$ , we can obtain from (70) and (71) that  $c = 2/(m-\sqrt{m})$ ,  $d = -m^m/(m-\sqrt{m})^{(m+1)}$ , and  $am^m + b(m-\alpha)^m = m^m \neq 0$ . Using these parameters in (13), we get a new method. Consider

$$\begin{aligned}
 y_n &= x_n - \sqrt{m} \frac{f(x_n)}{f'(x_n)}, \\
 x_{n+1} &= y_n - (f(x_n)) \\
 &\quad \times \left( \frac{2}{m-\sqrt{m}} f(x_n) - \frac{m^m}{(m-\sqrt{m})^{m+1}} f(y_n) \right)^{-1} \\
 &\quad \times \frac{f(x_n)}{f'(x_n)},
 \end{aligned} \tag{80}$$

which has cubic convergence by Theorem 6.

### 5. Numerical Examples

We employ Method 1 (RM1), (75)–Method 6 (RM6), (80) to solve some nonlinear equations and compare them with the

TABLE 1: Comparison of various iterative methods for the function  $f_1(x)$  under the same total number of function evaluations (TNFE) required by all methods.

Method	$ x_n - x^* $	$ f(x_n) $	COC
MNM	$3.92e - 94$	$9.48e - 187$	2.0000000
DM	$4.83e - 117$	$1.44e - 232$	3.0000000
VM	$1.42e - 118$	$1.24e - 235$	3.0000000
RM1	$4.63e - 116$	$1.32e - 230$	3.0000000
RM2	$4.63e - 116$	$1.32e - 230$	3.0000000
RM3	$5.53e - 130$	$1.88e - 258$	3.0000000
RM4	$3.98e - 140$	$9.77e - 279$	3.0000000
RM5	$1.14e - 102$	$7.95e - 204$	3.0000000
RM6	$1.04e - 103$	$6.71e - 206$	3.0000000

TNFE = 12.

TABLE 2: Comparison of various iterative methods for the function  $f_2(x)$  under the same total number of function evaluations (TNFE) required by all methods.

Method	$ x_n - x^* $	$ f(x_n) $	COC
MNM	$2.83e - 95$	$1.06e - 283$	2.0000000
DM	$1.88e - 109$	$3.11e - 326$	3.0000000
VM	$3.19e - 107$	$1.52e - 319$	3.0000000
RM1	$2.04e - 106$	$4.00e - 317$	3.0000000
RM2	$2.59e - 104$	$8.11e - 311$	3.0000000
RM3	$8.19e - 110$	$2.57e - 327$	3.0000000
RM4	$2.09e - 113$	$4.27e - 338$	3.0000000
RM5	$4.66e - 100$	$4.73e - 298$	3.0000000
RM6	$1.05e - 144$	$5.39e - 432$	3.0000000

TNFE = 12.

TABLE 3: Comparison of various iterative methods for the function  $f_3(x)$  under the same total number of function evaluations (TNFE) required by all methods.

Method	$ x_n - x^* $	$ f(x_n) $	COC
MNM	$1.53e - 119$	$4.09e - 478$	2.0000000
DM	$5.39e - 167$	$6.27e - 668$	3.0000000
VM	$1.91e - 176$	$9.90e - 706$	3.0000000
RM1	$9.39e - 185$	$5.79e - 739$	3.0000000
RM2	$3.08e - 169$	$6.69e - 677$	3.0000000
RM3	$9.92e - 164$	$7.21e - 655$	3.0000000
RM4	$1.38e - 156$	$2.73e - 626$	3.0000000
RM5	$1.84e - 152$	$8.48e - 610$	3.0000000
RM6	$7.89e - 143$	$2.89e - 571$	3.0000000

TNFE = 12.

modified Newton method (MNM), (1), Dong's method (DM), (4), and Victory-Neta's method (VM).

Tables 1, 2, and 3 show the difference of the root  $x^*$  and the approximation  $x_n$  to  $x^*$  for the function  $f_1 - f_3$ , respectively, where  $x^*$  is the exact root computed with 650 significant digits and  $x_n$  is calculated by using the same total number of function evaluations (TNFE) for all methods. The absolute values of the function ( $|f(x_n)|$ ) and the computational order

of convergence (COC) are also shown in these tables. Here, COC is defined by [27]

$$\rho \approx \frac{\ln |(x_{n+1} - x^*) / (x_n - x^*)|}{\ln |(x_n - x^*) / (x_{n-1} - x^*)|}. \quad (81)$$

The following functions are used for the comparison:

$$f_1(x) = (\sin^2 x - x^2 + 1)^2,$$

$$x_0 = 1.45,$$

$$x^* \approx 1.4044916482153412260350868178,$$

$$f_2(x) = (\cos x - x)^3,$$

$$x_0 = 0.9,$$

$$x^* \approx 0.73908513321516064165531208767,$$

$$f_3(x) = (\ln(x) + \sqrt{x} - 5)^4,$$

$$x_0 = 8.0,$$

$$x^* \approx 8.3094326942315717953469556827.$$

As shown in Tables 1, 2, and 3, the presented methods in this contribution are preferable to the modified Newton method by numerical tests. Note also that the presented methods show at least equal performance as compared with some other methods of the same order.

## 6. Conclusion

In this work, we obtained two families of third-order methods by using new techniques of divided differences for solving nonlinear equations with multiple roots. The proposed methods contain many known methods, such as Dong's methods and Neta's method. We conclude from the numerical examples that the proposed methods have at least equal performance as compared with the other iterative methods of the same order. Moreover, it was observed that these methods have better performance than the modified Newton method. We will continue our study to confirm if some fourth-order methods can be obtained using the same ideas in this contribution.

## Conflict of Interests

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

## Acknowledgments

This work is supported by the National Basic Research 973 Program of China (no. 2011JB105001), National Natural Science Foundation of China (Grant no. 11371320), Zhejiang Natural Science Foundation (Grant no. LZ14A010002), the Foundation of Science and Technology Department (Grant no. 2013C31084) of Zhejiang Province, Scientific Research



Fund of Zhejiang Provincial Education Department (nos. Y201431077 and Y201329420), and the Grant FEKT-S-11-2-921 of Faculty of Electrical Engineering and Communication, Brno University of Technology, Czech Republic.

## References

- [1] E. Schröder, "Ueber unendlich viele Algorithmen zur Auflösung der Gleichungen," *Mathematische Annalen*, vol. 2, no. 2, pp. 317–365, 1870.
- [2] E. Hansen and M. Patrick, "A family of root finding methods," *Numerische Mathematik*, vol. 27, no. 3, pp. 257–269, 1976.
- [3] J. F. Traub, *Iterative Methods for the Solution of Equations*, Chelsea, New York, NY, USA, 1977.
- [4] C. Dong, "A basic theorem of constructing an iterative formula of the higher order for computing multiple roots of an equation," *Mathematica Numerica Sinica*, vol. 11, pp. 445–450, 1982.
- [5] H. D. Victory Jr. and B. Neta, "A higher order method for multiple zeros of nonlinear functions," *International Journal of Computer Mathematics*, vol. 12, no. 3-4, pp. 329–335, 1983.
- [6] C. Dong, "A family of multipoint iterative functions for finding multiple roots of equations," *International Journal of Computer Mathematics*, vol. 21, pp. 363–367, 1987.
- [7] N. Osada, "An optimal multiple root-finding method of order three," *Journal of Computational and Applied Mathematics*, vol. 51, no. 1, pp. 131–133, 1994.
- [8] M. Frontini and E. Sormani, "Modified Newton's method with third-order convergence and multiple roots," *Journal of Computational and Applied Mathematics*, vol. 156, no. 2, pp. 345–354, 2003.
- [9] B. Neta, "New third order nonlinear solvers for multiple roots," *Applied Mathematics and Computation*, vol. 202, no. 1, pp. 162–170, 2008.
- [10] B. Neta and A. N. Johnson, "High-order nonlinear solver for multiple roots," *Computers and Mathematics with Applications*, vol. 55, no. 9, pp. 2012–2017, 2008.
- [11] B. Neta, "Extension of Murakami's high-order non-linear solver to multiple roots," *International Journal of Computer Mathematics*, vol. 87, no. 5, pp. 1023–1031, 2010.
- [12] C. Chun and B. Neta, "A third-order modification of Newton's method for multiple roots," *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 474–479, 2009.
- [13] C. Chun, H. J. Bae, and B. Neta, "New families of nonlinear third-order solvers for finding multiple roots," *Computers and Mathematics with Applications*, vol. 57, no. 9, pp. 1574–1582, 2009.
- [14] L. Shengguo, L. Xiangke, and C. Lizhi, "A new fourth-order iterative method for finding multiple roots of nonlinear equations," *Applied Mathematics and Computation*, vol. 215, no. 3, pp. 1288–1292, 2009.
- [15] S. Li, H. Li, and L. Cheng, "Some second-derivative-free variants of Halley's method for multiple roots," *Applied Mathematics and Computation*, vol. 215, no. 6, pp. 2192–2198, 2009.
- [16] H. H. Homeier, "On Newton-type methods for multiple roots with cubic convergence," *Journal of Computational and Applied Mathematics*, vol. 231, no. 1, pp. 249–254, 2009.
- [17] M. S. Petkovi, J. Duni, and M. Miloevi, "Traub's accelerating generator of iterative root-finding methods," *Applied Mathematics Letters*, vol. 24, no. 8, pp. 1443–1448, 2011.
- [18] B. I. Yun, "Iterative methods for solving nonlinear equations with finitely many roots in an interval," *Journal of Computational and Applied Mathematics*, vol. 236, no. 13, pp. 3308–3318, 2012.
- [19] P. Jarratt, "Some fourth order multipoint methods for solving equations," *Mathematics of Computation*, vol. 20, pp. 434–437, 1966.
- [20] S. Kumar, V. Kanwar, and S. Singh, "On some modified families of multipoint iterative methods for multiple roots of nonlinear equations," *Applied Mathematics and Computation*, vol. 218, no. 14, pp. 7382–7394, 2012.
- [21] X. Zhou, X. Chen, and Y. Song, "Families of third and fourth order methods for multiple roots of nonlinear equations," *Applied Mathematics and Computation*, vol. 219, no. 11, pp. 6030–6038, 2013.
- [22] B. Neta and C. Chun, "On a family of Laguerre methods to find multiple roots of nonlinear equations," *Applied Mathematics and Computation*, vol. 219, no. 23, pp. 10987–11004, 2013.
- [23] D. Sbibaha, A. Serghini, A. Tijini, and A. Zidna, "A general family of third order method for finding multiple roots," *Applied Mathematics and Computation*, vol. 233, pp. 338–350, 2014.
- [24] B. I. Yun, "Transformation methods for finding multiple roots of nonlinear equations," *Applied Mathematics and Computation*, vol. 217, no. 2, pp. 599–606, 2010.
- [25] R. F. King, "A family of fourth order methods for nonlinear equations," *SIAM Journal on Numerical Analysis*, vol. 10, pp. 876–879, 1973.
- [26] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer, New York, NY, USA, 1980.
- [27] S. Weerakoon and T. G. I. Fernando, "A variant of Newton's method with accelerated third-order convergence," *Applied Mathematics Letters*, vol. 13, no. 8, pp. 87–93, 2000.