Filomat 37:26 (2023), 8995–9008
https://doi.org/10.2298/FIL2326995D

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Root-multiplicity and root iterative refinement

Jovana Džunić[a]

[a]*Faculty of Electronic Engineering, University of Niš, Serbia*

Abstract. We discuss root-finding algorithms for multiple zeros of nonlinear equations in one variable. Recent investigations regarding this problem were mainly aimed at deriving schemes that use the beforehand knowledge of root multiplicity. In this communication we investigate several such root-finding methods under the assumption that the multiplicity of the sought root is not early known. We analyze strategies where root refinement is calculated along side to its multiplicity assessment, and put them to use through numerical experiments. Presented results go in favor of a more realistic use of the analysed methods.

Let $f : [a, b] \subset \mathbb{R} \mapsto \mathbb{R}$ be a real valued function, and the task is to approximate a solution α to the equation $f(x) = 0$, isolated in the interval $[a, b]$. When the solution α is a simple zero of the function f, iterative refinement algorithms are very well investigated (see for example [2], [15], [16], [17] and references therein). In the construction of root-finding algorithms Kung-Traub conjecture ([9]) has steered investigations in previous decades.

Kung-Traub conjecture: Multipoint iterative methods without memory, costing $n + 1$ function evaluations per iteration, have order of convergence at most 2^n.

In the case of multiple roots, for the sake of the optimal convergence order, constructed root-finding methods assume a priori the knowledge of root's multiplicity m, (see for example [1], [5], [6], [12], [13], [19], [21], [22], [26], [27] and references therein). When solving real-life problems, beforehand root multiplicity is seldom known. For this reason we investigate behaviour of the existing two-point optimal methods for multiple zeros when root multiplicity is yet to be determined. In such case both approximation to the root α and its multiplicity m have to be simultaneously improved through iterations ([8], [23], [25]). Following the analysis presented in [19] we restrict our attention to non root-ratio optimal methods.

Note that in the absence of m value there are other ways to approximate only the sought root α. One approach is to transform the function f to relieve α of its multiplicity ([19], [28], [29]). However, we do not consider it in this communication.

The paper is organized as follows: in the first section we introduce a list of basic notions and statements regarding iterative methods and root multiplicity. In the second section strategies for multiplicity assessment are analyzed. The third section deals with a general family of two-point iterative methods presented in [30]. A hybrid scheme is compiled and analyzed. Particular methods of such hybrid scheme are put to work on several test functions. Numerical results are presented in the fourth section.

2020 *Mathematics Subject Classification*. Primary 65H05
Keywords. Nonlinear equations, Multiple roots, Multipoint methods, Convergence order.
Received: 27 June 2022; Revised: 01 December 2022; Accepted: 08 May 2023
Communicated by Marko Petković
This research was supported by the Serbian Ministry of Education, Science and Technological Development under grant 451-03-68/2022-14/ 200102
ORCID number 0000-0003-3609-8399
Email address: jovana.dzunic@elfak.ni.ac.rs (Jovana Džunić)

1. Preliminaries

Throughout the paper we will use $f[x,y]$ to denote the first order divided difference

$$f[x,y]=\begin{cases}\dfrac{f(y)-f(x)}{y-x}, & x\neq y,\\ f'(x), & x=y.\end{cases}$$

Next, we introduce the relevant terminology.

Definition 1.1. *Let a function f be sufficiently differentiable. Number α that satisfies $f(\alpha)=0$ is the zero of the function f, or the root of an equation $f(x)=0$. Zero α is of multiplicity $m\in\mathbb{N}$ if*

$$f(\alpha)=f'(\alpha)=\cdots=f^{(m-1)}(\alpha)=0,\qquad f^{(m)}(\alpha)\neq 0.$$

For $m=1$, the zero α is called simple.

Definition 1.2. *For a sequence $\{x_k\}\subset\mathbb{R}$, generated by an iterative formula*

$$x_{k+1}=\varphi(x_k,x_{k-1},\ldots,x_{k-m}),\quad m\geq 0, \tag{1}$$

that converges to α, we say that it is of order of convergence $r>0$ if

$$\Delta_{k+1}=|x_{k+1}-\alpha|=O(\Delta_k)^r=O(|x_k-\alpha|)^r,\quad k\to\infty.$$

In other words, for the sequence $\{x_k\}$ there exists a constant $C_r\geq 0$ such that for k large enough the following holds

$$\Delta_{k+1}\leq C_r(\Delta_k)^r.$$

When $r\leq 1$ for the convergence $\lim\limits_{k\to\infty}x_k=\alpha$ it is necessary to have $C_r<1$. Specially, when the following is valid

$$\Delta_{k+1}\leq c_k\Delta_k,\qquad c_k\to 0,\ k\to\infty,$$

we say that convergence is superlinear.

Theorem 1.3 ([16]). *Let the sequence $\{x_k\}$ generated by an iterative procedure* (1) *be convergent to α. If there exists a constant $\gamma>0$, and some non-negative integers s_i, $0\leq i\leq m$, such that the inequality holds*

$$\Delta_k\leq\gamma\prod_{i=0}^{m}(\Delta_{k-i})^{s_i},$$

then the order of convergence r of (1) *satisfies inequality*

$$r\geq s^*,$$

where s^* is the unique positive root of

$$t^{m+1}-\sum_{i=0}^{m}s_it^{m-i}=0.$$

Having recalled the basics regarding root iterative refinement, we now turn our attention to root-multiplicity assessment. Results that follow were proven in [28].

Corollary 1.4. *For $m+1$ times continuously differentiable function f with zero α of multiplicity m there exists a unique function $g_0(x)$ such that*

$$f(x) = (x-\alpha)^m g_0(x), \qquad \lim_{x\to\alpha} g_0(x) \neq 0,$$

and $g_0'(x)$ is bounded in some neighbourhood of α.

Lemma 1.5. *Let α be the zero of $f(x)$ of multiplicity $m \geq 1$, and define function*

$$D(x) = f[x+\gamma_1 f(x), x-\gamma_2 f(x)], \qquad \gamma_1, \gamma_2 \geq 0.$$

Then, α is the simple zero of $u(x) = \dfrac{f(x)}{D(x)}$, where

$$u(\alpha) = \frac{f(\alpha)}{f'(\alpha)} = \lim_{x\to\alpha} \frac{f(x)}{D(x)}.$$

What is more,

$$\lim_{x\to\alpha} u'(x) = \frac{1}{m} \tag{2}$$

and $u''(x)$ exists and is bounded in some neighbourhood of α.

From (2) of lemma 1.5 we conclude that a good approximation of $u'(\alpha) = \lim_{x\to\alpha} u'(x)$ is required in order to obtain a good estimate of the multiplicity m.

Note that for $\gamma_1 = \gamma_2 = 0$, function $u(x) = \dfrac{f(x)}{f'(x)}$ is the Newton's correction and is used in the modified Newton's method for multiple zeros [20]

$$x_{k+1} = x_k - m\frac{f(x_k)}{f'(x_k)}\,.$$

When $\gamma_2 = 0$ and $\gamma_1 \neq 0$ we have $u(x) = \dfrac{f(x)}{f[x+\gamma_1 f(x), f(x)]}$, thus it represents a correction in the Traub-Steffensen's modified method for multiple zeros ([4], [7], [11], [29], etc.),

$$x_{k+1} = x_k - m\frac{f(x_k)}{f[x_k+\gamma f(x_k), x_k]}\,.$$

We investigate optimal non root-ratio methods presented in [10], [21] and their generalization from [30]. These two-point optimal methods obtain order of convergence 4 when the multiplicity m is known. The Zhou et al. family of methods is of the form

$$\begin{cases} y_k = x_k - t\,u_k, & u_k = \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = x_k - Q(v_k)u_k, & v_k = \dfrac{f'(y_k)}{f'(x_k)}, \end{cases} \tag{3}$$

where $t \in \mathbb{R}$ is a parameter, and properties of the weight function $Q(v)$ are described through its Taylor series expansion. Optimal value of t, and particular forms of the weight function were suggested in [30] and specific derivative values of Q were managed through more parameter values.

In [30] an error relation for (3) was obtained[1)]

$$\varepsilon_{k+1} = \left(1 - \frac{Q(c)}{m}\right)\varepsilon_k + O(\varepsilon_k)^2, \qquad \varepsilon_s = x_s - \alpha, \tag{4}$$

[1)]In [30] an error relation was given with coefficients of higher order terms, up to ε_k^4.

where $c = \left(1 - \frac{t}{m}\right)^{m-1}$. Obviously, when $Q(c) \to m,\ k \to \infty$ the method is of superlinear convergence. When $Q(c) = m$ method (3) is at least of second order.

From the complete error relation of the method (3) it was concluded in [30] that the coefficient t and the weight function $Q(v)$ have to satisfy the following conditions in order to obtain the fourth order of convergence,

$$
\begin{aligned}
& t = \frac{2m}{2+m}, \quad c = \left(\frac{m}{2+m}\right)^{m-1}, \\
& Q(c) = m, \quad Q'(c) = -\frac{1}{4} m^{3-m}(2+m)^m, \\
& Q''(c) = \frac{1}{4} m^4 \left(\frac{m}{2+m}\right)^{-2m}.
\end{aligned}
$$

Parameters contained in the definition of $Q(v)$ have high dependence on the multiplicity m. Thus, when m is not known, the weight function $Q(v)$ needs to be updated through iterations to obtain better convergence speed. This is done by improving $m-$value approximations. For this reason we first discuss $m-$approximation.

2. Strategies for multiplicity assessment

For the multiplicity m assessment, from Lemma 1.5 we consider Newton's correction $u(x) = \frac{f(x)}{f'(x)}$. Therefore,

$$u'(\alpha) \approx u'(x_k) \approx u[x_k, x_{k-1}] = \frac{u_k - u_{k-1}}{x_k - x_{k-1}}.$$

The estimate of multiplicity m can be obtained in each iteration as proposed in [7], here only using the rounding:

$$m_k = \text{Round}\left|\frac{x_k - x_{k-1}}{u_k - u_{k-1}}\right|, \tag{5}$$

or, as in [23] only using the rounding,

$$
\begin{cases}
\mu_k = \dfrac{\mu_{k-1}}{\left|1 - \dfrac{u_k}{u_{k-1}}\right|}, \\
m_k = \text{Round}\,(\mu_k).
\end{cases} \tag{6}
$$

Note that an initial approximation $m_0 = 1$ makes a good choice based on (4). We now analyze approximation formulas (5) and (6). We use the rounding function for $m-$strategies with mainly polynomials in mind. The work presented in [18] can be thus modified for the simultaneous approximation of all polynomial zeros and their multiplicity based on the analysis conducted here. When the function f contains radicals, definition 1.1 may not be applicable to all zeros of f, the rounding in (5) and (6) should be omitted. This observation is supported through numerical experiments.

Theorem 2.1. *Let $\{x_k\}$ be a sequence that converges to α which is the zero of f of multiplicity m. Then for m_k defined in* (5) *the following is valid*

$$m - m_k \leq O(x_{k-1} - \alpha).$$

Proof. According to Corollary 1.4, we can write $f(x) = (x-\alpha)^m g_0(x)$, leading to

$$\begin{aligned} f'(x) &= m(x-\alpha)^{m-1} g_0(x) + (x-\alpha)^m g_0'(x) \\ &= (x-\alpha)^{m-1}\Big(m\, g_0(x) + (x-\alpha) g_0'(x)\Big) \\ &= (x-\alpha)^{m-1} g_1(x), \qquad g_1(x) = m\, g_0(x)\left(1 + (x-\alpha)\frac{g_0'(x)}{m\, g_0(x)}\right). \end{aligned}$$

Let us denote the error $x - \alpha = \varepsilon$. Then,

$$u(x) = \frac{f(x)}{f'(x)} = \frac{x-\alpha}{m\left(1 + (x-\alpha)\dfrac{g_0'(x)}{m g_0(x)}\right)} = \frac{\varepsilon}{m}\Big(1 + O(\varepsilon)\Big) = \frac{\varepsilon}{m} + O(\varepsilon)^2.$$

Introduce the iteration subscripts: $\varepsilon_k = x_k - \alpha$ and $u_k = \dfrac{f(x_k)}{f'(x_k)}$. We assume $\varepsilon_k = O(\varepsilon_{k-1})^r,\ r \geq 1$, for large enough $k \in \mathbb{N}$.

$$u_k = \frac{\varepsilon_k}{m}\Big(1 + O(\varepsilon_k)\Big) = \frac{\varepsilon_k}{m} + O(\varepsilon_k)^2, \tag{7}$$

$$u_{k-1} = \frac{\varepsilon_{k-1}}{m}\Big(1 + O(\varepsilon_{k-1})\Big) = \frac{\varepsilon_{k-1}}{m} + O(\varepsilon_{k-1})^2, \tag{8}$$

$$u_k - u_{k-1} = \frac{\varepsilon_k - \varepsilon_{k-1}}{m} + O(\varepsilon_{k-1})^2 = \frac{\varepsilon_k - \varepsilon_{k-1}}{m}\Big(1 + O(\varepsilon_{k-1})\Big),$$

$$\frac{x_k - x_{k-1}}{u_k - u_{k-1}} = \frac{\varepsilon_k - \varepsilon_{k-1}}{\dfrac{\varepsilon_k - \varepsilon_{k-1}}{m}\Big(1 + O(\varepsilon_{k-1})\Big)} = m\Big(1 + O(\varepsilon_{k-1})\Big),$$

$$m_k = \text{Round}\left|\frac{x_k - x_{k-1}}{u_k - u_{k-1}}\right| = \text{Round}\left|m\Big(1 + O(\varepsilon_{k-1})\Big)\right|. \tag{9}$$

For k large enough, that is for $|mO(\varepsilon_{k-1})| = |O(\varepsilon_{k-1})| < \dfrac{1}{2}$, based on (9) the formula (5) gives the exact multiplicity value m. Otherwise, when $m_k \neq m$ we have $m_k = m + O(1)$ due to rounding. □

We note from (4) and (9) that the m approximation (5) is enhanced by the improvement in α approximation and vice versa.

It is of interest to analyze the formula (6), as well.

Theorem 2.2. *Let $\{x_k\}$ be a sequence that converges to α which is the zero of f of multiplicity m. Then for m_k defined in* (6) *the following is valid*

$$m - m_k \leq O(1).$$

Proof. Assume again $\varepsilon_k = O(\varepsilon_{k-1})^r,\ r \geq 1$, for large enough $k \in \mathbb{N}$. Let us first observe that when $m_{k-1} \neq m$, due to the rounding process we have

$$m - m_{k-1} = O(1). \tag{10}$$

If m_k is calculated with (6) then, based on (7) and (8),

$$\begin{aligned} \frac{u_k}{u_{k-1}} &= \frac{\varepsilon_k}{\varepsilon_{k-1}}\Big(1 + O(\varepsilon_{k-1})\Big) = \varepsilon_{k-1}^{r-1}\Big(1 + O(\varepsilon_{k-1})\Big), \\ \mu_k &= \frac{\mu_{k-1}}{\left|1 - \dfrac{u_k}{u_{k-1}}\right|} = \frac{\mu_{k-1}}{\left|1 - \varepsilon_{k-1}^{r-1}\Big(1 + O(\varepsilon_{k-1})\Big)\right|} \\ &= \mu_{k-1}\left|1 + \varepsilon_{k-1}^{r-1}\Big(1 + O(\varepsilon_{k-1})\Big)\right| = \mu_{k-1}\left|1 + O(\varepsilon_{k-1})^{r-1}\right|, \\ m - \mu_k &= m - \mu_{k-1} + O(\varepsilon_{k-1})^{r-1}. \end{aligned}$$

After the rounding process

$$m_k = \text{Round}(\mu_k) = m + \text{Round}\Big(O(1) + O(\varepsilon_{k-1})^{r-1}\Big), \tag{11}$$

a very rough estimate is obtained. □

We can conclude that once a high quality approximation of α is obtained it may not influence much the quality of m approximation through formula (6). We expect for this reason that (5) proves itself as the better choice for the root-finding procedures. This assumption will be tested through numerical examples.

3. Compiling the root-finding methods

The family of methods (3) is here combined with the $m-$strategies (5) and (6) to produce hybrid schemes where approximations of m and α are produced through iterations. For some initial value x_0 and $m_0 = 1$, the transformed iterative schemes read

$$\begin{cases} u_k = \dfrac{f(x_k)}{f'(x_k)}, & m_k = \text{Round}\left|\dfrac{x_k - x_{k-1}}{u_k - u_{k-1}}\right|, \ k \geq 1, \\ y_k = x_k - 2t_k u_k, & t_k = \dfrac{m_k}{m_k+2}, \qquad v_k = \dfrac{f'(y_k)}{f'(x_k)}, \\ x_{k+1} = x_k - Q(v_k)u_k, \end{cases} \tag{12}$$

and

$$\begin{cases} u_k = \dfrac{f(x_k)}{f'(x_k)}, & \begin{cases} \mu_k = \dfrac{\mu_{k-1}}{\left|1 - \dfrac{u_k}{u_{k-1}}\right|}, \\ m_k = \text{Round}\,(\mu_k)\,. \end{cases}, \ k \geq 1, \\ y_k = x_k - 2t_k u_k, & t_k = \dfrac{m_k}{m_k+2}, \qquad v_k = \dfrac{f'(y_k)}{f'(x_k)}, \\ x_{k+1} = x_k - Q(v_k)u_k, \end{cases} \tag{13}$$

where the weight function Q satisfies conditions

$$c = \left(1 - \frac{2t_k}{m}\right)^{m-1}, \qquad Q(c) = m_k,$$

$$Q'(c) = -\frac{1}{4}m_k^3 t_k^{-m_k}, \qquad Q''(c) = \frac{1}{4}m_k^4 t_k^{-2m_k}.$$

Different particular methods of (12) and (13) are obtained for particular choices of the weight function Q. Obviously, once $m_k = m$ is obtained in either of iterative schemes (12) and (13) order of convergence will be 4, as proven in [30]. It is of relevance to estimate convergence speed before $m_k = m$ occurs.

The error relation (4) suggests that when $m_k \neq m$ we have $m_k = m + O(1)$. Thus, steps in the first few iterations of schemes (12) and (13) are of linear convergence. However, since we are dealing with multipoint iterations, we can expect more then linear convergence of these methods quite soon.

In the case when rounding is not applied in $m-$strategy, we use error relations from the previous section

$$\frac{x_k - x_{k-1}}{u_k - u_{k-1}} = m + O(\varepsilon_{k-1})\,, \qquad \mu_k = \mu_{k-1}\Big|1 + O(\varepsilon_{k-1})^{r-1}\Big|$$

Using Theorem 1.3 and error relation (4), leads to $\varepsilon_{k+1} \sim \varepsilon_k \varepsilon_{k-1}$. Thus, we can conclude that order of convergence of the method with the first $m-$strategy is not smaller then the positive root of the equation

$$t^2 - t - 1 = 0, \qquad s^* = \frac{1+\sqrt{5}}{2} \approx 1.62.$$

For the other $m-$strategy:

$$
\begin{aligned}
\mu_k &= \mu_{k-1}\left|1+O(\varepsilon_{k-1})^{r-1}\right| \\
&= \mu_{k-2}\left|1+O(\varepsilon_{k-1})^{r-1}\right|\left|1+O(\varepsilon_{k-2})^{r-1}\right| \\
&= \mu_0\prod_{i=1}^{k}\left|1+O(\varepsilon_{k-i})^{r-1}\right| \\
&= m(1+\delta)\prod_{i=1}^{k}\left|1+O(\varepsilon_{k-i})^{r-1}\right| \\
\varepsilon_{k+1} &\sim \varepsilon_k\delta.
\end{aligned}
$$

For this reason we can only guarantee linear convergence with the second $m-$strategy. However, for both strategies, it is likely that order 4 is not achieved when rounding isn't applied.

4. Numerical experiments

Algorithms investigated have been implemented in Wolfram Mathematica language for its ability to deliver results in arithmetic of arbitrary precision. This is very convenient when testing high order methods. We used transformed iterative schemes (12) and (13) with particular choices of the weight function. Methods were tested with and without rounding the values of m_k. Also, results for methods (3) with the exact $m-$values were included.

For the tested methods of schemes (12) and (13) we will just give particular form of the weight function $Q(v)$.

1. Sharma et al. [21]

$$
(M1):\begin{cases}
Q(v) = A_k + \dfrac{B_k}{v} + \dfrac{C_k}{v^2}, \\
A_k = \dfrac{m_k}{8}\left(m_k^3 - 4m_k + 8\right), \\
B_k = -\dfrac{m_k}{4}(m_k-1)(m_k+2)^2 t_k^m \\
C_k = \dfrac{m_k}{8}(m_k+2)^3 t_k^{2m}.
\end{cases}
$$

2. Li et al. [10]

$$
(M2):\begin{cases}
Q(v) = \dfrac{A_k}{v} + \dfrac{1}{B_k + C_k v}, \\
A_k = -\dfrac{m_k(m_k-2)(m_k+2)^3}{2\left(m_k^3 - 4m_k + 8\right)} t_k^{m_k}, \\
B_k = -\dfrac{\left(m_k^3 - 4m_k + 8\right)^2}{m_k\left(m_k^2 + 2m_k - 4\right)^3}, \\
C_k = \dfrac{m_k^2\left(m_k^3 - 4m_k + 8\right)}{\left(m_k^2 + 2m_k - 4\right)^3} t_k^{-m_k}.
\end{cases}
$$

3. Li et al. [11]

$$(M3):\begin{cases} Q(v)=\dfrac{B_k+C_kv}{1+A_kv},\\ A_k=-t_k^{-m_k},\quad B_k=-\dfrac{m_k^2}{2},\\ C_k=\dfrac{1}{2}m_k(m_k-2)t_k^{-m_k}.\end{cases}$$

4. Zhou et al. [30]

$$(M4):\begin{cases} Q(v)=A_kv+\dfrac{B_k}{v}+C_k,\\ A_k=\dfrac{m_k^4}{8}t_k^{-m_k},\quad B_k=\dfrac{m_k(m_k+2)^3}{8}t_k^{m_k},\\ C_k=-\dfrac{m_k\left(m_k^3+3m_k^2+2m_k-4\right)}{4}.\end{cases}$$

5. Zhou et al. [30]

$$(M5):\begin{cases} Q(v)=A_kv^2+B_kv+C_k,\\ A_k=\dfrac{m_k^4}{8}t_k^{-2m_k},\quad B_k=-\dfrac{m_k^3(m_k+3)}{4}t_k^{-m_k},\\ C_k=\dfrac{m_k\left(m_k^3+6m_k^2+8m_k+8\right)}{8}.\end{cases}$$

Selected particular methods are of low combinatorial complexity. Note that Zhou et al. [30] incorporates all known optimal multipoint methods of order four that use derivatives. Therefore, for comparison purposes in numerical tests we included some non optimal methods of the similar type (using derivatives).

1. Modified Newton's method $r=2$ when $m_k=m$, [24]

$$(M6):\begin{cases} u_k=\dfrac{f(x_k)}{f'(x_k)},\\ x_{k+1}=x_k-m_ku_k\end{cases}$$

2. Chebishev's method $r=3$ when $m_k=m$, [24]

$$(M7):\begin{cases} u_k=\dfrac{f(x_k)}{f'(x_k)},\\ x_{k+1}=x_k-\dfrac{m_k}{2}\,u_k\left(3-m_k+m_ku_k\dfrac{f''(x_k)}{f'(x_k)}\right)\end{cases}$$

3. Halley's method $r=3$ when $m_k=m$, [24]

$$(M8):\begin{cases} u_k=\dfrac{f(x_k)}{f'(x_k)},\\ x_{k+1}=x_k-2m\,\dfrac{u_k}{m+1-m\,u_k\,\dfrac{f''(x_k)}{f'(x_k)}}\end{cases}$$

4. Osada $r=3$ when $m_k=m$, [14]

$$(M9):\begin{cases} u_k=\dfrac{f(x_k)}{f'(x_k)},\\ x_{k+1}=x_k-\dfrac{m_k(m_k+1)}{2}u_k+\dfrac{(m_k-1)^2}{2}\dfrac{f'(x_k)}{f''(x_k)}\end{cases}$$

5. Dong $r = 3$ when $m_k = m$, [3]

$$(M10):\begin{cases} u_k = \dfrac{f(x_k)}{f'(x_k)}, \\ y_k = x_k - \dfrac{m_k}{m_k+1}u_k, \\ v_k = \dfrac{f'(y_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \dfrac{m_k}{m_k+1}\dfrac{u_k}{\left(\dfrac{m_k}{m_k+1}\right)^{-m_k} v_k - 1} \end{cases}$$

6. Neta-Li et al. $r = 4$ when $m_k = m$, [12], [10]

$$(M11):\begin{cases} u_k = \dfrac{f(x_k)}{f'(x_k)}, \quad t_k = \dfrac{m_k}{m_k+2} \\ y_k = x_k - 2t_k u_k, \\ v_k = \dfrac{f'(y_k)}{f'(x_k)} \\ z_k = y_k + 2t_k^{m_k}\dfrac{f(x_k)}{f'(y_k)}, \\ w_k = \dfrac{f'(z_k)}{f'(x_k)} \\ x_{k+1} = x_k - \dfrac{u_k}{a_k + b_k v_k + c_k w_k}, \\ a_k = -\dfrac{3m_k^4 + 16m_k^3 + 40m_k^2 - 176}{16m_k(m_k+8)}, \\ b_k = \dfrac{m_k^4 + 3m_k^3 + 10m_k^2 - 4m_k + 8}{8m_k(m_k+8)}t_k^{-m_k}, \\ c_k = \dfrac{(m_k-2)(m_k+2)^4}{16m_k^2(m_k+8)} \end{cases}$$

7. Neta-Li et al. $r = 4$ when $m_k = m$, [12], [10]

$$
(M12):\begin{cases}
u_k = \dfrac{f(x_k)}{f'(x_k)}, \quad t_k = \dfrac{m_k}{m_k+2} \\
y_k = x_k - 2t_k u_k, \\
v_k = \dfrac{f'(y_k)}{f'(x_k)} \\
z_k = x_k - 2t_k^{m_k}\dfrac{f(x_k)}{f'(y_k)}, \\
w_k = \dfrac{f'(z_k)}{f'(x_k)} \\
x_{k+1} = x_k - \dfrac{u_k}{a_k + b_k v_k + c_k w_k}, \\
a_k = \dfrac{m_k^6 - m_k^5 - 14m_k^4 + 12m_k^3 + 48m_k^2 - 80m_k + 32}{8m_k(m_k^3 + 2m_k^2 - 8m_k + 4)}, \\
b_k = -\dfrac{m_k\left(3m_k^4 - 6m_k^3 - 20m_k^2 + 40m_k - 16\right)}{16(m_k^3 + 2m_k^2 - 8m_k + 4)} t_k^{-m_k}, \\
c_k = \dfrac{m_k^3\left(m_k^2 - 4\right)}{16(m_k^3 + 2m_k^2 - 8m_k + 4)} t_k^{-m_k}
\end{cases}
$$

8. Neta-Li et al. $r = 4$ when $m_k = m$, [13], [10]

$$
(M13):\begin{cases}
u_k = \dfrac{f(x_k)}{f'(x_k)}, \quad t_k = \dfrac{m_k}{m_k+2} \\
y_k = x_k - 2t_k u_k, \\
v_k = \dfrac{f'(y_k)}{f'(x_k)} \\
z_k = y_k + 2t_k^{m_k}\dfrac{f(x_k)}{f'(y_k)}, \\
w_k = \dfrac{f'(z_k)}{f'(x_k)} \\
x_{k+1} = x_k - u_k\left(a_k + \dfrac{b_k}{v_k} + \dfrac{c_k}{w_k}\right), \\
a_k = \dfrac{m_k\left(m_k^4 + 4m_k^3 - 8m_k + 48\right)}{8\left(m_k^2 + 2m_k + 6\right)}, \\
b_k = \dfrac{m_k\left(m_k^3 + 12m_k^2 + 36m_k + 32\right)}{4\left(m_k^2 + 2m_k + 6\right)} t_k^{m_k}, \\
c_k = -\dfrac{m_k^2\left(m_k^3 + 6m_k^2 + 12m_k + 8\right)}{8\left(m_k^2 + 2m_k + 6\right)}
\end{cases}
$$

9. Neta-Li et al. $r = 4$ when $m_k = m$, [13], [10]

$$
(M14):\begin{cases}
u_k = \dfrac{f(x_k)}{f'(x_k)}, \quad t_k = \dfrac{m_k}{m_k+2}\\
y_k = x_k - 2t_k u_k,\\
v_k = \dfrac{f'(y_k)}{f'(x_k)}\\
z_k = x_k - 2t_k^{m_k}\dfrac{f(x_k)}{f'(y_k)},\\
w_k = \dfrac{f'(z_k)}{f'(x_k)}\\
x_{k+1} = x_k - u_k\left(a_k + \dfrac{b_k}{v_k} + \dfrac{c_k}{w_k}\right),\\
a_k = -\dfrac{m_k\left(2m_k^4 - m_k^3 - 12m_k^2 + 20m_k - 8\right)}{4\left(m_k^2 - 4m_k + 2\right)},\\
b_k = \dfrac{m_k\left(5m_k^4 + 10m_k^3 - 16m_k^2 - 24m_k + 16\right)}{8\left(m_k^2 - 4m_k + 2\right)}t_k^{m_k},\\
c_k = -\dfrac{m_k^3(m_k+2)^2}{8\left(m_k^2 - 4m_k + 2\right)}t_k^{m_k}
\end{cases}
$$

One more method was included into tests. It does not require the knowledge of the multiplicity m :
Modified Newton's method $r = 2$, [24]

$$
(M15):\left\{x_{k+1} = x_k - \frac{u(x_k)}{u'(x_k)} = x_k - \frac{f(x_k)f'(x_k)}{f'(x_k)^2 - f(x_k)f''(x_k)}\right.
$$

In total 15 iterative methods were tested under the same initial conditions. For those that require multiplicity m, strategies (5) and (6) were employed. Formulas (5) and (6) with and without rounding were tested, and some test functions with non integer multiplicity m were included in the tests.

Iterations were performed until at least one of the conditions was fulfilled:

$$
no.iter. \geq 10, \qquad |x_k - x_{k-1}| \leq 10^{-200}, \qquad |f(x_k)| \leq 10^{-200}.
$$

Computational order of convergence [16]

$$
r_C = \frac{\log|f(x_{k+1})/f(x_k)|}{\log|f(x_k)/f(x_{k-1})|},
$$

was calculated in each example, for each iteration, to verify conclusions derived in theory. Along side, a variant of the COR was calculated, as well: [16]

$$
r_\alpha = \frac{\log|(x_{k+1}-\alpha)/(x_k-\alpha)|}{\log|(x_k-\alpha)/(x_{k-1}-\alpha)|}.
$$

The following test functions were used with integer multiplicities.

$f_1(x) = (x-2)^{10}(x^4+x+1)e^{-x^2-4x}$,	$\alpha = 2$ $x_0 = 0.55$,	$m = 10$ $m_0 = 1$;
$f_2(x) = \left(e^{-x^4+x+2} - \cos(x+1) + x^7 + 1\right)^5$,	$\alpha = -1$, $x_0 = -2.5$,	$m = 5$, $m_0 = 1$;
$f_3(x) = (x-1)^{13}(x^{10}+x^3+1)\sin x$,	$\alpha = 1$, $x_0 = 0.1$,	$m = 13$, $m_0 = 1$;
$f_4(x) = e^{x^2-1}(\sin x)^4 + x^4\log(x^2+1)$,	$\alpha = 0$, $x_0 = 1.3$,	$m = 4$, $m_0 = 1$;
$f_5(x) = x^2 - (1-x)^{25}$,	$\alpha = 0.143739\ldots$, $x_0 = 1.25$,	$m = 1$, $m_0 = 1$;
$f_6(x) = \left(\sin^2 x - x^2 + 1\right)^2$,	$\alpha = 1.4044916\ldots$, $x_0 = 0.75$,	$m = 2$, $m_0 = 1$;
$f_7(x) = \left(x^2 - e^x - 3x + 2\right)^5$,	$\alpha = 0.257530\ldots$, $x_0 = 1.25$,	$m = 5$, $m_0 = 1$;
$f_8(x) = (\cos x - x)^3$,	$\alpha = 0.739085\ldots$, $x_0 = 1.25$,	$m = 3$, $m_0 = 1$;
$f_9(x) = \left(xe^{x^2} - \sin^2 x + 3\cos x + 5\right)^4$,	$\alpha = -1.20764\ldots$, $x_0 = 1.25$,	$m = 4$, $m_0 = 1$;
$f_{10}(x) = \left(e^{x^2+7x-30} - 1\right)^4$,	$\alpha = 3$, $x_0 = 1.25$,	$m = 4$, $m_0 = 1$;
$f_{11}(x) = \left(\ln x + \sqrt{x} - 5\right)^4$,	$\alpha = 8.309432\ldots$, $x_0 = 1.25$,	$m = 4$, $m_0 = 1$.

The set of test functions also included ones with rational root multiplicity:

$f_{12}(x) = (x-2)^{3/2}\left(x^4+x+1\right)e^{-x^2-4x}$,	$\alpha = 2$, $x_0 = 1.75$,	$m = 3/2$, $m_0 = 1$;
$f_{13}(x) = \left(x^5 - 8x^4 + 24x^3 - 34x^2 + 23x - 6\right)^{5/2}$,	$\alpha = 1$, $x_0 = 0$,	$m = 15/2$, $m_0 = 1$;
$f_{14}(x) = \left(x^2e^x - \sin x + x\right)^{2/3}$,	$\alpha = 0$, $x_0 = 1$,	$m = 4/3$, $m_0 = 1$;
$f_{15}(x) = \left((x-1)^3 - 1\right)^{9/4}$,	$\alpha = 2$, $x_0 = 3$,	$m = 27/4$, $m_0 = 1$

Results of the numerical experiments are given in the supplementary data sheet. We can note excellent convergence properties of the transformed methods. Methods $(M6)-M(14)$ were not analyzed through their error relations, however $m-$strategies hold applicable as can be seen from the data. Analysis presented in section 3 can be similarly conducted for methods $(M6)-M(14)$ to obtain convergence rate with $m-$strategies.

Conclusions drawn from numerical results confirm theoretical findings of sections two and three. $m-$strategy (5) remains stronger through experiments. Multipoint methods show supremacy in performance over one-point methods. As anticipated, the approximation quality of a particular method highly depends on the test function to which it was employed. As for any numerical method, it is a tool that should be designed to a particular application. Then it is most likely to perform at its best. However, during numerical trials $(M3)$, (5) scheme (both with and without rounding) has stood out as a particularly reliable and with excellent convergence rate. These are the results and conclusions based on the given test set and in the very high precision computing environment.

5. Acknowledgments

This research was supported by the Serbian Ministry of Education, Science and Technological Development under grant 451-03-68/2022-14/ 200102. Author wishes to thank the referees for their valuable comments. They have contributed largely to an improved content and presentation of this work.

References

[1] H. Arora, A. Cordero, J.R. Torregrosa, R. Behl, S. Alharbi, *Derivative-Free Iterative Schemes for Multiple Roots of Nonlinear*, Mathematics **10** (2022), 1530.

[2] C. Chun, B. Neta, *Comparative study of eighth-order methods for finding simple roots of nonlinear equations*, Numer. Algor. **74** (2017), 1169–1201.

[3] C. Dong, *A family of multipoint iterative functions for finding multiple roots of equations*, Int. J. Comput. Math. **21** (1987), 363–367.

[4] H. Esser, *Eine stets quadratisch konvergente Modifikation des Steffensen Verfahrens*, Computing **14** (1975), 367–369.

[5] Y.H. Geum, Y.I. Kim, B. Neta, *A class of two-point sixth-order multiple-zero finders of modified double-Newton type and their dynamics*, Appl. Math. Comput. **70** (2015), 387–400.

[6] J.L. Hueso, E. Martínez, C. Teruel, *Determination of multiple roots of nonlinear equations and applications*, J. Math. Chem. **53** (2015), 880–892.

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

[8] R. F. King, *Improving the Van de Vel Root-Finding Method*, Computing **30** (1983), 373–378.

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

[10] S. Li, L. Cheng, B. Neta, *Some fourth-order nonlinear solvers with closed formulae for multiple roots*, Comp. Math. Appl. **59** (2010), 126–135.

[11] S. Li, X. Liao, L. Cheng, *A new-fourth-order iterative method for finding multiple root of nonlinear equations*, Appl. Math. Comput. **215** (2009), 1288–1292.

[12] B. Neta, A.N. Johnson, *High-order nonlinear solvers for multiple roots*, Comput. Math. Comput. **55** (2008), 2012–2017.

[13] B. Neta, *Extension of Murakamis high-order nonlinear solver to multiple roots*, Int. J. Comput. Math. **8** (2010), 1023–1031.

[14] N. Osada, *An optimal multiple root-finding method of order three*, J. Comput. Appl. Math., **51** (1994), 131–133.

[15] I. Petković, Dj. Herceg, *Computers in mathematical research: the study of three-point root-finding methods*, Numer. Algor. **84** (2020), 1179–1198.

[16] M. S. Petković, B. Neta, L. D. Petković, J. Džunić, *Multipoint Methods for Solving Nonlinear Equations*, Elsevier/Academic Press, 2013.

[17] M. S. Petković, B. Neta, L. D. Petković, J. Džunić, *Multipoint methods for solving nonlinear equations: a survey*, Appl. Math. Comput. **226** (2014), 635–660.

[18] M.S. Petković, Lj.D. Petković, J. Džunić, *On an efficient method for the simultaneous approximation of polynomial multiple roots*, AADM **8** (2014), 73–94.

[19] M.S. Petković, Lj.D. Petković, *Construction and efficiency of multipoint root-ratio methods for finding multiple zeros*, J. of Comput. and Appl. Math. **351** (2019), 54–65.

[20] E. Schröder, *Über unendlich viele Algorithmen zur Auflösung der Gleichungen*, Math. Ann. **2** (1870), 317–365.

[21] J.R. Sharma, R. Sharma, *Modified Jarratt method for computing multiple roots*, Appl. Math. Comput. **217** (2010), 878–881.

[22] J.R. Sharma, S. Kumar, L. Jäntschi, *On Derivative Free Multiple-Root Finders with Optimal Fourth Order Convergence*, Mathematics **8** (2020), 1091.

[23] M. Vander Straeten, H. Van de Vel, *Multiple root-finding methods*, J. of Comput. and Appl. Math. **40** (1992), 105–114.

[24] J.F Traub, *Iterative methods for the solution of equations*, Prentice-Hall, 1964.

[25] H. Van de Vel, *A Method for Computing a Root of a Single Nonlinear Equation, Including Its Multiplicity*, Computing **14** (1975), 167–171.

[26] 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.

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

[28] T.J. Ypma, *Finding a multiple zero by transformations and Newton-like methods*, SIAM Review **25** (1983), 365–378.
[29] B.I. Yun, *A derivative free iterative method for finding multiple roots of nonlinear equations*, Appl. Math. Lett. **22** (2009), 1859–1863.
[30] X. Zhou, X. Chen, Y. Song, *Constructing higher-order methods for obtaining the multiple roots of nonlinear equations*, J. Comput. Appl. Math. **235** (2011), 4199–4206.