THREE-STEP ITERATIVE METHOD WITH FIFTH-ORDER CONVERGENCE FOR SOLVING MULTIPLE ROOTS OF NONLINEAR EQUATIONS

Zhinan Wu
School of Mathematics and Computer Science, Yichun University, Yichun 336000,PR China

ABSTRACT
This paper presents a three-step iterative method with fifth-order iterative convergence as a new modification of Newton's method for finding multiple roots of nonlinear equations with unknown multiplicity m and for finding system of nonlinear equations. Its order of convergence is analyzed and proved. Results for some numerical examples show the efficiency of the new method.

Keywords: *Nonlinear equation; Multiple roots; Newton-like method; High-order convergence; Iterative methods.*

1. Introduction

This paper addresses the problem of finding multiple roots x^* of nonlinear equation $f(x)=0$, where $f:[a,b]\subset R\to R$ is a nonlinear differential function on $[a,b]$. In case the multiplicity m is given explicitly, there are many iterative methods established via various techniques(see[1-15] for more details). If the multiplicity m is not known explicitly, Traub [16] utilized a simple transformation $F(x)=f(x)/f'(x)$ instead of $f(x)$ for computing a multiple root of $f(x)=0$. In this case, the aim of solving a multiple root is reduced to that of solving a simple root of the transformed equation $f(x)=0$, and thus any iterative method can be used to preserve the original convergence order. However, Newton's method for this transformed equation requires evaluations of the derivatives $f'(x)$ and $f''(x)$. In order to avoid the evaluations of these derivatives with the multiplicity m unknown, for multiple roots, King[17] proposed the secant method which does not use the function $F=f/f'$, but rather use $F=\dfrac{f(x)}{\dfrac{f(x-f(x))-f(x)}{(x-f(x))-x}}=\dfrac{-f^2(x)}{f(x-f(x))-f(x)}$. Wu and Fu [18] further used $F(x)=\dfrac{f^2(x)}{f(x)-f(x-f(x))}$ and transformed the problem of solving multiple roots of $f(x)=0$ into that of solving simple root of $f(x)=0$. Actually, they established the following iteration formulae:

$$x_{n+1}=x_n-\frac{F^2(x_n)}{p\cdot F^2(x_n)+F(x_n)-F(x_n-F(x_n))}, \tag{1}$$

where $p\in R$, $|p|<\infty$. So, the sequence $\{x_n\}$ produced by the iteration formulae (1) is at least quadratically convergent for multiple roots. Moreover, Wu et al.[19] defined a function

$$F(x)=\frac{sign(f(x))f(x)|f(x)|^{1/m}}{sign(f(x+sign(f(x))|f(x)|^{1/m})-f(x))f(x)|f(x)|^{1/m}+f(x+sign(f(x))|f(x)|^{1/m})-f(x)}, \tag{2}$$

where m is the multiplicity, and employed the modified Steffensen's method (see[20],[21])

$$\begin{aligned} x_{n+1} &= x_n - h_n \frac{F^2(x_n)}{t\cdot F^2(x_n) + F(x_n + F(x_n)) - F(x_n)} \\ &= x_n - h_n \frac{F(x_n)}{t\cdot F(x_n) + (F(x_n + F(x_n)) - F(x_n))/(F(x_n))} \end{aligned} \tag{3}$$

to compute the approximate solution of the equation $f(x)=0$, where $h_n(>0)$ is the step size of iteration and $|t|<\infty$. Parida and Gupta [22] suggested another transformation

$$F(x) = \begin{cases} \dfrac{f^2(x)}{\delta + f(x+f(x)) - f(x)} & \text{iff(x)} \neq 0, \\ 0 & \text{iff(x)} = 0. \end{cases} \tag{4}$$

where $\delta = sign(f(x+f(x)) - f(x))f^2(x)$, and transform the task of solving multiple zeros of f into that of solving simple zero of F. In this case, they utilized a quadratically convergent derivative free Newton-like iterative method:

$$x_{n+1} = x_n - \frac{F^2(x_n)}{p\cdot F^2(x_n) + F(x_n) - F(x_n - F(x_n))} \tag{5}$$

where the parameter p should be chosen such that the denominator is the largest in magnitude. Yun [23] suggested a new transformation of $f(x)$ as

$$H_\varepsilon(x) = \frac{\varepsilon f^2(x)}{f(x+\varepsilon f(x)) - f(x)}, \tag{6}$$

took ε such that $\max_{a\le x\le b}|\varepsilon f(x)| = \delta$, that is

$$\varepsilon = \frac{\delta}{\max_{a\le x\le b}|f(x)|} = \frac{\delta}{\max\{|f(a)|,|f(b)|\}},$$

and proposed an iterative method as follows:

$$x_{n+1} = x_n - \frac{2(x_n - x_{n-1})H_\varepsilon(x_n)}{H_\varepsilon(2x_n - x_{n-1}) - H_\varepsilon(x_{n-1})}. \tag{7}$$

Recently, for the transformed equation $K(x)=0$ with a simple root, Yun[24] proposed a Steffensen-type iterative formula

$$p_{k+1} = p_k - \frac{\mu K(p_k)^2}{K(p_k + \mu K(p_k)) - K(p_k)}, \quad k \geq 0, \tag{8}$$

where $K(x) = K(\varepsilon; x) = \{$ f(x) 0 ,0, if f(x) = 0 ..

In this paper we construct a new modification of Newton's method based on Newton's method. We will present the proof that the method is three-step iterative method with fifth-order convergence for nonlinear equations of multiple roots with unknown multiplicity m and without requiring the use of the second derivative.

2. Iterative method with fifth-order convergence for solving multiple roots

We consider the simple transformation (see[16],[25]):

$$F(x) = \begin{cases} \dfrac{f(x)}{f'(x)}, & \text{iff(x)} \neq 0, \\ 0, & \text{iff(x)} = 0, \end{cases} \tag{9}$$

and use a Newton-like iterative method:

$$\begin{cases} y_n = x_n - \dfrac{F(x_n)}{F'(x_n)}, \\ z_n = y_n - \dfrac{F(y_n)}{F'(y_n)}, \\ x_{n+1} = z_n - \dfrac{F(z_n)}{F'(z_n)}. \end{cases} \tag{10}$$

In order to avoid computing the first derivatives of function $F(x_n)$, $F(y_n)$ and $F(z_n)$, we approximate them as follows:

$$F'(x_n) \approx \frac{F(x_n + F(x_n)) - F(x_n)}{F(x_n)} = g_1(x_n), \tag{11}$$

$$F'(y_n) \approx \frac{2(F(y_n) - F(x_n))}{y_n - x_n} - g_1(x_n) = g_2(x_n) \tag{12}$$

$$\begin{aligned} F'(z_n) \approx \quad & F[z_n, y_n] + F[z_n, x_n, x_n](z_n - y_n) \\ = \quad & \frac{F(z_n) - F(y_n)}{z_n - y_n} + \frac{\dfrac{F(z_n) - F(x_n)}{z_n - x_n} - g_1(x_n)}{z_n - x_n}(z_n - y_n) \\ = \quad & g_3(x_n) \end{aligned} \tag{13}$$

(See [25-28] for the detail discussions of (11-13) respectively.) Substituting the approximations of $F'(x_n)$, $F'(y_n)$ and $F'(z_n)$ given by (11)-(13) in (10), we establish the following new iterative method:

$$\begin{cases} y_n = x_n - \dfrac{F(x_n)}{g_1(x_n)}, \\ z_n = y_n - \dfrac{F(y_n)}{g_2(x_n)}, \\ x_{n+1} = z_n - \dfrac{F(z_n)}{g_3(x_n)}. \end{cases} \tag{14}$$

We give the following convergence theorem for the proposed method (14) as follows.

Theorem 1 Suppose that $F \in C^1(D)$ ($D \subseteq R \to R$) has a single root $x^* \in D$, where D is an open interval. If the initial point x_0 is sufficiently close to x^*, the iterative method defined by (14) has fifth-order convergence.

Proof: We assume that $f(x)$ can be written as

$$f(x) = (x - x^*)^m h(x), \tag{15}$$

where x^* is a multiple root of (15) with multiplicity m, $h(x)$ is a continuous function with $h(x^*) \neq 0$. According to (15), we have

$$f'(x) = m(x - x^*)^{m-1} h(x) + (x - x^*)^m h'(x). \tag{16}$$

Dividing (15) by (16), we get

$$F(x)=\frac{f(x)}{f^{'}(x)}=\frac{(x-x^{*})h(x)}{mh(x)+(x-x^{*})h'(x)}. \tag{17}$$

From (17), we can see that the problem of computing multiple roots of $f(x)=0$ can be reduced to the equivalent problem of computing simple root x^{*} of $F(x)=0$.

Using Taylor's expansion, we have

$$h(x_n)=h(x^{*})[1+b_1e_n+b_2e_n^2+b_3e_n^3+b_4e_n^4+b_5e_n^5+b_6e_n^6+o(e_n^7)] \tag{18}$$

where $b_k=\dfrac{h^{(k)}(x^{*})}{k!h(x^{*})}, k=1,2,...,$ and $e_n=x_n-x^{*}$.

By (18), we obtain

$$h'(x_n)=h(x^{*})[b_1+2b_2e_n+3b_3e_n^2+4b_4e_n^3+5b_5e_n^4+6b_6e_n^5+o(e_n^6)]. \tag{19}$$

Substituting (18)-(19) into (17), we get

$$F(x_n)=\frac{e_nh(x_n)}{mh(x_n)+e_nh'(x_n)}=c_1e_n+c_2e_n^2+c_3e_n^3+c_4e_n^4+c_5e_n^5+c_6e_n^6+o(e_n^7) \tag{20}$$

where

$$c_1=\frac{1}{m},\quad c_2=-\frac{b_1}{m^2},\quad c_3=\frac{-2b_2m+b_1^2m+b_1^2}{m^3} \tag{21}$$

$$c_4=\frac{3b_1b_2m^2+4b_1b_2m-3b_3m^2-b_1^3m^2-2b_1^3m-b_1^3}{m^4} \tag{22}$$

$$\begin{aligned}c_5=\ &\frac{1}{m^5}(6b_1b_3m^2+4b_1b_3m^3-4b_4m^3+2b_2^2m^3+4b_2^2m^2-4b_2b_1^2m^3\\&-10b_2b_1^2m^2-6b_2b_1^2m+b_1^4m^3+3b_1^4m^2+3b_1^4m+b_1^4)\end{aligned} \tag{23}$$

$$\begin{aligned}c_6=\ &\frac{1}{m^6}(8b_1b_4m^3-b_1^5m^4-4b_1^5m^3-6b_1^5m^2-4b_1^5m-b_1^5+5b_1b_4m^4-5b_5m^4\\&+12b_2b_3m^3+5b_2b_3m^4-14b_1^2b_3m^3-5b_1^2b_3m^4-9b_1^2b_3m^2-5b_1b_2^2m^4\\&-16b_1b_2^2m^3-12b_1b_2^2m^2+5b_1^3b_2m^4+18b_1^3b_2m^3+21b_1^3b_2m^2+8b_1^3b_2m)\end{aligned} \tag{24}$$

Substituting (20) into (11), we obtain

$$\begin{aligned}g_1(x_n)=\ &c_1+c_2(2+c_1)e_n+(3c_3+3c_1c_3+c_1^2c_3+c_2^2)e_n^2+(4c_2c_3+2c_1c_2c_3+4c_4\\&+6c_1c_4+4c_1^2c_4+c_1^3c_4)e_n^3+(5c_5+10c_1c_5+5c_1^3c_5+10c_1^2c_5+2c_3^2c_1+c_2^2c_3\\&+7c_2c_4+c_1^4c_5+3c_3^2+3c_1^2c_2c_4+8c_1c_2c_4)e_n^4+o(e_n^5)\end{aligned} \tag{25}$$

Substituting (20) and (25) into the first formula of (14), we have

$$
\begin{aligned}
y_n = \quad & x^* + \frac{c_2(1+c_1)}{c_1} e_n^2 + \frac{1}{c_1^2}(-2c_2^2 + 2c_1c_3 + 3c_1^2c_3 + c_1^3c_3 - 2c_1c_2^2 - c_1^2c_2^2)e_n^3 \\
& + \frac{1}{c_1^3}(3c_1^2c_4 + 6c_1^3c_4 + 4c_1^4c_4 + c_1^5c_4 + 5c_1c_2^3 + 3c_1^2c_2^3 + c_1^3c_2^3 + 4c_2^3 - 10c_1^2c_2c_3 \\
& - 7c_1c_2c_3 - 7c_1^3c_2c_3 - 2c_1^4c_2c_3)e_n^4 + o(e_n^5)
\end{aligned} \tag{26}
$$

With (26), we get

$$
\begin{aligned}
F(y_n) = \quad & c_2(1+c_1)e_n^2 + \frac{1}{c_1}(-2c_2^2 + 2c_1c_3 + 3c_1^2c_3 + c_1^3c_3 - 2c_1c_2^2 - c_1^2c_2^2)e_n^3 \\
& + \frac{1}{c_1^2}(3c_1^2c_4 + 6c_1^3c_4 + 4c_1^4c_4 + c_1^5c_4 + 7c_1c_2^3 + 4c_1^2c_2^3 + c_1^3c_2^3 + 5c_2^3 - 10c_1^2c_2c_3 \\
& - 7c_1c_2c_3 - 7c_1^3c_2c_3 - 2c_1^4c_2c_3)e_n^4 + o(e_n^5)
\end{aligned} \tag{27}
$$

Thereby, with (20), (25-27), we obtain

$$
\begin{aligned}
g_2(x_n) = \quad & c_1 - c_1c_2e_n - \frac{1}{c_1}(c_1c_3 + 3c_1^2c_3 + c_1^3c_3 - c_1c_2^2 - 2c_2^2)e_n^2 \\
& - \frac{1}{c_1^2}(2c_1^2c_4 + 6c_1^3c_4 + 4c_1^4c_4 + c_1^5c_4 + 4c_1c_2^3 + 2c_1^2c_2^3 + 4c_2^3 - 4c_1^2c_2c_3 - 6c_1c_2c_3)e_n^3 \\
& - \frac{1}{c_1^3}(10c_1^4c_5 - 3c_1^3c_3^2 + 3c_1^3c_5 - 4c_1^2c_3^2 - 8c_2^4 + c_2c_4c_1^5 + 20c_2^2c_1^2c_3 + 15c_3c_1^3c_2^2 \\
& + 16c_1c_2^2c_3 + 4c_1^4c_2^2c_3 + 10c_5c_1^5 + 5c_5c_1^6 + c_5c_1^7 - 10c_1c_2^4 - 6c_1^2c_2^4 - 2c_2^4c_1^3 \\
& - 7c_2c_4c_1^3 - 8c_2c_4c_1^2)e_n^4 + o(e_n^5)
\end{aligned}
$$

(28)

From (26-28), it follows that

$$
\begin{aligned}
z_n = \quad & x^* - \frac{c_2^2(1+c_1)}{c_1} e_n^3 - \frac{c_2}{c_1^3}(-3c_2^2c_1 - 2c_2^2c_1^2 - c_2^2 + 6c_3c_1^2 + c_3c_1 + 7c_3c_1^3 + 2c_1^4c_3)e_n^4 \\
& - \frac{1}{c_1^4}(9c_3^2c_1^3 + 2c_3^2c_1^2 + 4c_2^2 + 16c_1^4c_2c_4 + 2c_1^6c_2c_4 + 9c_2c_4c_1^5 - 21c_1^2c_2^2c_3 - 10c_1^3c_2^2c_3 - 8c_1c_2^2c_3 \\
& + 7c_1c_2^4 + 3c_1^2c_2^4 + 2c_1^3c_2^4 + 11c_2c_4c_1^3 + 2c_2c_4c_1^2 + 12c_1^4c_3^2 + 6c_1^5c_3^2 + c_1^6c_3^2 + c_1^4c_2^4)e_n^5 + o(e_n^6)
\end{aligned} \tag{29}
$$

It is similar to (27), we have

$$
\begin{aligned}
F(z_n) = \quad & -c_2^2(1+c_1)e_n^3 - \frac{c_2}{c_1^2}(-3c_2^2c_1 - c_2^2c_1^2 - c_2^2 + 6c_1^2c_3 + c_3c_1 + 7c_3c_1^3 + 2c_3c_1^4)e_n^4 \\
& + o(e_n^5)
\end{aligned} \tag{30}
$$

Moreover, substituting (20), (25), (26), (27), (29), (30) into (13), we get

$$
g_3(x_n) = c_1 - c_2^2(1+c_1)e_n^2 - \frac{c_2}{c_1}(2c_3c_1^3 + 7c_3c_1^2 + 7c_3c_1 + 2c_2^2c_1 + c_2^2 + 2c_3)e_n^e + o(e_n^4) \tag{31}
$$

Substituting (29-31) into the third formula of (14), we get

$$x_{n+1} = z_n - \frac{F(z_n)}{g_3(x_n)} = x^* + \frac{(1+2c_1+c_1^2)c_2^4}{c_1^2} e_n^5 + o(e_n^6) \quad (32)$$

which just means that the iterative method defined by (14) has fifth-order convergence. The proof is completed.

We further consider how to find the multiplicity of the root x^* in the iterative method. If x_n is the nth iteration computed by an iterative method applied to f, then from (9), we have

$$f_n \approx \frac{(x_n - x^*)h(x_n)}{mh(x_n) + (x_n - x^*)h'(x_n)} = \frac{\varepsilon_n h(x_n)}{mh(x_n) + \varepsilon_n h'(x_n)},$$

where $f_n = f(x_n)$. Because ε_n is small, we get $f_n \approx \frac{\varepsilon_n}{m}$. Similarly, we can compute that $f_{n+1} \approx \frac{\varepsilon_{n+1}}{m}$. Furthermore $\varepsilon_{n+1} - \varepsilon_n = x_{n+1} - x_n$. Consequently, when the iteration becomes closer to the root x^*, we can estimate its multiplicity by computing

$$m \approx \frac{x_{n+1} - x_n}{f_{n+1} - f_n}.$$

In the practical computing root x^* process, some iteration number is no more than two by using (14). According to this case, we compute the root x^* by using (14), then we select the initial value near the root x^*, and we use Newton iterative method to compute the multiplicity. From Table 1, we can know that Newton iterative method's the iteration number is enough to ensure computing the correct multiplicity. Therefore, m is approximately the reciprocal of the divided difference of f for successive iteration x_n and x_{n+1}. It may be computed, and displayed at each step along with the current iteration(see[17],[22]).

3. Numerical results

We employ the proposed modification of Newton's method with three-step (14)(MNM) to solve some nonlinear equations and compare them with King's method [17] (KM, (4),(13)), the high-order convergence iteration methods without employing derivatives given in [18](WFM,(6)), the improved method for finding multiple roots and it¡¯s multiplicity of nonlinear equations given in [22](PGM,((6),(11))), the derivative free iterative method for finding multiple roots of nonlinear equations given in [23](YM,((7),(10))), transformation methods for finding multiple roots of nonlinear equations [24](YM,((10),(12))), as well as Newton's first order method(NM). Displayed in Table 1 are the number of iterations required such that $| f(x_n)| < 1.E-17$, $| x_n - x^* | < 1.E-17$. All the computations were done by using Visual C++ 6.0. Since King's method and Yun's method require two starting values, we have used $x_1 = x_0 - 0.1$. We used the following test functions and obtained the approximate zeros x^* round up to the 17th decimal place:

$$f_1(x) = \frac{(x-\sqrt{5})^4}{(x-1)^2+1}, \quad m = 4, \quad x^* = 2.23606797\,7499790$$

$$f_2(x) = (\sin^2(x) - 2x + 1)^5, \quad m = 5, \quad x^* = 0.71483582\,544138924$$

$$f_3(x) = (8xe^{-x^2} - 2x - 3)^8, \quad m = 8, \quad x^* = -1.79035317\,91589544$$

$$f_4(x) = \frac{(2x\cos(x) + x^2 - 3)^{10}}{(x^2+1)}, \quad m = 10, \quad x^* = 2.98064527\,94385368$$

$$f_5(x) = (e^{-x^2+x+3} - x + 2)^9, \quad m = 9, \quad x^* = 2.49053982\,76083051$$

$$f_6(x) = (e^{-x} + 2\sin(x))^4, \quad m = 4, \quad x^* = 3.16274887\,09263654$$

$$f_7(x) = (\ln(x^2 + 3x + 5) - 2x + 7)^8, \quad m = 8, \quad x^* = 5.46901233\,59101421$$

$$f_8(x) = (\sqrt{x^2 + 2x + 5} - 2\sin(x) - x^2 + 3)^5, \quad m = 5, \quad x^* = 2.33196765\,58839640$$

$$f_9(x) = (x-2)^4/((x-1)^2 + 1), \quad m = 4, \quad x^* = 2.00000000\,00000000$$

$$f_{10}(x) = (x-2.5)^{\frac{15}{4}} e^x, \quad m = \frac{15}{4}, \quad x^* = 2.50000000\,0000000$$

$$f_{11}(x) = (\sqrt{x} - \frac{1}{x} - 1)^7, \quad m = 7, \quad x^* = 2.14789903\,5704787$$

$$f_{12}(x) = (\ln(x) + \sqrt{x} - 5)^3, \quad m = 3, \quad x^* = 8.30943269\,4231572$$

$$f_{13}(x) = (\sin(x)\cdot\cos(x) - x^3 + 1)^9, \quad m = 9, \quad x^* = 1.11707877\,0687451$$

$$f_{14}(x) = ((x-3)\exp(x))^5, \quad m = 5, \quad x^* = 3.00000000\,00000000$$

$$f_{15}(x) = (\ln(x) + \sqrt{x^4 + 1} - 2)^7, \quad m = 7, \quad x^* = 1.22281396\,3628973$$

Note that we used NC in Table 1 to mean that the method does not convergence to the root. And these methods can converge to root by using closer initial values. The computational results in Table 1 demonstrate that our proposed iterative method (MNM) requires less number of iterations than those of KM, WFM, PGM, YM, and NM. Therefore, it is significant and applicable and can compete with other existing methods. Table 1 Comparison of various iterative methods

$f(x)$	MNM	KM[17]	WFM[18]	PGM[22]	YM[23]	YM[24]	NM
Parameters			$p = 1$	$p = 1$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-8}$	
$f_1, x_0 = 4.30$	2	6	7	6	NC	NC	122
$f_2, x_0 = 4.50$	4	NC	NC	NC	NC	NC	168
$f_3, x_0 = 5.00$	2	NC	NC	NC	NC	NC	400
$f_4, x_0 = 4.90$	3	NC	NC	NC	NC	NC	339
$f_5, x_0 = 6.50$	3	NC	NC	NC	NC	NC	290
$f_6, x_0 = 3.80$	2	NC	NC	3	NC	NC	119
$f_7, x_0 = 8.30$	2	NC	NC	NC	NC	NC	257
$f_8, x_0 = 5.50$	2	NC	NC	NC	NC	NC	164
$f_9, x_0 = 3.50$	2	5	6	5	4	4	121
$f_{10}, x_0 = 3.10$	2	6	4	20	NC	3	86
$f_{11}, x_0 = 4.50$	2	4	NC	NC	NC	2	224
$f_{12}, x_0 = 12.0$	2	4	9	8	NC	NC	87

$f_{13}, x_0 = 2.20$	3	NC	NC	NC	NC	NC	300
$f_{14}, x_0 = 7.00$	3	NC	NC	NC	NC	NC	182
$f_{15}, x_0 = 6.00$	3	NC	NC	NC	NC	NC	244

4. Efficiency of iterative methods

In the following we compare the efficiency of methods mentioned in Section 1. We consider the definition of efficiency index [29-31] as EFF $= r^{\frac{1}{\theta}}$, where r is the order of the method and θ is number of function (and derivatives) evaluations per iteration required by the method. These results are presented in table 2. In table 2, we listed the methods according to decreasing order of efficiency index, and multiplicity m of roots with all methods under the row of King's method (including King's method) are unknown. For unknown multiplicity m, our new method comes second, next to King's method. Table 2 Comparison the efficiency of various iterative methods

Method	Reference	r	θ	EFF
Neta	[8](49) $m \neq 3$	2.732	2	1.653
Neta	[8](51)	2.732	2	1.653
Neta and Johnson	[10] $m = 2$	4	3	1.587
Neta	[8](39)	3	3	1.442
Neta	[8](29) $m \neq 3$	3	3	1.442
Chun and Neta	[7](22)	3	3	1.442
Chun, Bae, and Neta	[9](14)	3	3	1.442
Victory and Neta	[4](3)	3	3	1.442
Hansen and Patrick	[2](8.1)	3	3	1.442
Halley	[12]	3	3	1.442
Laguerre	[13]	3	3	1.442
Dong	[14](7),(8)	3	3	1.442
Dong	[5](9),(10)	3	3	1.442
Osada	[6]	3	3	1.442
E.Schröder	[1]	2	2	1.414
Neta and Johnson	[10] $m \neq 2$	4	4	1.414
Neta	[11]	4	4	1.414
Neta	[8](32) $m = 3$	2	3	1.259
Werner	[15](16) $m = 2$	1.5	3	1.145
King	[17]	1.618	2	1.272
Our method	This paper	5	8	1.223
Xinyuan Wu	[18]	2	4	1.190

Xinyuan Wu	[19]	2	4	1.190
P.K.Parida	[22]	2	4	1.190
Beong In Yun	[23]	2	4	1.190
Beong In Yun	[24]	2	4	1.190

5. Conclusion

A new iterative method with fifth-order convergence has been developed as a modification of Newton's method for finding multiple roots of nonlinear equations with unknown multiplicity m and for finding system of nonlinear equations. Several numerical examples demonstrate that the proposed iterative method is more efficient and performs better than classical Newton's method and many other existing methods.

References

[1]. E.Schröder, Über unendlich viele Algorithmen zur Auflösung der Gleichungen, Math.Ann. 2(1870)317-365.
[2]. E. Hansen, M. Patrick, A family of root finding methods, Numer. Math. 27(1977) 257-269.
[3]. S.G.Li, L.Z.Cheng, B. Neta, Some fourth-order nonlinear solvers with closed formulae for Multiple roots, Computers and Mathematics with Applications 59(2010)126-135.
[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 family of multipoint iterative functions for finding multiple roots of equations, Int. J. Comput. Math. 21(1987)363-367.
[6]. N. Osada, An optimal multiple root-finding method of order three, J. Comput. Appl. Math. 51(1994)131-133.
[7]. C. Chun, B. Neta, A third-order modification of Newton¡¯s method for multiple roots, Applied Mathematics and Computation 211(2009)474-479.

[8]. B. Neta, New third order nonlinear solvers for multiple roots, Applied Mathematics and Computation 202(2008) 162-170.

[9]. C. Chun, H.J. Bae, B. Neta, New families of nonlinear third-order solvers for finding multiple roots,Computers and Mathematics with Applications 57(2009) 1574-1582.

[10]. B. Neta, A.N. Johnson, High-order nonlinear solver for multiple roots, Computers and Mathematics with Applications 55(2008)2012-2017.

[11]. B. Neta, Extension of Murakami's High order nonlinear solver to multiple roots, Int. J. Comput. Math. 87(2010)1023-1031.

[12]. E. Halley, A new, exact and easy method of finding the roots of equations generally and that without any previous reduction, Phil. Trans. Roy. Soc. London 18(1694) 136-148.

[13]. E.N. Laguerre, Sur une méthode pour obtener par approximation les racines d'une équation algébrique qui a toutes ses racines réelles, Nouvelles Ann. de Math. 2e séries 19(1880)88-103.

[14]. C. Dong, A basic theorem of constructing an iterative formula of the higher order for computing multiple roots of an equation, Math. Numer. Sinica 11(1982)445-450.

[15]. W. Werner, Iterationsverfahren höherer Ordnung zur Lösung nicht linearer Gleichungen, Z. Angew. Math. Mech., 61(1981)T322-T324.

[16]. J.F. Traub, Iterative Methods for the Solution of Equations, Prentice Hall, Englewood, 1964.

[17]. R.F. King, A secant method for multiple roots, BIT 17(1977) 321-328.

[18]. X.Y. Wu, D.S. Fu, New higher-order convergence iteration methods without employing derivatives for solving nonlinear equations, Comput. Math. Appl. 41(2001) 489-495.

[19]. X.Y. Wu, J.L. Xia, R. Shao, Quadratically convergent multiple roots finding method without derivatives, Comput. Math. Appl. 42(2001) 115-119.

[20]. I.F. Steffensen, Remark on Iteration, Volume 16, pp. 64-72, Skand, Aktuarietidskr, (1933).

[21]. X.Y. Wu, A new continuation Newton-like method and its deformation, Applied Mathematics and Computation 112(2000) 75-78.

[22]. P.K. Parida, D.K. Gupta, An improved method for finding multiple roots and it's multiplicity of nonlinear equations in R, Appl. Math. Comput. 202(2008) 498-503.

[23]. B.I. Yun, A derivative free iterative method for finding multiple roots of nonlinear equations, Appl. Math. Lett. 22(2009) 1859-1863.

[24]. B.I. Yun, Transformation methods for finding multiple roots of nonlinear equations, Applied Mathematics and Computation 217 (2010)599-606.

[25]. J.B. Kioustelidis, A derivative-free transformation preserving the order of convergence of iteration methods in case of multiple zeros, Numerische Mathematik 33(1979) 385-389.

[26]. X. Wang and L. Liu, Modified Ostrowski's method with eighth-order convergence and high efficiency index, Applied Mathematics Letters, 23(2010), pp.549-554.

[27]. W. Bi, H. Ren, Q. Wu, New family of seventh-order methods for nonlinear equations, Applied Mathematics and Computation 203(2008) 408-412.

[28]. W. Bi, H. Ren, Q. Wu, Three-step iterative methods with eighth-order convergence for solving nonlinear equations, Journal of Computational and Applied Mathematics 225(2009) 105-112.

[29]. W. Gautschi, Numerical Analysis: an Introduction, Birkh $\ddot{a}$ user, 1997.

[30]. B. Neta, On a family of multipoint methods for non-linear equations, International Journal of Computer Mathematics, 9(1981)353-361.

[31]. X. Li, C. Mu, J. Ma, C. Wang, Sixteenth-order method for nonlinear equations, Applied Mathematics and Computation 215 (2010)3754-3758.