Computers and Mathematics with Applications 59 (2010) 126–135

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Some fourth-order nonlinear solvers with closed formulae for multiple roots

S.G. Li, L.Z. Cheng, B. Neta *

School of Science, National University of Defense Technology, Changsha, 410073, China

ARTICLE INFO

Article history:
Received 2 July 2009
Accepted 25 August 2009

Keywords:
Newton's method
Multiple roots
Nonlinear equations
Iterative methods
Root finding

ABSTRACT

In this paper, we present six new fourth-order methods with closed formulae for finding multiple roots of nonlinear equations. The first four of them require one-function and three-derivative evaluation per iteration. The last two require one-function and two-derivative evaluation per iteration. Several numerical examples are given to show the performance of the presented methods compared with some known methods.

Published by Elsevier Ltd

1. Introduction

Finding the roots of nonlinear equations is very important in numerical analysis and has many applications in engineering and other applied sciences. In this paper, we consider iterative methods to find a multiple root α of multiplicity m, i.e., $f^{(j)}(\alpha) = 0, j = 0, 1, \ldots, m-1$ and $f^{(m)}(\alpha) \neq 0$, of a nonlinear equation $f(x) = 0$.

The modified Newton's method for multiple roots is written as [1]

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

which is quadratically convergent.

In recent years, some modifications of Newton's method for multiple roots have been proposed and analyzed, most of which are of third-order convergence. For example, see Traub [2], Hansen and Patrick [3], Victory and Neta [4], Dong [5,6], Osada [7], Neta [8], Chun and Neta [9], Chun, Bae and Neta [10], etc. All of these methods require the knowledge of the multiplicity m.

The third-order Chebyshev's method for finding multiple roots [2,8] is given by

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

where

$$u_n = \frac{f(x_n)}{f'(x_n)}. \tag{3}$$

* Corresponding address: Naval Postgraduate School, Department of Applied Mathematics, CA 93943, Monterey, United States. Tel.: +1 (831) 656 2235.
E-mail addresses: lsg_998@yahoo.com.cn (S.G. Li), bneta@nps.edu (B. Neta).

0898-1221/$ – see front matter. Published by Elsevier Ltd
doi:10.1016/j.camwa.2009.08.066

The cubically convergent Halley's method which is a special case of the Hansen and Patrick's method [3], is written as

$$x_{n+1} = x_n - \frac{f(x_n)}{\frac{m+1}{2m}f'(x_n) - \frac{f(x_n)f''(x_n)}{2f'(x_n)}}. \tag{4}$$

The third-order Osada method [7] is written as

$$x_{n+1} = x_n - \frac{1}{2}m(m+1)u_n + \frac{1}{2}(m-1)^2\frac{f'(x_n)}{f''(x_n)}. \tag{5}$$

There are, however, not yet so many fourth- or higher-order methods known that can handle the case of multiple roots. In [11], Neta and Johnson have proposed a fourth-order method requiring one-function and three-derivative evaluation per iteration. This method is based on the Jarratt method [12] given by the iteration function

$$x_{n+1} = x_n - \frac{f(x_n)}{a_1f'(x_n) + a_2f'(y_n) + a_3f'(\eta_n)}, \tag{6}$$

where

$$\begin{cases} y_n = x_n - au_n, \\ v_n = \dfrac{f(x_n)}{f'(y_n)}, \\ \eta_n = x_n - bu_n - cv_n. \end{cases} \tag{7}$$

Neta and Johnson [11] give a table of values for the parameters a, b, c, a_1, a_2, a_3 for several values of m.

Neta [13] has developed another fourth-order method requiring one-function and three-derivative evaluation per iteration. This method is based on Murakami's method [14] given by

$$x_{n+1} = x_n - a_1u_n - a_2v_n - a_3w_3(x_n) - \psi(x_n), \tag{8}$$

where u_n is defined by (3), v_n, y_n and η_n are given by (7) and

$$\begin{aligned} w_3(x_n) &= \frac{f(x_n)}{f'(\eta_n)}, \\ \psi(x_n) &= \frac{f(x_n)}{b_1f'(x_n) + b_2f'(y_n)}. \end{aligned} \tag{9}$$

A table of values for the parameters a, b, c, a_1, a_2, a_3, b_1, b_2 for several values of m is also given by Neta [13].

In [15], a fourth-order method is proposed,

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}u_n, \\ x_{n+1} = x_n - \dfrac{\frac{1}{2}m(m-2)\left(\frac{m}{m+2}\right)^{-m}f'(y_n) - \frac{m^2}{2}f'(x_n)}{f'(x_n) - \left(\frac{m}{m+2}\right)^{-m}f'(y_n)}u_n. \end{cases} \tag{10}$$

This method requires one-function and two-derivative evaluation per iteration.

The methods proposed in [11,13] do not have closed formulae there. In this paper, by further investigating these methods in [11,13], we present six fourth-order methods with closed formulae for multiple roots of nonlinear equations. The first four of these new methods require one-function and three-derivative evaluation per iteration. The last two require one-function and two-derivative evaluation per iteration. These last ones are more efficient since they require less functional evaluations. Finally, we use some numerical examples to compare the new fourth-order methods with some known third-order methods. From the results, we can see that the fourth-order methods can be competitive to these third-order methods and usually require less functional evaluations.

2. The fourth-order methods

For simplicity, we define

$$A_j = \frac{f^{(m+j)}(\alpha)}{f^{(m)}(\alpha)}, \quad j = 1, 2, \ldots, \tag{11}$$

$$\mu = \frac{m-a}{m}. \tag{12}$$

First, we consider the method (6) proposed in [11]. Let $\alpha \in \mathbb{R}$ be a multiple root of multiplicity m of a sufficiently smooth function $f(x)$. To maximize the order of convergence to the root α, we need to find six parameters a, b, c, a_1, a_2 and a_3.

Let $e_n, \hat{e}_n, \tilde{e}_n$ be the errors at the nth step, i.e.

$$\begin{aligned} e_n &= x_n - \alpha, \\ \hat{e}_n &= y_n - \alpha, \\ \tilde{e}_n &= \eta_n - \alpha, \end{aligned} \tag{13}$$

where y_n and η_n are defined in (7).

Using the Taylor expansion of $f(x_n), f(y_n)$ and $f(\eta_n)$ about α, we have

$$f(x_n) = \frac{f^{(m)}(\alpha)}{m!} e_n^m [1 + C_1 e_n + C_2 e_n^2 + C_3 e_n^3 + C_4 e_n^4 + O(e_n^5)], \tag{14}$$

$$f'(x_n) = \frac{f^{(m)}(\alpha)}{(m-1)!} e_n^{m-1} [1 + D_1 e_n + D_2 e_n^2 + D_3 e_n^3 + D_4 e_n^4 + O(e_n^5)], \tag{15}$$

$$f'(y_n) = \frac{f^{(m)}(\alpha)}{(m-1)!} \hat{e}_n^{m-1} [1 + D_1 \hat{e}_n + D_2 \hat{e}_n^2 + D_3 \hat{e}_n^3 + D_4 \hat{e}_n^4 + O(\hat{e}_n^5)], \tag{16}$$

$$f'(\eta_n) = \frac{f^{(m)}(\alpha)}{(m-1)!} \tilde{e}_n^{m-1} [1 + D_1 \tilde{e}_n + D_2 \tilde{e}_n^2 + D_3 \tilde{e}_n^3 + D_4 \tilde{e}_n^4 + O(\tilde{e}_n^5)], \tag{17}$$

$$\begin{aligned} u_n \;=\; & \frac{e_n}{m} [1 + (C_1 - D_1) e_n + (C_2 - D_2 + D_1^2 - C_1 D_1) e_n^2 + (C_3 - D_3 \\ & + (D_1 - C_1) D_2 + (D_2 - C_2 + C_1 D_1 - D_1^2) D_1) e_n^3 + O(e_n^4)], \end{aligned} \tag{18}$$

where $C_j = \frac{m!}{(m+j)!} A_j$, $D_j = \frac{(m-1)!}{(m+j-1)!} A_j$ and u_n is defined by (3).

From (14), (15) and (18), we can get

$$\hat{e}_n = e_n - a u_n = d_0 e_n + d_1 e_n^2 + d_2 e_n^3 + d_3 e_n^4 + d_4 e^5 + O(e_n^6), \tag{19}$$

where

$$\begin{aligned} d_0 &= \mu = \frac{m-a}{m}, \qquad d_1 = -\frac{a(C_1 - D_1)}{m}, \quad d_2 = -\frac{a(C_2 - D_2 + D_1^2 - C_1 D_1)}{m}, \\ d_3 &= -\frac{a(C_3 - D_3 + (D_1 - C_1) D_2 + (D_2 - C_2 + C_1 D_1 - D_1^2) D_1)}{m}, \\ d_4 &= -\frac{a(C_4 - D_4 + (D_1 - C_1) D_3 + (D_2 - C_2 + D_1 C_1 - D_1^2) D_2 + (D_3 - C_3 + D_2 C_1 - 2 D_2 D_1 + D_1 C_2 - D_1^2 C_1 + D_1^3) D_1)}{m}. \end{aligned}$$

Substituting into (16) we have

$$f'(y_n) = \frac{f^{(m)}(\alpha)}{(m-1)!} e_n^{m-1} \Delta [1 + D_1 \hat{e}_n + D_2 \hat{e}_n^2 + D_3 \hat{e}_n^3 + D_4 \hat{e}_n^4 + O(\hat{e}_n^5)], \tag{20}$$

where

$$\begin{aligned} \Delta \;=\; & (d_0 + d_1 e_n^1 + d_2 e_n^2 + d_3 e_n^3 + d_4 e^4 + O(e_n^5))^{m-1} \\ =\; & d_0^{m-1} + (m-1) d_0^{m-2} d_1 e_n + (C_{m-1}^2 d_1^2 d_0^{m-3} + (m-1) d_2 d_0^{m-2}) e_n^2 \\ & + (2 C_{m-1}^2 d_1 d_2 d_0^{m-3} + (m-1) d_3 d_0^{m-2} + C_{m-1}^3 d_1^3 d_0^{m-4}) e_n^3 \\ & + \left((m-1) d_4 d_0^{m-2} + 2 C_{m-1}^2 d_1 d_3 d_0^{m-3} + C_{m-1}^2 d_2^2 d_0^{m-3} + C_{m-1}^4 d_1^4 d_0^{m-5} + 3 C_{m-1}^3 d_1^2 d_2 d_0^{m-4}\right) e_n^4 + O(e_n^5). \end{aligned} \tag{21}$$

In what follows, for simplicity, we let

$$a = \frac{2m}{m+2}, \tag{22}$$

then $\mu = \frac{m}{m+2}$. The error in η_n is given by

$$\tilde{e}_n = e_n - b u_n - c v_n = \bar{d}_0 e_n + \bar{d}_1 e_n^2 + \bar{d}_2 e_n^3 + \bar{d}_3 e_n^4 + \bar{d}_4 e_n^5 + O(e_n^6), \tag{23}$$

where

$$\begin{aligned} \bar{d}_0 &= \frac{m-b}{m} - \frac{c}{m+2} \mu^{-m}, \\ \bar{d}_1 &= \left[\frac{(m^2 + 2m - 4)c}{m^2 (m+2)^2 (m+1)} \mu^{-m} + \frac{b}{m^2 (m+1)} \right] A_1, \end{aligned}$$

$$
\begin{aligned}
\bar{d}_2 &= \left[-\frac{(m^4+5m^3+4m^2-8m-16)c}{m^3(m+1)^2(m+2)^3}\mu^{-m}-\frac{b}{m^3(m+1)}\right]A_1^2\\
&\quad+\left[\frac{2(m^2+2m-4)c}{m^2(m+1)(m+2)^3}\mu^{-m}+\frac{2b}{m^2(m+1)(m+2)}\right]A_2,\\
\bar{d}_3 &= \left[\frac{(3m^7+24m^6+67m^5+66m^4-64m^3-184m^2-144m-32)c}{3m^5(m+1)^3(m+2)^4}\mu^{-m}+\frac{b}{m^4(m+1)}\right]A_1^3\\
&\quad-\left[\frac{(3m^4+16m^3+20m^2-16m-64)c}{m^3(m+1)^2(m+2)^4}\mu^{-m}+\frac{(3m+4)b}{m^3(m+1)^2(m+2)}\right]A_1A_2\\
&\quad+\left[\frac{(3m^4+18m^3+28m^2-24m-48)c}{m^2(m+1)(m+2)^5(m+3)}\mu^{-m}+\frac{3b}{m^2(m+1)(m+2)(m+3)}\right]A_3,\\
\bar{d}_4 &= \left[-\frac{(3m^9+30m^8+123m^7+262m^6+250m^5-36m^4-384m^3-448m^2-288m-64)c}{3m^7(m+1)^3(m+2)^5}\mu^{-m}\right.\\
&\quad\left.-\frac{b}{m^5(m+1)}\right]A_1^4\\
&\quad+\left[\frac{2(2m^7+17m^6+57m^5+86m^4+4m^3-136m^2-160m-32)c}{m^5(m+1)^3(m+2)^5}\mu^{-m}+\frac{2(2m+3)b}{m^4(m+1)^2(m+2)}\right]A_1^2A_2\\
&\quad-\left[\frac{2(2m^6+19m^5+72m^4+116m^3-240m-192)c}{m^3(m+1)^2(m+2)^6(m+3)}\mu^{-m}+\frac{2(2m+3)b}{m^3(m+1)^2(m+2)(m+3)}\right]A_3A_1\\
&\quad-\left[\frac{2(m^4+6m^3+12m^2-32)c}{m^3(m+1)^2(m+2)^5}\mu^{-m}+\frac{2b}{m^3(m+1)^2(m+2)}\right]A_2^2\\
&\quad+\left[\frac{4(m^5+8m^4+24m^3+12m^2-32m-32)c}{m^2(m+1)(m+2)^6(m+3)(m+4)}\mu^{-m}+\frac{4b}{m^2(m+1)(m+2)(m+3)(m+4)}\right]A_4.
\end{aligned}
$$

We now expand $f'(\eta_n)$,

$$f'(\eta_n)=\frac{f^{(m)}(\alpha)}{(m-1)!}e_n^{m-1}\widetilde{\Delta}[1+D_1\tilde{e}_n+D_2\tilde{e}_n^2+D_3\tilde{e}_n^3+D_4\tilde{e}_n^4+O(\tilde{e}_n^5)], \tag{24}$$

where

$$
\begin{aligned}
\widetilde{\Delta} &= (\bar{d}_0+\bar{d}_1e_n^1+\bar{d}_2e_n^2+\bar{d}_3e_n^3+\bar{d}_4e^4+O(e_n^5))^{m-1}\\
&= \bar{d}_0^{m-1}+(m-1)\bar{d}_0^{m-2}\bar{d}_1e_n+(C_{m-1}^2\bar{d}_1^2\bar{d}_0^{m-3}+(m-1)\bar{d}_2\bar{d}_0^{m-2})e_n^2\\
&\quad+(2C_{m-1}^2\bar{d}_1\bar{d}_2\bar{d}_0^{m-3}+(m-1)\bar{d}_3\bar{d}_0^{m-2}+C_{m-1}^3\bar{d}_1^3\bar{d}_0^{m-4})e_n^3\\
&\quad+\left((m-1)\bar{d}_4\bar{d}_0^{m-2}+2C_{m-1}^2\bar{d}_1\bar{d}_3\bar{d}_0^{m-3}+C_{m-1}^2\bar{d}_2^2\bar{d}_0^{m-3}+C_{m-1}^4\bar{d}_1^4\bar{d}_0^{m-5}+3C_{m-1}^3\bar{d}_1^2\bar{d}_2\bar{d}_0^{m-4}\right)e_n^4+O(e_n^5).
\end{aligned}
\tag{25}
$$

Now substituting (14), (15), (20) and (24) into (6) and expand the quotient $\frac{f(x_n)}{a_1f'(x_n)+a_2f'(y_n)+a_3f'(\eta_n)}$ in Taylor series, we get

$$
\begin{aligned}
e_{n+1} &= e_n-\frac{f(x_n)}{a_1f'(x_n)+a_2f'(y_n)+a_3f'(\eta_n)}\\
&= K_1e_n+K_2e_n^2+K_3e_n^3+K_4e_n^4+O(e_n^5),
\end{aligned}
\tag{26}
$$

where the coefficient K_i depends on the parameters b, c, a_1, a_2, a_3 and the function $f(x)$. If we choose

$$b=a=\frac{2m}{m+2}, \tag{27}$$

we have

$$
\begin{aligned}
K_1 &= 1+\frac{c-mx}{a_2m(m+2)x^2+a_3m(m+2)xy-a_2c(m+2)x+a_1m(mx-c)},\\
K_2 &= \frac{a_2m^2(m^2+2m-4)x^3-(2ma_2(m^2+2m-4)c-m^2a_3(m^2+2m-4)y-a_1m^4)x^2}{m^2(m+1)(a_2m(m+2)x^2+a_3m(m+2)xy-a_2c(m+2)x+a_1m(mx-c))^2}\\
&\quad+\frac{(a_2(m^2+2m-4)c^2+a_3m(m^2-6m+4)yc-2a_1m^3c)x+(a_3m^2(m+1)y+a_1m^2)c^2}{m^2(m+1)(a_2m(m+2)x^2+a_3m(m+2)xy-a_2c(m+2)x+a_1m(mx-c))^2},
\end{aligned}
$$

where

$$x = \left(\frac{m}{m+2}\right)^m$$
$$y = \left(\frac{m - x^{-1}c}{m+2}\right)^m. \tag{28}$$

Before we list K_3, we choose a_1 and a_3 to annihilate the coefficients K_1 and K_2,

$$\begin{aligned} a_1 &= -\frac{2a_2(m^2-4) - m^2 - 2m + 4}{-4mx^2 + (2m^2 - 4m + 4)cx + m(m+1)c^2}x^2 \\ &\quad - \frac{ma_2(m+1)(m+2)c^2x - (m^2 - 6m + 4)cx - m(m+1)c^2}{m(-4mx^2 + (2m^2 - 4m + 4)cx + m(m+1)c^2)}, \end{aligned} \tag{29}$$

$$a_3 = \frac{4a_2m^2x^3 - m(m^2 + 8a_2c)x^2 + (2m^2 + 4a_2c)cx - mc^2}{my(-4mx^2 + (2m^2 - 4m + 4)cx + m(m+1)c^2)}, \tag{30}$$

where x, y are given by (28).

In this case, we have

$$K_3 = \varphi(a_2, c)A_1^2 + \psi(a_2, c)A_2, \tag{31}$$

where

$$\begin{aligned} \varphi(a_2, c) &= -\frac{((-8m^5 - 40m^4 - 80m^3 + 64m)a_2c + 8m^4 - 32m^2)x^3}{m^3(-4mx^2 + 2(m^2 - 2m + 2)cx + m(m+1)c^2)(c - mx)(m+1)^2(m+2)} \\ &\quad - \frac{(-4m^4 + 24m^3 - 4m^5 - 64 + 16m^2 + 160m)a_2c^2 + (-48m^2 + 2m^6 + 6m^5 + 28m^4 + 64m)c}{m^3(-4mx^2 + 2(m^2 - 2m + 2)cx + m(m+1)c^2)(c - mx)(m+1)^2(m+2)}x^2 \\ &\quad - \frac{(8m^4 + 32m^3 - 64m - 32 - 8m^2)a_2c^3 + (m^6 - m^5 - 4m^4 - 4m^3 - 40m^2 + 48m - 32)c^2}{m^3(-4mx^2 + 2(m^2 - 2m + 2)cx + m(m+1)c^2)(c - mx)(m+1)^2(m+2)}x \\ &\quad + \frac{2m^2(m^3 + 3m^2 - 2m - 4)c^3}{m^3(-4mx^2 + 2(m^2 - 2m + 2)cx + m(m+1)c^2)(c - mx)(m+1)^2(m+2)}, \end{aligned} \tag{32}$$

$$\psi(a_2, c) = -\frac{-8ma_2x^3 + (2m^2 - 4(m-2)a_2c)x^2 + (m^2 - 2m + 4a_2c)cx - mc^2}{m(-4mx^2 + (2m^2 - 4m + 4)cx^2 + m(m+1)c^2x)(m+1)(m+2)}c, \tag{33}$$

and A_1 and A_2 are defined in (11).

Let $\varphi(a_2, c) = 0$ and $\psi(a_2, c) = 0$, we can get

$$a_2 = \frac{1}{8}\frac{m^4 + 3m^3 + 10m^2 - 4m + 8}{\left(\frac{m}{m+2}\right)^m m(m+8)}, \tag{34}$$

$$c = -2x = -2\left(\frac{m}{m+2}\right)^m. \tag{35}$$

Substituting (34) and (35) into a_1 and a_3, we can get

$$a_1 = -\frac{1}{16}\frac{3m^4 + 16m^3 + 40m^2 - 176}{m(m+8)}, \tag{36}$$

$$a_3 = \frac{1}{16}\frac{m^5 + 6m^4 + 8m^3 - 16m^2 - 48m - 32}{m^2(m+8)}. \tag{37}$$

Substituting (34)–(37), (22) and (27) into (6), we will get a fourth-order method for finding multiple roots of nonlinear equations (denoted by M1),

$$\begin{cases} x_{n+1} = x_n - \dfrac{f(x_n)}{a_1f'(x_n) + a_2f'(y_n) + a_3f'(\eta_n)}, \\ y_n = x_n - \dfrac{2m}{m+2}u_n, \\ \eta_n = x_n - \dfrac{2m}{m+2}u_n + 2\left(\dfrac{m}{m+2}\right)^m v_n, \end{cases} \tag{38}$$

where a_1 is given by (36), a_2 is given by (34) and a_3 is given by (37). The error equation for M1 is

$$e_{n+1} = K_4 e_n^4 + O(e_n^5), \tag{39}$$

where

$$K_4 = \frac{1}{6}\frac{2m^8 + 25m^7 + 103m^6 + 234m^5 + 320m^4 + 328m^3 - 624m^2 + 64m - 128}{m^6(m+1)^3(m+2)^2(m+8)}A_1^3 - \frac{m^4 + 7m^3 - 2m^2 + 4m + 8}{m^4(m+2)(m+1)^2(m+8)}A_1A_2 + \frac{m}{(m+1)(m+2)^3(m+3)}A_3. \tag{40}$$

If we choose

$$b = 0 \tag{41}$$

then we have

$$K_1 = 1 + \frac{(m+2)x - c}{(mc - m(m+2)x)a_1 - a_2(m^2+4m+4)x^2 + (a_2c - a_3ym)(m+2)x},$$

$$K_2 = -\frac{a_2(m^2+2m-4)(m+2)^2x^3 + (a_1+a_3y)m^2(m+2)^2x^2 - 2a_2(m+2)(m^2+2m-4)cx^2}{m^2(m+1)(a_1mc - (m+2)((m+2)a_2x^2 + m(a_1+a_3y) - a_2c)x)^2} - \frac{(m^2+2m-4)a_2c^2x - ma_3(3m^2+10m-4)cyx - 2m^2a_1(m+2)cx + a_3m^2(m+1)yc^2 + a_1m^2c^2}{m^2(m+1)(a_1mc - (m+2)((m+2)a_2x^2 + m(a_1+a_3y) - a_2c)x)^2},$$

where

$$x = \left(\frac{m}{m+2}\right)^m, \qquad y = \left(\frac{m+2-x^{-1}c}{m+2}\right)^m. \tag{42}$$

Let $K_1 = 0$ and $K_2 = 0$, then we can get

$$a_1 = -\frac{1}{4}\frac{(4ma_3y + m^2 + 2m - 4)(m+2)^2x^2 - 2(m^2a_3(m+4)y + m^2 + 2m - 4)(m+2)cx}{m(mx + 2x - c)^2} + \frac{1}{4}\frac{a_3m^2(m+1)(m+2)y + m^2 + 2m - 4}{m(mx+2x-c)^2}c^2, \tag{43}$$

$$a_2 = \frac{m}{4}\frac{(m+2)^2x^2 - 2((m^2+4m-2)a_3y + m + 2)cx + (1 + ma_3(m+1)y)c^2}{x((m+2)^2x^2 - (2m+4)cx + c^2)}, \tag{44}$$

where x, y are defined in (42).

Substituting a_1 and a_2 into K_3, we get

$$K_3 = \varphi(a_3, c)A_1^2 + \psi(a_3, c)A_2, \tag{45}$$

where

$$\varphi(a_3, c) = \frac{-2(m-2)(m+2)^4x^3 + 2(a_3(m^5 + 7m^4 + 12m^3 - 12m^2 - 48m + 16)y + 3(m-2)(m+2)^2)(m+2)cx^2}{(c - mx - 2x)((m+2)^2x^2 - (2m+4)cx + c^2)m^3(m+1)^2(m+2)} - \frac{ma_3(m^5 + 13m^4 + 42m^3 - 120m - 32)y + 6(m-2)(m+2)^2}{(c - mx - 2x)((m+2)^2x^2 - (2m+4)cx + c^2)m^3(m+1)^2(m+2)}c^2x + \frac{2a_3m(m+1)(m^3 + 3m^2 - 4m - 4)c^3y + (2m^2 - 8)c^3}{(c - mx - 2x)((m+2)^2x^2 - (2m+4)cx + c^2)m^3(m+1)^2(m+2)}, \tag{46}$$

$$\psi(a_3, c) = \frac{(c - 2x)ya_3c}{x(c - mx - 2x)(m+1)(m+2)}, \tag{47}$$

and A_1 and A_2 are defined in (11) and x, y are defined in (42).

Let $\varphi(a_3, c) = 0$ and $\psi(a_3, c) = 0$, we can get

$$a_3 = \frac{1}{16}\frac{m^3(m^2-4)}{\left(\frac{m}{m+2}\right)^m(m^3 + 2m^2 - 8m + 4)}, \tag{48}$$

$$c = 2\left(\frac{m}{m+2}\right)^m. \tag{49}$$

Substituting (48) and (49) into (43) and (44), we can get

$$a_1 = \frac{1}{8}\frac{m^6 - m^5 - 14m^4 + 12m^3 + 48m^2 - 80m + 32}{m(m^3 + 2m^2 - 8m + 4)}, \tag{50}$$

$$a_2 = -\frac{m}{16}\frac{3m^4 - 6m^3 - 20m^2 + 40m - 16}{\left(\frac{m}{m+2}\right)^m (m^3 + 2m^2 - 8m + 4)}. \tag{51}$$

Substituting (48)–(51), (22) and (41) into (6), we will get another fourth-order method (denoted by M2),

$$\begin{cases} x_{n+1} = x_n - \dfrac{f(x_n)}{a_1 f'(x_n) + a_2 f'(y_n) + a_3 f'(\eta_n)}, \\ y_n = x_n - \dfrac{2m}{m+2}u_n, \\ \eta_n = x_n - 2\left(\dfrac{m}{m+2}\right)^m v_n, \end{cases} \tag{52}$$

where a_1, a_2 and a_3 are given by (50), (51) and (48) respectively. The error equation for M2 is

$$e_{n+1} = K_4 e_n^4 + O(e_n^5), \tag{53}$$

where

$$K_4 = \frac{1}{6}\frac{2m^9 + 9m^8 + 9m^7 - 28m^6 - 124m^5 + 328m^4 + 144m^3 - 1248m^2 + 1280m - 384}{m^6(m+1)^3(m+2)(m^3 + 2m^2 - 8m + 4)}A_1^3 - \frac{m^3 + m^2 - 6m + 4}{m(m+2)(m+1)^2(m^3 + 2m^2 - 8m + 4)}A_1A_2 + \frac{m}{(m+1)(m+2)^3(m+3)}A_3. \tag{54}$$

Now, we consider the method (8) proposed in [13]. There are eight parameters in (8). It is difficult to discuss this method directly with eight parameters. By considering the following method (dropping b_1 and b_2) which is simpler, we get two fourth-order methods which require one-function and three-derivative evaluation per iteration, i.e.

$$\begin{cases} x_{n+1} = x_n - a_1u_n - a_2v_n - a_3\dfrac{f(x_n)}{f'(\eta_n)}, \\ y_n = x_n - au_n, \\ \eta_n = x_n - bu_n - cv_n. \end{cases} \tag{55}$$

There are six parameters a, b, c, a_1, a_2 and a_3 to be determined. The computing process is the same as that of getting the fourth-order methods (38) and (52) before. Substituting (14), (15), (20) and (24) into the error equation

$$e_{n+1} = e_n - a_1u_n - a_2v_n - a_3\frac{f(x_n)}{f'(\eta_n)}, \tag{56}$$

and expand the quotients in Taylor series, we get

$$e_{n+1} = K_1^1 e_n + K_2^1A_1e_n^2 + (K_3^1A_1^2 + K_3^2A_2)e_n^3 + (K_4^1A_1^3 + K_4^2A_1A_2 + K_4^3A_3)e_n^4 + O(e_n^5), \tag{57}$$

where the coefficient K_i^j depends on the parameters a, b, c, a_1, a_2 and a_3. Let the coefficients of e_n, e_n^2, e_n^3 be zero, i.e. $K_1^1 = 0$, $K_2^1 = 0$, $K_3^1 = 0$ and $K_3^2 = 0$. Then, we can determine the parameters and get a fourth-order method.

If $a = b$ given by (27), we can get a fourth-order method based on (58) (denoted by M3),

$$\begin{cases} x_{n+1} = x_n - a_1u_n - a_2v_n - a_3\dfrac{f(x_n)}{f'(\eta_n)}, \\ y_n = x_n - \dfrac{2m}{m+2}u_n, \\ \eta_n = x_n - \dfrac{2m}{m+2}u_n + 2\left(\dfrac{m}{m+2}\right)^m v_n, \end{cases} \tag{58}$$

where

$$a_1 = \frac{m}{8}\frac{m^4 + 4m^3 - 8m + 48}{m^2 + 2m + 6}, \tag{59}$$

$$a_2 = \frac{1}{4}\frac{\left(\frac{m}{m+2}\right)^m m(m^3 + 12m^2 + 36m + 32)}{m^2 + 2m + 6} \tag{60}$$

$$a_3 = -\frac{1}{8}\frac{m^2(m^3 + 6m^2 + 12m + 8)}{m^2 + 2m + 6}, \tag{61}$$

and which satisfies the error equation

$$e_{n+1} = K_4 e_n^4 + O(e_n^5), \tag{62}$$

where

$$K_4 = \frac{1}{3}\frac{m^8 + 10m^7 + 51m^6 + 175m^5 + 412m^4 + 612m^3 + 488m^2 + 112m - 160}{m^4(m+1)^3(m+2)^3(m^2+2m+6)}A_1^3 - \frac{m^3 + 2m^2 + 8m + 4}{m^2(m+1)^2(m+2)(m^2+2m+6)}A_1A_2 + \frac{m}{(m+1)(m+2)^3(m+3)}A_3.$$

If a is given by (22) and b is given by (41), we can get another fourth-order method based on (55) (denoted by M4),

$$\begin{cases} x_{n+1} = x_n - a_1u_n - a_2v_n - a_3\dfrac{f(x_n)}{f'(\eta_n)}, \\ y_n = x_n - \dfrac{2m}{m+2}u_n, \\ \eta_n = x_n - 2\left(\dfrac{m}{m+2}\right)^m v_n, \end{cases} \tag{63}$$

where

$$a_1 = -\frac{1}{4}\frac{m(2m^4 - m^3 - 12m^2 + 20m - 8)}{m^2 - 4m + 2}, \tag{64}$$

$$a_2 = \frac{1}{8}\frac{\left(\frac{m}{m+2}\right)^m m(5m^4 + 10m^3 - 16m^2 - 24m + 16)}{m^2 - 4m + 2}, \tag{65}$$

$$a_3 = -\frac{1}{8}\frac{m^3(m+2)^2\left(\frac{m}{m+2}\right)^m}{m^2 - 4m + 2}, \tag{66}$$

and which satisfies the error equation

$$e_{n+1} = K_4 e_n^4 + O(e_n^5), \tag{67}$$

where

$$K_4 = \frac{1}{3}\frac{m^8 + 2m^7 - 5m^6 - 15m^5 + 14m^4 - 100m^3 + 88m^2 + 32m - 32}{m^5(m+1)^3(m+2)^2(m^2-4m+2)}A_1^3 - \frac{m^3 - 2m^2 - 4m + 4}{m(m+1)^2(m+2)^2(m^2-4m+2)}A_1A_2 + \frac{m}{(m+1)(m+2)^3(m+3)}A_3.$$

Furthermore, by considering the following iteration function having $c = a_1 = a_2 = 0$ in (8), we get two fourth-order methods which require one-function and two-derivative evaluation per iteration, i.e.

$$\begin{cases} x_{n+1} = x_n - a_3\dfrac{f(x_n)}{f'(\eta_n)} - \dfrac{f(x_n)}{b_1f'(x_n) + b_2f'(y_n)}, \\ y_n = x_n - au_n, \\ \eta_n = x_n - bu_n. \end{cases} \tag{68}$$

There are five parameters a, b, a_3, b_1, b_2. For simplicity, we let either $b = a$ or $b = 0$. Then, $f'(\eta_n)$ reduces to $f'(y_n)$ and $f'(x_n)$, respectively.

When $b = a$, we can get a fourth-order method based on (69) (denoted by M5),

$$\begin{cases} x_{n+1} = x_n - a_3\dfrac{f(x_n)}{f'(y_n)} - \dfrac{f(x_n)}{b_1f'(x_n) + b_2f'(y_n)}, \\ y_n = x_n - \dfrac{2m}{m+2}u_n, \end{cases} \tag{69}$$

where

$$a_3 = -\frac{1}{2}\frac{\left(\frac{m}{m+2}\right)^m m(m^4 + 4m^3 - 16m - 16)}{m^3 - 4m + 8}, \tag{70}$$

$$b_1 = -\frac{(m^3 - 4m + 8)^2}{m(m^4 + 4m^3 - 4m^2 - 16m + 16)(m^2 + 2m - 4)}, \tag{71}$$

$$b_2 = \frac{m^2(m^3 - 4m + 8)}{\left(\frac{m}{m+2}\right)^m (m^4 + 4m^3 - 4m^2 - 16m + 16)(m^2 + 2m - 4)} \tag{72}$$

and $a = b$ given by (27). This method satisfies the error equation

$$e_{n+1} = K_4 e_n^4 + O(e_n^5), \tag{73}$$

where

$$K_4 = \frac{1}{3}\frac{m^6 + 6m^5 + 10m^4 - 2m^3 - 24m^2 + 8m - 32}{m^4(m+1)^3(m+2)(m^2+2m-4)}A_1^3 - \frac{A_1A_2}{m(m+2)(m+1)^2} + \frac{m}{(m+1)(m+2)^3(m+3)}A_3. \tag{74}$$

When $b = 0$, we can get another fourth-order method based on (75) (denoted by M6),

$$\begin{cases} x_{n+1} = x_n - a_3\dfrac{f(x_n)}{f'(x_n)} - \dfrac{f(x_n)}{b_1 f'(x_n) + b_2 f'(y_n)}, \\ y_n = x_n - \dfrac{2m}{m+2}u_n, \end{cases} \tag{75}$$

where

$$a = \frac{2m}{m+2}, \qquad a_3 = -\frac{1}{2}m^2 + m, \qquad b_1 = -\frac{1}{m}, \qquad b_2 = \frac{1}{m\left(\frac{m}{m+2}\right)^m}. \tag{76}$$

This method satisfies the error equation

$$e_{n+1} = K_4 e_n^4 + O(e_n^5), \tag{77}$$

where

$$K_4 = \frac{1}{3}\frac{m^3 + 2m^2 + 2m - 2}{m^4(m+1)^3}A_1^3 - \frac{A_1A_2}{m(m+2)(m+1)^2} + \frac{m}{(m+1)(m+2)^3(m+3)}A_3.$$

After some computations, it is easy to see that (75) can reduce to the method (10) proposed in [15]. So, (M6) and (10) are equivalent.

In this section, we have proposed six fourth-order methods with closed formulae. The first four methods denoted (M1)–(M4) require one-function and three-derivative evaluation per iteration. The last two methods, denoted (M5) and (M6), require one-function and two-derivative evaluation per iteration. (M6) is the equivalent to the method (10). We have tried to get a fifth-order method based on (8). But, the equations we get are very complex and we could not yet find such methods.

3. Numerical example

In this section, we employ the presented six methods (38), (52), (58), (63), (69) and (75) to solve some nonlinear equations and compare them with the modified Newton's method (1) (NM), Chebyshev's method (2) (CM), Halley's method (4) (HM) and Osada's method (5) (OM). All computations were done using the MAPLE using 1024 digit floating point arithmetics (Digits := 1024). We use the following functions, which have also been considered in [9,10], respectively.

$f(x)$	m	x_*
$f_1(x) = (x^3 + 4x^2 - 10)^3$	3	1.365230013414096845760806829 0
$f_2(x) = (\sin^2 x - x^2 + 1)^2$	2	1.404491648215341226035086817 8
$f_3(x) = (x^2 - e^x - 3x + 2)^5$	5	0.257530285439860760455367304 9
$f_4(x) = (\cos x - x)^3$	3	0.739085133215160641655312087 6
$f_5(x) = (xe^{x^2} - \sin^2 x + 3\cos x + 5)^4$	4	−1.207647827130918927009416758 4
$f_6(x) = (e^{x^2+7x-30} - 1)^4$	4	3.0
$f_7(x) = (e^x + x - 20)^2$	2	2.842438953784447067816585940 2

Table 1 shows the number of iterations required such that $|f(x)| < 10^{-60}$ and the number of function evaluations (NFEs) after required iterations in parentheses. From the results displayed in Table 1, we can see that, for the functions we tested, the fourth-order methods proposed in this paper can be competitive to the known third-order methods and Newton's method and usually converge faster and require less function evaluations. Methods (69) and (75) require three function evaluations per iteration, which is the same as the third-order methods compared, while (69) and (75) require less iterations to converge to the root α.

Acknowledgment

The second author's work was partially supported by the National Natural Science Foundation of China (60573027 and 60621003).

Table 1
The number of iterations and NFEs.

$f(x)$	x_0	NM	CM	HM	OM	M1	M2	M3	M4	M5	M6
f_1	−0.4	167(334)	7(21)	92(276)	12(36)	5(20)	4(16)	30(120)	9(36)	9(27)	7(21)
	3.0	7(14)	5(15)	4(12)	5(15)	4(16)	4(16)	4(16)	3(12)	4(12)	4(12)
f_2	2.7	7(14)	5(15)	5(15)	5(15)	4(16)	4(16)	4(16)	4(16)	4(12)	4(12)
	3.5	7(14)	5(15)	5(15)	5(15)	4(16)	4(16)	4(16)	4(16)	4(12)	4(12)
f_3	1.8	5(10)	3(9)	4(12)	3(9)	3(12)	3(12)	3(12)	3(12)	3(9)	3(9)
	3.0	6(12)	4(12)	4(12)	4(12)	4(16)	3(12)	3(12)	4(16)	3(9)	3(9)
f_4	1.7	5(10)	4(12)	4(12)	4(12)	3(12)	3(12)	3(12)	3(12)	3(9)	3(9)
	2.2	5(10)	4(12)	4(12)	4(12)	3(12)	3(12)	3(12)	3(12)	3(9)	3(9)
f_5	−3.5	17(34)	11(33)	9(27)	12(36)	10(40)	8(32)	9(36)	12(48)	9(27)	9(27)
	−2.5	11(22)	7(21)	6(18)	8(24)	6(24)	5(20)	6(24)	7(28)	6(18)	6(18)
f_6	3.2	8(16)	5(15)	4(12)	5(15)	4(16)	4(16)	4(16)	5(20)	4(12)	4(12)
	4.2	22(44)	15(45)	12(36)	16(48)	13(52)	11(44)	12(48)	16(64)	12(36)	11(33)
f_7	3.0	5(10)	4(12)	3(9)	4(12)	3(12)	3(12)	3(12)	3(12)	3(9)	3(9)
	5.5	9(18)	6(18)	5(15)	6(18)	4(16)	4(16)	5(20)	4(16)	4(12)	4(12)

References

[1] E. Schröder, Über unendlich viele Algorithmen zur Auflösung der Gleichungen, Math. Ann. 2 (1870) 317–365.
[2] J.F. Traub, Iterative methods for the solution of equations, Chelsea Publishing Company, New York, 1977.
[3] E. Hansen, M. Patrick, A family of root finding methods, Numer. Math. 27 (1977) 257–269.
[4] H.D. Victory, B. Neta, A higher order method for multiple zeros of nonlinear functions, Int. J. Comput. Math. 12 (1983) 329–335.
[5] 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 (1982) 445–450.
[6] C. Dong, A family of multipoint iterative functions for finding multiple roots of equations, Int. J. Comput. Math. 21 (1987) 363–367.
[7] N. Osada, An optimal multiple root-finding method of order three, J. Comput. Appl. Math. 51 (1994) 131–133.
[8] B. Neta, New third order nonlinear solvers for multiple roots, Appl. Math. Comput. 202 (2008) 162–170.
[9] C. Chun, B. Neta, A third-order modification of Newton's method for multiple roots, Appl. Math. Comput. 211 (2009) 474–479.
[10] C. Chun, H.J. Bae, B. Neta, New families of nonlinear third-order solvers for finding multiple roots, Comput. Math. Appl. 57 (2009) 1574–1582.
[11] B. Neta, A.N. Johnson, High order nonlinear solver for multiple roots, Comput. Math. Appl. 55 (2008) 2012–2017.
[12] P. Jarratt, Multipoint iterative methods for solving certain equations, Comput. J. 8 (1966) 398–400.
[13] B. Neta, Extension of Murakami's High order nonlinear solver to multiple roots, Int. J. Comput. Math., in press (doi:10.1080/00207160802272263).
[14] T. Murakami, Some fifth order multipoint iterative formulae for solving equations, J. Inform. Process. 1 (1978) 138–139.
[15] S.G. Li, L.Z. Cheng, A new fourth-order iterative method for finding multiple roots of nonlinear equations, Appl. Math. Comput. 215 (2009) 1288–1292.