A new family of fourth-order methods for multiple roots of nonlinear equations*

Baoqing Liu[a], **Xiaojian Zhou**[a,b,1]

[a]Jiangsu Key Laboratory for NSLSCS, Institute of Mathematics
School of Mathematical Sciences, Nanjing Normal University
Nanjing 210023, China
lyberal@163.com

[b]School of Science, Nantong University
Nantong, 226008, China
zxjntu@gmail.com

Received: 8 October 2012 / **Revised:** 10 January 2013 / **Published online:** 18 April 2013

Abstract. Recently, some optimal fourth-order iterative methods for multiple roots of nonlinear equations are presented when the multiplicity m of the root is known. Different from these optimal iterative methods known already, this paper presents a new family of iterative methods using the modified Newton's method as its first step. The new family, requiring one evaluation of the function and two evaluations of its first derivative, is of optimal order. Numerical examples are given to suggest that the new family can be competitive with other fourth-order methods and the modified Newton's method for multiple roots.

Keywords: nonlinear equations, iterative method, multiple roots, the modified Newton's method, optimal order.

1 Introduction

Solving nonlinear equations is one of the most important problems in numerical analysis. In this paper, we consider iterative methods to find the multiple root $x^\star$ of a nonlinear equation $f(x) = 0$ with multiplicity m, i.e., $f^{(i)}(x^\star) = 0$, $i = 0, 1, \dots, m-1$, and $f^{(m)}(x^\star) \neq 0$.

The modified Newton's method is one of the most well known iterative methods for multiple roots and defined by [1]

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

which converges quadratically.

*This work is supported by the National Natural Science Foundation of China (11271196) and the Foundation for Innovative Program of Jiangsu Province (CXZZ11_0870, CXZZ12_0383).

[1]Corresponding author.

In order to improve the convergence of iterative methods for multiple roots, some researchers, such as Dong [2, 3], Neta et al. [4–7], Li et al. [8, 9], have developed some iterative methods with higher order of convergence. These methods require the evaluations of the target function and its first order derivative. Also, there are some methods requiring the second order derivative, such as Osada [10], Euler–Chebyshev [11], Neta et al. [12, 13]. For more iterative methods, see also Neta [14]. Most of these methods require the knowledge of the multiplicity m.

According to the famous Traub's conjecture of optimal order for the iterative method without memory [15], the modified Newton's method is optimal since two function/derivative-evaluations are required per step. Recently, some optimal fourth-order iterative methods have been developed [8, 9, 16, 17]. For example, Sharma and Sharma [16] have developed the following variant of Jarratt's method for multiple roots:

$$\begin{aligned} y_n &= x_n - \frac{2m}{2+m}\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \frac{m}{8}\left[\left(m^3-4m+8\right)-(m+2)^2\left(\frac{m}{m+2}\right)^m\frac{f'(x_n)}{f'(y_n)}\right. \\ &\quad \times\left.\left(2(m-1)-(m+2)\left(\frac{m}{m+2}\right)^m\frac{f'(x_n)}{f'(y_n)}\right)\right]\frac{f(x_n)}{f'(x_n)}. \end{aligned} \tag{2}$$

Li et al. have developed six fourth-order methods with closed formulae for multiple roots [9], based on the results of Neta and Johnson [5] and Neta [7]. The following is one of them, which has optimal order:

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - a_1\frac{f(x_n)}{f'(x_n)} - \frac{f(x_n)}{b_1 f'(x_n) + b_2 f'(y_n)} \end{aligned} \tag{3}$$

with $a_1 = -(1/2)m(m-2)$, $b_1 = -(1/m)$, $b_2 = (1/m)((2+m)/m)^m$.

Very recently, we have developed a more general iteration scheme for multiple roots [17]

$$\begin{aligned} y_n &= x_n - t\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - Q\left(\frac{f'(y_n)}{f'(x_n)}\right)\frac{f(x_n)}{f'(x_n)}, \end{aligned} \tag{4}$$

where the function $Q(\cdot) \in C^2(\mathbb{R})$ and t is an undefined parameter. We have shown that the convergence order of the family (4) is four at least, when $t = 2m/(2+m)$ and the following conditions hold:

$$Q(u) = m, \qquad Q'(u) = -\frac{1}{4}m^{3-m}(2+m)^m, \qquad Q''(u) = \frac{1}{4}m^4\left(\frac{m}{2+m}\right)^{-2m},$$

where $u = (m/(2+m))^{m-1}$. Obviously, the fourth-order iterative family (4) is of optimal order. It has been pointed out (4) contains almost all optimal fourth-order methods for multiple roots known already, including (2), (3) and methods presented in [8] (see Section 3 in [17] for details).

Remark 1. For $m = 1$, that is simple roots of nonlinear equations, iteration (4) and the corresponding result in [17] are also valid.

There are two interesting phenomena in (2)–(4). One is that the first step is of order one while the second step is of order four, which is not coincide with the experience of constructing higher-order iterative method for simple roots. The other is that according to the local analysis in [17], if the modified Newton's method is used in the first step of (4), that is, $t = m$, then only quadratically convergent iterative methods can be obtained no matter what $Q(\cdot)$ is. However, almost all the higher order multistep iterative methods for simple roots use the Newton's method or some other second-order iterations as the first step. So in this paper, based on the evaluations of $f(x_n)$, $f'(x_n)$ and $f'(y_n)$, we try to construct a fourth-order optimal iterative scheme, using the modified Newton's method as the first step. Meanwhile, we always assume that the multiplicity $m \geqslant 2$.

2 New family of fourth-order methods

For our purpose, we first investigate the Taylor expression of $f'(y_n)/f'(x_n)$.

Let $e_n = x_n - x^\star$. Expanding $f(x_n)$ and $f'(x_n)$ at $x = x^\star$ with Taylor series, we have

$$f(x_n) = \frac{f^{(m)}(x^\star)}{m!} e_n^m \bigl(1 + c_1 e_n + c_2 e_n^2 + c_3 e_n^3 + O\bigl(e_n^4\bigr)\bigr)$$

and

$$f'(x_n) = \frac{f^{(m)}(x^\star)}{(m-1)!} e_n^{m-1} \left(1 + \frac{m+1}{m} c_1 e_n + \frac{m+2}{m} c_2 e_n^2 + \frac{m+3}{m} c_3 e_n^3 + +O\bigl(e_n^4\bigr)\right),$$

where $c_i = (m!/i!) f^{(i)}(x^\star)/f^{(m)}(x^\star)$ and $i \geqslant 1$.

Hence we have

$$\begin{aligned} \frac{f(x_n)}{f'(x_n)} = {} & \frac{1}{m} e_n - \frac{c_1}{m^2} e_n^2 + \frac{c_1^2(m+1) - 2mc_2}{m^3} e_n^3 \\ & + \frac{c_1 c_2 m(3m+4) - 3m^2 c_3 - (m+1)^2 c_1^3}{m^4} e_n^4 + O\bigl(e_n^5\bigr). \end{aligned} \tag{5}$$

Then, for the modified Newton's step, we get

$$\begin{aligned} d_n & \triangleq y_n - x^* \\ & = \frac{c_1}{m} e_n^2 - \frac{(m+1)c_1^2 - 2mc_2}{m^2} e_n^3 \\ & \quad + \frac{(m+1)^2 c_1^3 - (3m+4)mc_1 c_2 + 3m^2 c_3}{m^3} e_n^4 + O\bigl(e_n^5\bigr) \end{aligned}$$

and

$$f'(y_n) = m + (m+1)c_1 d_n + (m+2)c_2 d_n^2 + (m+3)c_3 d_n^3 + O\big(d_n^4\big).$$

So, we can get

$$\frac{f'(y_n)}{f'(x_n)} = \left(\frac{c_1}{m}\right)^{m-1} e_n^{m-1}\big(1 + \psi_1 e_n + \psi_2 e_n^2 + O\big(e_n^3\big)\big),$$

where

$$\begin{aligned}
\psi_1 &= 2(m-1)\frac{c_2}{c_1} - (m+1)c_1,\\
\psi_2 &= 3\frac{c_3}{c_1}(m-1) + 2\frac{c_2^2}{c_1^2}(m-2)(m-1) - m(2m+1)c_2\\
&\quad + \frac{(m+1)(m+2)(m^2+1)}{2m^2}c_1^2.
\end{aligned}$$

Noting that $d_n = O(e_n^2)$ and $f(x_n)/f'(x_n) = O(e_n)$, if we want to obtain an iterative scheme of order three at least, $G(f'(y_n)/f'(x_n))$ should be the same order as e_n. From $f'(y_n)/f'(x_n) = O(e_n^{m-1})$, we have $\sqrt[m-1]{f'(y_n)/f'(x_n)} = O(e_n)$. Thus, to obtain a higher-order iterative family, we can pin our hope on the following revised scheme:

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

where $w_n = \sqrt[m-1]{f'(y_n)/f'(x_n)}$.

Since

$$\begin{aligned}
(1+x)^{1/(m-1)} &= 1 + \frac{x}{m-1} - \frac{m-2}{2(m-1)^2}x^2 + \frac{(m-2)(2m-3)}{6(m-1)^3}x^3\\
&\quad - \frac{(m-2)(2m-3)(3m-4)}{24(m-1)^4}x^4 + O\big(x^5\big),
\end{aligned}$$

we have

$$\begin{aligned}
w_n &= \frac{c_1}{m}\bigg(e_n + \frac{\psi_1}{m-1}e_n^2 - \frac{(m-2)\psi_1^2 - 2(m-1)\psi_2}{2(m-1)^2}e_n^3\\
&\quad + \frac{(m-2)\psi_1((2m-3)\psi_1^2 - 6(m-1)\psi_2)}{6(m-1)^3}e_n^4 + O\big(e_n^5\big)\bigg).
\end{aligned} \tag{7}$$

Submitting (5) and (7) into the second equation of (6), we have

$$\begin{aligned}
e_{n+1} &= -G(0)e_n + \frac{1+G(0)-G'(0)}{m}c_1 e_n^2 + \bigg(\frac{2}{m}\big(1+G(0)-G'(0)\big)c_2\\
&\quad - \bigg(\frac{m+1}{m^2} + \frac{m+1}{m^2}G(0) - \frac{m^2+2m-1}{(m-1)m^2}G'(0) + \frac{1}{2m^2}G''(0)\bigg)c_1^2\bigg)e_n^3\\
&\quad + \frac{K}{6(m-1)^2m^3}e_n^4 + O\big(e_n^5\big),
\end{aligned}$$

where

$$\begin{aligned}
K &= 18(m+1)^2 m^2\bigl(1+G(0)-G'(0)\bigr)c_3\\
&\quad + 6(m+1)m\bigl(4-m-3m^2+\bigl(4-m-3m^2\bigr)G(0)\\
&\quad + \bigl(-4+8m+3m^2\bigr)G'(0)+(2-2m)G''(0)\bigr)c_1c_2\\
&\quad + \bigl(6-12m^2+6m^4+\bigl(6-12m^2+6m^4\bigr)G(0)\\
&\quad + \bigl(15m-21m^3-6m^4\bigr)G'(0)+\bigl(3-12m+9m^2\bigr)G''(0)\\
&\quad - (1+m)^2G'''(0)\bigr)c_1^3.
\end{aligned}$$

To obtain a fourth-order method, the coefficients of e_n, e_n^2 and e_n^3 should all be zeros. Furthermore, we hope the fourth-order method is independent of the function $f(x)$, that is $c_1, c_2, c_3, \dots$, thus we shall solve the equations

$$\begin{aligned}
&G(0)=0,\\
&1+G(0)-G'(0)=0,\\
&\frac{m+1}{m^2}+\frac{m+1}{m^2}G(0)-\frac{m^2+2m-1}{(m-1)m^2}G'(0)+\frac{1}{2m^2}G''(0)=0.
\end{aligned}$$

By simple computation, we have

$$G(0)=0, \qquad G'(0)=1, \qquad G''(0)=\frac{4m}{m-1}. \tag{8}$$

Furthermore, we get the corresponding error equation

$$\begin{aligned}
e_{n+1} &= \frac{1}{6(m-1)^2m^3}\bigl(\bigl(3\bigl(m^3+8m^2+m+2\bigr)-(m-1)^2G'''(0)\bigr)c_1^3\\
&\quad - 6(m-1)m^2c_1c_2\bigr)e_n^4+O\bigl(x^5\bigr).
\end{aligned} \tag{9}$$

The above discussion shows the following conclusion.

Theorem 1. *Let $x^\star \in \mathbb{R}$ be a multiple root of multiplicity m of a sufficiently differentiable function $f: I \to \mathbb{R}$ for an open interval I. If the initial point x_0 is sufficiently close to $x^\star$, then, when $G(0)=0$, $G'(0)=1$, $G''(0)=4m/(m-1)$ and $G'''(0)<+\infty$, the convergence order of method defined by* (6) *is four at least with the error equation given by* (9).

Remark 2. Consider the definition of efficiency index as $p^{1/q}$, where p is the order of the method and q is the number of function evaluations per iteration required by the method. The fourth-order family (6) has the efficiency index $4^{1/3} \approx 1.587$, which equals to those of (2)–(4) and is better than $2^{1/2} \approx 1.414$ of the modified Newton's method (1).

Noting that $w_n = \sqrt[m-1]{f'(y_n)/f'(x_n)}$ is used in iteration (6) and the condition $G''(0)=4m/(m-1)$, Theorem 1 cannot be held for the case of simple roots, i.e., $m=1$. In fact, after similar computations, we can deduce the following conclusion.

Theorem 2. *Let $x^\star \in \mathbb{R}$ be a simple root of a sufficiently differentiable function $f : I \to \mathbb{R}$ for an open interval I. If the initial point x_0 is sufficiently close to $x^\star$, then, when $G(1)=0$, $G'(1) = -1/2$ and $G'''(0) < +\infty$, the convergence order of method defined by*

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - G\left(\frac{f'(y_n)}{f'(x_n)}\right)\frac{f(x_n)}{f'(x_n)} \end{aligned} \tag{10}$$

is three at least with the error equation given by

$$e_{n+1} = \left(\frac{c_2}{2} - 2\big(G''(1) - 1\big)c_1^2\right)e_n^3 + O\big(e_n^4\big).$$

Remark 3. This paper concerns on the iterative methods for multiple roots only, so we will not do further discussion on (10).

Remark 4. Due to the first step in (6), all the members given by (6) are different from those given by (4).

3 Two special members of (6)

Iterative family (6) for multiple roots of nonlinear equations can deduce a lot of optimal fourth-order iterative methods according to the conditions (8). For simplicity, we only give two of them in this section.

Case 1. From (8), it is easy to obtain the fourth-order iteration

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - m\left(w_n + \frac{2m}{m-1}w_n^2\right)\frac{f(x_n)}{f'(x_n)}. \end{aligned} \tag{11}$$

Case 2. Let

$$G(w_n) = \frac{aw_n}{1 + bw_n}.$$

Then

$$G'(w_n) = \frac{a}{(1+bw_n)^2} \quad \text{and} \quad G''(w_n) = -\frac{2ab}{(1+bw_n)^3}.$$

By (8), we have

$$a = 1, \qquad b = -\frac{2m}{m-1}.$$

Thus we have another new fourth-order iteration

$$\begin{aligned} y_n &= x_n - m\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n + \frac{(m-1)mw_n}{1 - m + 2mw_n}\frac{f(x_n)}{f'(x_n)}. \end{aligned} \tag{12}$$

4 Numerical results

In this section, some numerical tests are carried out to show the performance of the new iterative family (6). All numerical computations have been carried out in a Matlab 7.0 environment using 128 digit floating point arithmetics.

The following six test functions have been used with stopping criterion $|f(x_{n+1})| \leqslant 10^{-120}$, where $x^\star$ is a root of $f(x)$ with multiplicity of m, as well as the maximum number of iteration is less than 100:

$f(x)$	$x^\star$	m
$f_1(x) = (\sin^2 x - x^2 + 1)^2$	$1.4044916482153412260350868178$	2
$f_2(x) = (x^2 - e^x - 3x + 2)^5$	$0.2575302854398607604553673049$	5
$f_3(x) = (\cos x - x)^3$	$0.7390851332151606416553120876$	3
$f_4(x) = (xe^{x^2} - \sin^2 x + 3\cos x + 5)^4$	$-1.2076478271309189270094167584$	4
$f_5(x) = (e^{x^2+7x-30} - 1)^4$	3.0	4
$f_6(x) = (\ln x + \sqrt{x} - 5)^4$	$8.3094326942315717953469556827$	4

We present the numerical test results for various fourth-order methods in Tables 1 and 2. Methods compared contain Sharma's method (2) (SM), Li's methods (3) (LM), the presented two new methods (11) (PM1) and (12) (PM2). Also, the results of the modified Newton's method (1) (NM) are given. For convenience, the average numbers of iteration steps are listed in the last line in Table 1. In Tables 1 and 2, "$-$" means that the iterative method fails to converge to the multiple root.

From Table 1, we can see that, for the given test functions and initial guesses, the performance of the new presented method PM2 is the best, which has the minimum numbers of iteration step and function evaluations. On the other hand, although the average number of iterations of NM is maximal, it requires less number of function evaluations than SM. Our presented method PM1 has the same average number of function evaluations with NM, however, the average number of iterations of PM1 is less than NM. Thus, PM1 is superior to NM. In short, among the iterative methods listed in this paper, the best one is PM2, then LM, PM1, NM. SM seems to be the worst one due to the maximum number of function evaluations, let alone failing to converge for test function f_4 from $x_0 = 2.0$.

Table 2 gives the absolute values of $f(x)$ when the stopping criterion is satisfied. It can be seen that even with less iteration steps, the presented methods PM1 and PM2 can obtain higher precision solutions. So our methods given in this paper are more suitable for high precise calculations. Since that SM and LM are members of iterative family (4), we can conclude that the presented family (6) can be competitive with the family (4), as well as the modified Newton's method (1). Of course, the performances of all these iterative methods depend not only on the initial guesses but also on the choice of parameters in families and behavior of testing function.

In applying multipoint root-finding methods, a good initial guess should be chosen very carefully. Yun [18] presented a so-called numerical integration method (NIM) to get sufficiently accurate initial guesses of iterative methods for simple roots. In [19], based on the transformation methods for finding multiple roots, a new modified non-iterative way to get initial guesses of iterative methods for multiple roots is presented. Applying this modified method to the last tested function, $f_6(x)$, defined on an interval $[3, 15]$, we can have the initial approximation $x_0 = 8.30992$ with $|x_0 - x^{\star}| \approx 4.91562 \times 10^{-4}$. However, we use $x_0 = 0.5$ and $x_0 = 10.0$ to show the convergence behavior of the mentioned methods in Tables 1 and 2.

Table 3 shows the number of iterations, function evaluations and the absolute values of $f(x)$ of different iterative methods from the initial guess $x_0 = 8.30992$. From Table 3 we know that our presented methods PM1 and PM2 only require two iteration steps to obtain the desired precision solutions while the others require three steps. From the view of the numbers of function-evaluations, SM and LM require nine function-evaluations which are more than PM1, PM2 and even more than NM. These results demonstrate that

Table 1. The number of iterations and function evaluations of various iterative methods from different initial guesses.

f	x_0	NM	SM	LM	PM1	PM2
f_1	4.5	10(20)	7(21)	7(21)	6(18)	5(15)
	2.5	9(18)	6(18)	6(18)	5(15)	5(15)
f_2	1.8	6(12)	4(12)	4(12)	4(12)	4(12)
	2.0	6(12)	4(12)	4(12)	3(9)	4(12)
f_3	1.5	6(12)	5(15)	5(15)	4(12)	4(12)
	2.5	7(14)	5(15)	5(15)	7(21)	7(21)
f_4	3.5	18(36)	17(51)	10(30)	12(36)	10(30)
	2.0	9(18)	−	5(15)	6(18)	5(15)
f_5	3.25	9(18)	6(18)	6(18)	6(18)	5(15)
	5.0	36(72)	20(60)	19(57)	24(72)	19(57)
f_6	10.0	5(10)	3(9)	3(9)	4(12)	3(9)
	0.5	8(16)	4(12)	4(12)	5(15)	4(12)
Average		10.75(21.5)	7.36(22.09)	6.5(19.5)	7.17(21.5)	**6.25**(**18.75**)

Table 2. Values of $|f(x)|$ of various iterative methods from different initial guesses when the stopping criterion satisfied.

f	x_0	NM	SM	LM	PM1	PM2
f_1	4.5	2.97E−148	3.65E−154	2.13E−157	**8.91E−165**	7.91E−144
	2.5	1.3E−144	6.56E−131	1.45E−133	3.69E−136	**5.76E−148**
f_2	1.8	2.33E−201	2.70E−169	1.59E−174	**3.77E−260**	2.5E−208
	2.0	7.64E−171	2.62E−168	**1.05E−185**	3.45E−132	7.19E−158
f_3	1.5	5.69E−143	3.93E−166	1.11E−173	**2.2E−210**	9.84E−166
	2.5	2.77E−152	1.90E−132	1.19E−137	1.85E−192	**9.88E−206**
f_4	3.5	7.25E−139	6.25E−146	1.99E−162	1.46E−192	**2.21E−232**
	2.0	2.05E−142	−	7.27E−139	**1.87E−226**	1.93E−205
f_5	3.25	5.34E−130	1.50E−174	**7.23E−186**	1.92E−138	2.4E−180
	5.0	1.78E−140	**1.61E−184**	4.33E−161	6.57E−137	2.56E−151
f_6	10.0	1.29E−138	3.37E−140	3.83E−142	4.18**E−178**	2.65E−131
	0.5	**6.72E−182**	2.77E−143	4.75E−150	8.76E−169	1.99E−136

Table 3. The number of iterations, function-evaluations and values of $|f_6(x)|$ of various iterative methods from the initial guess $x_0 = 8.30992$.

Method	NM	SM	LM	PM1	PM2
Number of iteration	3(6)	3(9)	3(9)	**2(6)**	**2(6)**
Function values	9.07E−140	2.46E−177	**4.21E−178**	1.81E−130	1.02E−130

Table 4. Values of $|f_6(x)|$.

Method	NM	SM	LM	PM1	PM2
Function values	2.55E−72	1.54E−117	1.31E−118	**1.81E−130**	**1.02E−130**

when the initial guess is sufficiently close to the exact root, PM1 and PM2 perform very well, so does NM. However, the performances of SM and LM are disappointing, although they can obtain higher precision solutions. Table 4 shows the values of function $|f_6(x)|$ for all methods after 2 iterations. We can see the presented methods PM1 and PM2 can produce higher precision solutions than others.

5 Conclusion

This paper presents a new family of optimal fourth-order iterative methods. The most difference between the presented family and other iterative methods known already is that the presented family using the modified Newton's method as its first step. The new family, requiring one evaluation of the function and two evaluations of its first derivative, is of optimal order. To our best knowledge, there is no higher order optimal iterative method than order four for multiple roots.

The numerical tests show that the two members of the new family are better than some others known already. What we should emphasize is that the performance of the new family depends not only on the test functions, initial guesses, but also on the choice of $Q(\cdot)$. However, at least, we can conclude that the presented family can compete with others.

Acknowledgment. The authors would like to thank the referees for their helpful comments and suggestions, which greatly improved the original manuscript.

References

1. E. Schröder, Über unendlich viele Algorithmen zur Auflösung der Gleichungen, *Math. Ann.,* **2**:317–365, 1870.
2. C. Dong, A basic theorem of constructing an iterative formula of the higher order for computing multiple roots of an equation, *Math. Numer. Sin.*, **11**:445–450, 1982.
3. C. Dong, A family of multiopoint iterative functions for finding multiple roots of equations, *Int. J. Comput. Math.*, **21**:363–367, 1987.

4. H.D. Victory, B. Neta, A higher order method for multiple zeros of nonlinear functions, *Int. J. Comput. Math.*, **12**:329–335, 1983.

5. B. Neta, A.N. Johnson, High-order nonlinear solver for multiple roots, *Comput. Math. Appl.*, **55**:2012–2017, 2008.

6. B. Neta, New third order nonlinear solvers for multiple roots, *Appl. Math. Comput.*, **202**:162–170, 2008.

7. B. Neta, Extension of Murakami's high order nonlinear solver to multiple roots, *Int. J. Comput. Math.*, **87**:1023–1031, 2010.

8. S. Li, X. Liao, L. Cheng, A new fourth-order iterative method for finding multiple roots of nonlinear equations, *Appl. Math. Comput.*, **215**:1288–1292, 2009.

9. S. Li, L. Cheng, B. Neta, Some fourth-order nonlinear solvers with closed formulae for multiple roots, *Comput. Math. Appl.*, **59**:126–135, 2010.

10. N. Osada, An optimal multiple root-finding method of order three, *J. Comput. Appl. Math.*, **51**:131–133, 1994.

11. J.F. Traub, *Iterative Methods for the Solution of Equations*, Prentice Hall, New Jersey, 1964.

12. C. Chun, B. Neta, A third-order modification of Newton's method for multiple roots, *Appl. Math. Comput.*, **211**:474–479, 2009.

13. C. Chun, H. Bae, B. Neta, New families of nonlinear third-order solvers for finding multiple roots, *Comput. Math. Appl.*, **57**:1574–1582, 2009.

14. B. Neta, *Numerical Methods for the Solution of Equations*, Net-A-Sof, California, 1983.

15. H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, *J. ACM*, **21**:643–651, 1974.

16. J.R. Sharma, R. Sharma, Modified Jarratt method for computing multiple roots, *Appl. Math. Comput.*, **217**:878–881, 2010.

17. X.J. Zhou, X. Chen, Y.Z. Song, Constructing higher order methods for multiple roots of nonlinear equations, *J. Comput. Appl. Math.*, **235**:4199–4206, 2011.

18. B.I. Yun, A non-iterative method for solving non-linear equations, *Appl. Math. Comput.*, **198**:691–699, 2008.

19. B.I. Yun, Transformation methods for finding multiple roots of nonlinear equations, *Appl. Math. Comput.*, **217**:599–606, 2010.