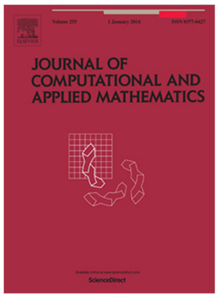

Construction and efficiency of multipoint root-ratio methods for finding multiple zeros

Miodrag S. Petković, Ljiljana D. Petković

PII: S0377-0427(18)30650-2
DOI: https://doi.org/10.1016/j.cam.2018.10.042
Reference: CAM 11988

To appear in: *Journal of Computational and Applied Mathematics*

Received date : 23 September 2018
Revised date : 16 October 2018

Construction and efficiency of multipoint root-ratio methods for finding multiple zeros

Miodrag S. Petković[1,*], **Ljiljana D. Petković**[2]

[1] *Faculty of Electronic Engineering, University of Niš, A. Medvedeva 14, 18000 Niš, Serbia*
[2] *Faculty of Mechanical Engineering, University of Niš, A. Medvedeva 14, 18000 Niš, Serbia*

Abstract

Several root-ratio multipoint methods for finding multiple zeros of univariate functions were recently presented. The characteristic of these methods is that they deal with m-th root of ratio of two functions (hence the name root-ratio methods), where m is the multiplicity of the sought zero, known in advance. Some of these methods were presented without any derivation and motivation, it could be said, out of the blue. In this paper we present an easy and entirely natural way for constructing root-ratio multipoint iterative methods starting from multipoint methods for finding simple zeros. In this way, a vast number of root-ratio multipoint methods for multiple zeros, existing as well new ones, can be constructed. For demonstration, we derive four root-ratio methods for multiple zeros. Besides, we study computational cost of the considered methods and give a comparative analysis that involves CPU time needed for the extraction of the m-th root. This analysis shows that root-ratio methods are pretty inefficient from the computational point of view and, thus, not suitable in practice. A general discussion on a practical need for multipoint methods of very high order is also considered.

AMS Mathematical Subject Classification (2010): 65H05.

Keywords: Solving nonlinear equations; Multipoint methods; Multiple zeros; Root-ratio approach; Computational efficiency.

1 Introduction

In the last ten years a lot of papers were published in the topic of iterative methods of optimal order for finding multiple zeros of univariate functions. Among them, several methods were constructed using *root-parameter approach* dealing with parameters computed by extracting m-th root of some values. In this paper this class of methods will be called *root-ratio methods*. We concentrate on the two issuses: (i) computational cost of these methods and (ii) demonstration of an easy procedure for natural construction of root-ratio methods starting from methods for finding simple zeros.

*Corresponding author
E-mail address: miodragpetkovic@gmail.con (M. S. Petković), ljiljana.petkovic@masfak.ni.ac.rs (L. D. Petković)

The paper is organized as follows. In Section 2 we present three two-point root-ratio methods of optimal order four published in [1]-[5]. In Section 3 we expose three tree-point root-ratio methods of optimal order eight proposed in [6]-[9]. An easy and fully natural procedure for constructing root-ratio methods of optimal order, demonstrated by three examples, is presented in Section 4. The computational efficiency of root-ratio methods is given in Section 5 applying a comparative analysis that involves CPU time needed for the extraction of the m-th root. This analysis has shown that root-ratio methods are pretty expensive and not-competitive with multipoint methods which do not require the extraction of the root even if the latter ones have lower order of convergence. See Remark 11 at the end of the paper.

2 Two-point fourth order methods

Let α be a zero of the known multiplicity $m \geq 1$ of a differentiable function f, and $k = 0, 1, \dots$ be the iteration index. We will assume that the chosen initial approximation x_0 is sufficiently close to the zero α so that the considered iterative processes are convergent. We start with two-point root-ratio methods for finding multiple zero of functions of one variable. Zhou et al. proposed in [1] the following family of iterative two-step methods for finding multiple zeros:

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - mG(u_k)\dfrac{f(x_k)}{f'(x_k)}, \quad u_k = \Big(\dfrac{f(y_k)}{f(x_k)}\Big)^{1/m} \end{cases} \qquad (k = 0, 1, \dots). \tag{1}$$

Under the conditions $G(0) = 0,\ G'(0) = 1,\ G''(0) = 4,\ G'''(0) < +\infty$, the order of convergence of the methods (1) is four. The iterative formula was derived correctly but in a complicated way. Observe that Taylor's series of G is

$$G(u) = u + 2u^2 + \cdots = u(1 + 2u) + \cdots = uZ(u) + \cdots = u(Z(0) + Z'(0)u^2) + \cdots$$

with $Z(0) = 1,\ Z'(0) = 2$ so that the weight function $G(u)$ in (1) can be replaced by $uZ(u)$ and the conditions $Z(0) = 1,\ Z'(0) = 2$. Compare this fact with the discussion given in Remark 1.

Lee et al. constructed in [2] the following family of fourth order methods:

$$\begin{cases} y_k = x_k - m \cdot \dfrac{f(x_k)}{f'(x_k) + \lambda f(x_k)}, \\ x_{k+1} = y_k - mW(u_k) \cdot \dfrac{f(x_k)}{f'(x_n) + 2\lambda f(x_k)}, \quad u_k = \Big(\dfrac{f(y_k)}{f(x_k)}\Big)^{1/m}, \end{cases} \tag{2}$$

where

$$W(u) = \frac{u(1 + (c + 2)u + ru^2)}{1 + cu}$$

and $\lambda,\ c,\ r$ are arbitrary parameters.

Remark 1. In most papers that study iterative methods with free parameters, the choice of parameters which improve convergence behavior of the proposed methods has not been considered. This is the case with the methods presented in this paper. It is hard to expect that the parameter λ in (2) can improve convergence characteristics of the methods (2) (after all, the proof was not given). Here, in (2) (see, also (4)), the situation is quite different than in the case of the modified Newton's method for simple zeros

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k) + af(x_k)}, \tag{3}$$

where the additional term $af(x_k)$ could adjust the tangent at the point $(x_k, f(x_k))$ in order to avoid overshooting. In spite of this (possibly favorable) property, the above Newton's modified method (3) is applied in practice very seldom since the choice of optimal parameter a is a very difficult problem. Considering the family (2) for $m \geq 2$, we observe that both values $f'(x_k)$ and $f(x_k)$ are very small in magnitude if the approximation x_k are sufficiently close (in magnitude) to the zero α of f so that the impact of of additional term is drastically lesser than in the case of the method (2). Consequently, the parameter λ does not play any important role so that one can take $\lambda = 0$ in (2) without loss of generality. The derivatives of W are given by

$$W'(u) = \frac{1 + 2(c+2)u + (2c + c^2 + 3r)u^2 + 2cru^3}{(1+cu)^2}, \quad W''(u) = \frac{4 + 2ru(3 + 3cu + c^2u^2)}{(1+cu)^3}.$$

Setting $u = 0$ one obtains $W(0) = 0,\ \ W'(0) = 1,\ \ W''(0) = 4$. Therefore, the method (2) is essentially a special case of the more general method (1).

Combining the approaches applied for the construction of the methods (1) and (2), Zafar et al. [3] proposed the two-point family of iterative methods

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k) + \gamma f(x_k)}, \\ x_{k+1} = y_k - mu_k Z(u_k)\dfrac{f(x_k)}{f'(x_k) + 2\gamma f(x_k)}, \quad u_k = \left(\dfrac{f(y_k)}{f(x_k)}\right)^{1/m} \end{cases} \quad (k = 0, 1, \ldots), \tag{4}$$

which reaches the order four under the conditions $Z(0) = 1,\ Z'(0) = 2,\ |Z''(0)| < \infty$. Here $c \in \boldsymbol{R}$ is an arbitrary parameter.

Remark 2. The discussion on the additional term given in Remark 1 for the family (2) is also valid for the family (4) and the additional terms $\gamma f(x_k)$ and $2\gamma f(x_k)$. Therefore, one can take $\gamma = 0$ without loss of generality and the iterative formula (4) reduces to the simpler form

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - mu_k Z(u_k)\dfrac{f(x_k)}{f'(x_k)}, \quad u_k = \left(\dfrac{f(y_k)}{f(x_k)}\right)^{1/m} \end{cases} \quad (k = 0, 1, \ldots). \tag{5}$$

Comparing the conditions related to (1) and (5) we observe one can take $G(u) \equiv uZ(u)$ with $Z(0) = 1,\ Z'(0) = 2$, as discussed above analyzing the family (1). Therefore, the families (1) and (5) (arising from (4) for $\gamma = 0$) are equivalent.

Using root ratio approach in a slightly different way, Liu et al. derived in [4] the family of two-point iterative methods

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - mG(u_k)\dfrac{f(x_k)}{f'(x_k)}, \quad u_k = \left(\dfrac{f'(y_k)}{f'(x_k)}\right)^{\frac{1}{m-1}} \end{cases} \quad (k = 0, 1, \ldots), \tag{6}$$

which converges with the order of convergence four under the conditions

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

The family (6) cannot be applied for finding simple zeros ($m = 1$), which is an obvious disadvantage. This drawback limits its applications in composite algorithms where a zero-finding method is part of the algorithm in which the case $m = 1$ is possible; actually, this case often appears in practical problems (for example, solving engineering problems).

Geum et al [5] constructed the following two-point root-ratio method

$$\begin{cases} y_k = x_k - \frac{2m}{m+2} \cdot \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - D(u_k)\dfrac{f(x_k)}{f'(x_k)}, \quad u_k = \left(\dfrac{f'(y_k)}{f'(x_k)}\right)^{1/n} \end{cases} \quad (k = 0, 1, \ldots), \tag{7}$$

where $n \in \boldsymbol{Q}$. The family (7) has the order four under specific conditions for the weight function D, presented in [5]. The choice $n \in \boldsymbol{N}$ is of greatest practical interest. In the case of simple zeros ($m = 1$), taking $n = 1$ the conditions for D reduce to

$$D(1) = 1, \quad D'(1) = -\frac{3}{4}, \quad D''(1) = \frac{9}{8} \tag{8}$$

(see Theorem 3.2 in [5]). Interestingly, the iterative formula (7) then becomes

$$\begin{cases} y_k = x_k - \frac{2}{3} \cdot \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - D(u_k)\dfrac{f(x_k)}{f'(x_k)}, \quad u_k = \dfrac{f'(y_k)}{f'(x_k)} \end{cases} \quad (k = 0, 1, \ldots),$$

which is Jarratt-like family of two-point methods (assuming that (8) holds), see, e.g., [14, pp. 74-75]. From this consideration we conclude that the choice $n = m$ in (7) would provide that the method (7) works for multiple as well as simple zeros without any altering iterative formula (7).

All five two-point methods (1), (2), (4), (6) and (7) are *optimal* in the sense of Kung-Traub conjecture [10]: *The highest order of convergence of an n-point method requiring $n+1$ function evaluations is 2^n.*

3 Three-point methods of eighth order

In this section we present some recent three-point optimal iterative methods for finding multiple zeros of the known multiplicity.

Behl et al. proposed in [6] (without derivation and motivation) the family of three-point methods of optimal order eight:

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - u_k S(h_k)\dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = z_k - u_k v_k R(h_k, v_k)\dfrac{f(x_k)}{f'(x_k)} \end{cases} \quad (k = 0, 1, \ldots), \tag{9}$$

where
$$u_k = \Big(\frac{f(y_k)}{f(x_k)}\Big)^{1/m}, \quad h_k = \frac{u_k}{a_1 + a_2 u_k}\ (a_1 \neq 0), \quad v_k = \Big(\frac{f(z_k)}{f(y_k)}\Big)^{1/m}.$$
Under some specific conditions for the function S (at the point 0) and R (at the point (0,0)), the order of convergence of the family of iterative methods (9) is eight.

Zafar et al. presented in [7] (without derivation and motivation) the following iterative three-point method

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - m u_k H(u_k)\dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = z_k - u_k v_k (B_1 + B_2 u_k) P(v_k) G(w_k)\dfrac{f(x_k)}{f'(x_k)} \end{cases} \quad (k = 0, 1, \ldots), \tag{10}$$

where $B_1, B_2 \in \boldsymbol{R}$ are free parameters, $H,\ P,\ G$ are the weight functions and

$$u_k = \Big(\frac{f(y_k)}{f(x_k)}\Big)^{1/m}, \quad v_k = \Big(\frac{f(z_k)}{f(y_k)}\Big)^{1/m}, \quad w_k = \Big(\frac{f(z_k)}{f(x_k)}\Big)^{1/m}. \tag{11}$$

Zafar et al. proposed in [8] (without derivation and motivation) the family of three-point methods

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - m u_k H(u_k)\dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = z_k - u_k P(u_k) G(v_k) L(w_k)\dfrac{f(x_k)}{f'(x_k)} \end{cases} \quad (k = 0, 1, \ldots), \tag{12}$$

where $u,\ v,\ w$ are defined by (11). The family (12) is obvious generalization of the family (10). Indeed, comparing (10) and (12), we note that $P(u) = B_1 + B_2 u,\ G(v) = vP(v)$. Clearly, the authors could work immediately at start with the product of weight functions $P(u)G(v)L(w)$ since the generalization is quite obvious.

Remark 3. Neither the idea nor the motivation for the construction of the methods (9), (10) and (12) were presented; the iterative formulas appear *out of the blue*, without preliminary explanation, introduction of the basic idea and derivation procedures. Such non-preamble approach is not in the spirit of the methodology of scientific work and educational principles, both useful for readers.

Remark 4. The need for arbitrary parameters in any iterative formula is discussed in Remark 1. This relates to the parameters λ, c, r in (2), γ in (4), and a_1 and a_2 in (9). Since any advantage of using a_1 and a_2 was not proved in [6], (actually, it is very difficult to prove it), it is logical to take $a_1 = 1$ and $a_2 = 0$ without loss of generality. At first sight, this is a special case but, in fact, it is a natural choice which simplifies artificially generalized iterative formula (9). Consequently, it follows $h_k = u_k$ and the family (9) reduces to a simpler iterative method (without h_k)

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - u_k S(u_k)\dfrac{f(x_k)}{f'(x_k)}, \qquad\qquad (k = 0, 1, \ldots) \\ x_{k+1} = z_k - u_k v_k R(u_k, v_k)\dfrac{f(x_k)}{f'(x_k)} \end{cases} \tag{13}$$

Another optimal three-point methods of order eight was recently constructed in [9] in the form

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - mL_f(u_k)\dfrac{f(x_k)}{f'(x_k)}, \qquad\qquad (k = 0, 1, \ldots) \\ x_{k+1} = z_k - m\big[L_f(u_k) + K_f(u_k, v_k)\big]\dfrac{f(x_k)}{f'(x_k)} \end{cases} \tag{14}$$

where u_k and v_k are defined by (11).

4 Can we construct them easier?

In this section we show that some of the above-presented methods can be derived using an easy way, well known in the literature for almost 150 years. The idea is credited to German mathematician E. Schröder, see his paper [11] and the English translation [12], and goes as follows in the case of Newton's method

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (k = 0, 1, \ldots)$$

applied to the function f having a zero α of multiplicity $m \geq 1$.

4.1 *Schröder's $f^{1/m}$ approach*

Let $F(x) = f(x)^{1/m}$. Then α is a simple zero of the function F. Applying Newton's method to F, we obtain

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)} = x_k - \frac{f(x)^{1/m}}{\frac{1}{m}f'(x)f(x)^{1/m-1}}$$

and hence

$$x_{k+1} = x_k - m\frac{f(x_k)}{f'(x_k)} \quad (k = 0, 1, \ldots),$$

which is well-known Schröder's method for finding multiple zero of the known multiplicity m. This useful idea was applied in many papers during the last 70 years.

Schröder's approach can be applied to multipoint methods for finding a simple zero in order to construct corresponding multipoint mehods for multiple zeros. As above, let f be a function having a zero of the known multiplicity $m \geq 1$. First, we set $F(x) = f(x)^{1/m}$ (Schröder's

$f^{1/m}$-approach) and derive the following relations:

$$
\begin{aligned}
\frac{F(x)}{F'(x)} &= \frac{f(x)^{1/m}}{\frac{1}{m}f'(x)f(x)^{1/m-1}} = m\frac{f(x)}{f'(x)},\\
\frac{F(y)}{F'(x)} &= \frac{f(y)^{1/m}}{\frac{1}{m}f'(x)f(x)^{1/m-1}} = m\frac{f(x)}{f'(x)}\Big(\frac{f(y)}{f(x)}\Big)^{1/m} = mu\frac{f(x)}{f'(x)}, \quad u = \Big(\frac{f(y)}{f(x)}\Big)^{1/m},\\
\frac{F(z)}{F'(x)} &= \frac{f(z)^{1/m}\cdot\frac{f(y)^{1/m}}{f(y)^{1/m}}}{\frac{1}{m}f'(x)f(x)^{1/m-1}} = m\frac{f(x)}{f'(x)}\Big(\frac{f(y)}{f(x)}\Big)^{1/m}\Big(\frac{f(z)}{f(y)}\Big)^{1/m} = muv\frac{f(x)}{f'(x)}, \quad v = \Big(\frac{f(z)}{f(y)}\Big)^{1/m},\\
\frac{F(y)}{F(x)} &= \frac{f(y)^{1/m}}{f(x)^{1/m}} = u, \qquad (15)\\
\frac{F(z)}{F(y)} &= \frac{f(z)^{1/m}}{f(y)^{1/m}} = v,\\
\frac{F(z)}{F(x)} &= \frac{f(z)^{1/m}}{f(x)^{1/m}} = w = uv.
\end{aligned}
$$

4.2 *Eighth-order family* (9) *derived by Schröder's approach*

For demonstration, we will apply the presented Schröder's approach to the following three-point family of iterative methods for finding a simple zero of a function f, proposed in [23]:

$$
\begin{cases}
y_k = x_k - \dfrac{f(x_k)}{f'(x_k)}\\[2mm]
z_k = y_k - p(u_k)\dfrac{f(y_k)}{f'(x_k)}, \qquad\qquad (k = 0, 1, \ldots),\\[2mm]
x_{k+1} = z_k - q(u_k, v_k)\dfrac{f(z_k)}{f'(x_k)}
\end{cases}
\qquad (16)
$$

where

$$u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(z_k)}{f(y_k)}$$

and $p(u)$ and $q(u, v)$ are functions of one and two variables, respectively. The following theorem has been proved in [23] (see also, [14]).

Theorem 1. *Let a, b and c be arbitrary constants. If p and q are arbitrary differentiable functions with Taylor's series of the form*

$$
\begin{aligned}
p(u) &= 1 + 2u + \frac{a}{2}u^2 + \frac{b}{6}u^3 + \cdots,\\
q(u, v) &= 1 + 2u + v + \frac{2+a}{2}u^2 + 4uv + \frac{c}{2}v^2 + \frac{6a+b-24}{6}u^3 + \cdots,
\end{aligned}
$$

then the family of three-point methods (16) *is of order eight. It is assumed that higher-order terms are represented by the dots, and they can take arbitrary values.*

Remark 5. The weight functions p and q in Theorem 1 are expressed in the form of polynomials. This form empirically shown poor convergence so that univariate rational functions (for p) and bivariate rational functions (for q) (with Taylor series given in Theorem 1) are used in practice, see, e.g., [9].

Assume that f has the zero α of multiplicity $m \geq 1$, known in advance. We modify (16) by using the above-mentioned Schröder's strategy with the function $F(x) = f(x)^{1/m}$ for which α is a simple zero. First, replacing f with F, rewrite (16) in the form

$$\begin{cases} y_k = x_k - \dfrac{F(x_k)}{F'(x_k)}, \\ z_k = y_k - P(u_k)\dfrac{F(y_k)}{F'(x_k)}, \\ x_{k+1} = z_k - Q(u_k, v_k)\dfrac{F(z_k)}{F'(x_k)} \end{cases} \quad (k = 0, 1, \ldots), \tag{17}$$

where

$$u_k = \frac{F(y_k)}{F(x_k)}, \quad v_k = \frac{F(z_k)}{F(y_k)}.$$

Using the relations (15), the family (17) can be written in the following form:

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - mu_kP(u_k)\dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = \Phi(x_k) := z_k - mu_kv_kQ(u_k, v_k)\dfrac{f(x_k)}{f'(x_k)} \end{cases} \quad (k = 0, 1, \ldots), \tag{18}$$

where

$$u_k = \Big(\frac{f(y_k)}{f(x_k)}\Big)^{1/m}, \quad v_k = \Big(\frac{f(z_k)}{f(y_k)}\Big)^{1/m}. \tag{19}$$

The weight functions $P(u)$ and $Q(u, v)$ should be determined in such a way that the iterative methods defined by (18) have the maximal order of convergence using only the calculation of $f(x)$, $f'(x)$, $f(y)$ and $f(z)$. In the ideal case, the order would reach *eight*, and the family (18) would be *optimal* according to Kung-Traub's hypothesis [10].

We proceed using a technique based on Taylor's series by employing symbolic computation in computer algebra system *Mathematica*. Since this technique was seen many times in existing papers, we present only an outline of convergence analysis.

We omit the iteration index k and define the errors

$$\varepsilon = x - \alpha, \quad \varepsilon_y = y - \alpha, \quad \varepsilon_z = z - \alpha, \quad \hat{\varepsilon} = \hat{x} - \alpha,$$

where $\hat{x}$ is a new approximation x_{k+1}. Introduce

$$C_r = \frac{m!}{(m+r)!}\frac{f^{(m+r)}(\alpha)}{f^{(m)}(\alpha)} \quad (r = 1, 2, \ldots).$$

We will use the following development of the function f about the zero α of multiplicity m

$$f(x) = \frac{f^{(m)}(\alpha)}{m!}\varepsilon^m\Big(1 + C_1\varepsilon + C_2\varepsilon^2 + C_3\varepsilon^3 + C_4\varepsilon^4 + C_5\varepsilon^5 + C_6\varepsilon^6 + C_7\varepsilon^7 + C_8\varepsilon^8 + O(\varepsilon^9)\Big),$$

and a program in *Mathematica*. As usual, in finding the weight functions P and Q, we represent these functions by their Taylor's series at the neighborhood of $u=0$ (for P) and $(u,v)=(0,0)$ (for Q):

$$\begin{aligned} &P(u) = P(0) + P'(0)u + \frac{P''(0)}{2}u^2 + \cdots, \\ &Q(u,v) = Q(0,0) + Q_{u0}(0,0)u + Q_{0v}(0,0)v + \frac{1}{2!}\Big(Q_{uu}(0,0)u^2 + 2Q_{uv}(0,0)uv + Q_{vv}(0,0)v^2\Big) + \cdots \end{aligned}$$

Subscript indices denote partial derivatives. In the last developments, as well as in the program, the following notation is used:

$$\begin{aligned} &fam = f^{(m)}(\alpha),\ fx = f(x),\ fy = f(y),\ fz = f(z),\ f1x = f'(x), \\ &e = \varepsilon,\ ey = \varepsilon_y,\ ez = \varepsilon_z,\ e1 = \hat{\varepsilon}, \\ &Pr = \frac{P^{(r)}(u)}{du}\Big|_{(u=0)}\ (r = 0,1,2), \\ &Q00 = Q(0,0),\quad Qu0 = \frac{\partial\, Q}{\partial\, u}\Big|_{(u,v)=(0,0)},\quad Q0v = \frac{\partial\, Q}{\partial\, v}\Big|_{(u,v)=(0,0)}, \\ &Quu = \frac{\partial^2 Q}{\partial\, u^2}\Big|_{(u,v)=(0,0)},\quad Quv = \frac{\partial^2 Q}{\partial\, u\,\partial\, v}\Big|_{(u,v)=(0,0)},\quad Qvv = \frac{\partial^2 Q}{\partial\, v^2}\Big|_{(u,v)=(0,0)}. \end{aligned}$$

The coefficients of Taylor's development of the weight functions P and Q are determined using an interactive approach by combining the program realized in *Mathematica* (two parts) and the annihilation of coefficients standing at ε of lower power.

PART I (*Mathematica*)

```
fxx=1+C1*e+C2*e^2+C3*e^3+ C4*e^4+C5*e^5+C6*e^6+ C7*e^7+C8*e^8;
fx=fam/m!*e^m*fxx; fx1=D[fx,e]; newt=Series[fx/fx1,{e,0,8}];
ey=e-m*newt; fyy=1+C1*ey+C2*ey^2+3*ey^3+C4*ey^4;
yx=fyy*Series[1/fxx,{e,0,8}];
yxm=Series[yx^1/m,{e,0,8}]
u=yxm*ey/e;
P=P0+P1*u+P2/2*u^2+P3/6*u^3+P4/24*u^4;
ez=Series[ey-m*u*P*newt//FullSimplify,{e,0,8}]
```

This program gives the following OUTCOME:

$$\begin{aligned} \varepsilon_z &= \sum_{r=2}^{8} S_r\varepsilon^r \\ &= \frac{(C_1 - C_1P_0)\varepsilon^2}{m} \\ &\quad + \frac{(-2C_2m(-1+P_0) + C_1^2(-1+m(-1+P_0)+3P_0-P_1))\varepsilon^3}{m^2} \\ &\quad + \frac{1}{2m^3}\Big(-6C_3m^2(-1+P_0) + 2C_1C_2m(-4+3m(-1+P_0)+11P_0-4P_1) \\ &\quad + C_1^3(2-13P_0+10P_1+m(4-2m(-1+P_0)-11P_0+4P_1)-P_2)\Big)\varepsilon^4 + \sum_{r=5}^{8} U_r + O(e^9). \end{aligned}$$

To annihilate coefficients by ε^2 and ε^3, from the condition $S_2 = 0$ and $S_3 = 0$ we find

$$P_0 = 1, \quad P_1 = 2, \quad P_2 \text{ arbitrary}, \tag{20}$$

yielding

$$\varepsilon_z = \frac{-2mC_1C_2 + C_1^3(9 + m - P_2)}{2m^3}\varepsilon^4 + O(\varepsilon^5). \tag{21}$$

The part II of the program uses previously found entries and serves for finding additional conditions which provide optimal order eight.

PART II - CONTINUATION (*Mathematica*)

```
fzz=1+C1*ez+C2*ez^2; fyy=1+C1*ey+C2*ey^2+ C3*ey^3+C4*ey^4;
zy=fzz*Series[1/fyy{e,0,8}]; zym=Series[zy^(1/m),{e,0,8}];
v=zym*ez/ey;
Q=Q00+Qu0*u+Q0v*v+Quu/2*u^2+Qvv/2*v^2+Quv*u*v;
e1=Series[ez-m*u*v*Q*newt,{e,0,8}]//FullSimplify
```

The error $\hat{\varepsilon} = \hat{x} - \alpha$ $(= e1)$ is given in the form

$$\varepsilon_1 = \sum_{r=4}^{8} T_r\varepsilon^r + O(\varepsilon^9)$$

From the conditions $T_4 = 0,\ T_5 = 0,\ T_6 = 0,\ T_7 = 0$, we find the coefficients

$$Q_{00} = 1,\ Q_{u0} = 2,\ Q_{0v} = 1,\ Q_{uv} = 4,\ P_2 \text{ is arbitrary},\ P_3 = 24 - 6P_2,\ Q_{uu} = P_2 + 2. \tag{22}$$

In addition, we obtain

$$\begin{aligned} T_8 \;=\; & -\frac{1}{48m^7}\Big\{C_1\Big[C_1^2(m - P_2 + 9) - 2mC_2\Big] \\ & \times\Big[C_1^4\left(-14m^2 + 3Q_{tt}(m - P_2 + 9)^2 + 6mP_2 - 204m + 150P_2 - P_4 - 1054\right) \\ & -12mC_1^2C_2\big(Q_{tt}(m - P_2 + 9) - 4m + P_2 - 34\big) - 24m^2C_1C_3 + 12m^2C_2^2(Q_{tt} - 2)\Big]\Big\}, \end{aligned}$$

so that we can write

$$\hat{\varepsilon} = T_8\varepsilon^8 + O(\varepsilon^9),$$

that is,

$$\varepsilon_{k+1} - \alpha = \Phi(x_k) - \alpha = O(\varepsilon_k^8), \quad \varepsilon_k = x_k - \alpha.$$

In this way we have proved the following assertion.

Theorem 2. *If the initial approximation x_0 is sufficiently close to the zero α of f and the conditions* (20) *and* (22) *are valid, then the order of the three-point family* (18) *is eight.*

Note that T_8 tends to the asymptotic error constant of the family (18) when $\varepsilon \to 0$. According to Theorem 2, it follows that the three-point family (18) is *optimal*.

The simplest forms of the weight functions P and Q are their truncated Taylor series

$$P(u) = 1 + 2u + \beta\, u^2 + (4 - 2\beta)\, u^3, \quad Q(u, v) = 1 + 2u + v + 4uv + (\beta + 1)u^2, \quad \beta \text{ is arbitrary.}$$

Finding a list of particular weight functions P and Q is a routine work and it is left to the interested reader as exercise.

4.3 *Eighth-order family* (12) *derived by Schröder's approach*

The presented Schröder's $f^{1/m}$-approach can also be applied for the derivation of the method (12). Let us start from the following three-point method for simple zeros

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - u_k H(u_k)\dfrac{f(x_k)}{f'(x_k)} \\ x_{k+1} = z_k - u_k P(u_k)G(v_k)L(w_k)\dfrac{f(x_k)}{f'(x_k)} \end{cases} \qquad (k = 0, 1, \ldots), \tag{23}$$

where

$$u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(z_k)}{f(y_k)}, \quad w_k = \frac{f(z_k)}{f(x_k)}.$$

This method has the order 8 under the conditions

$$\begin{aligned} &H(0) = 1,\ H'(0) = 2,\ P'(0) = 2P(0),\ P''(0) = P(0)(2 + H''(0)), \\ &L'(0) = 2L(0),\ P'''(0) = P(0)(H'''(0) + 6H''(0) - 24), \\ &G(0) = 0,\ G'(0) = \frac{1}{L(0)P(0)},\ G''(0) = \frac{2}{L(0)P(0)}. \end{aligned} \tag{24}$$

The denotation of weight functions in (23) and (24) is adjusted to the denotation used in [8].

Applying Schröder's $f^{1/m}$-approach and (15) we obtain the family of three-point methods for finding multiple zeros

$$\begin{cases} y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - mu_k H(u_k)\dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = z_k - mu_k P(u_k)G^*(v_k)L(w_k)\dfrac{f(x_k)}{f'(x_k)} \end{cases} \qquad (k = 0, 1, \ldots), \tag{25}$$

which is equivalent to (12) setting $mG^*(v) = G(v)$ (compare (12) and (25)).

Remark 6. Among other methods, the first author of this paper presented the family of methods (23) in his lecture under the title *Multipoint methods for solving nonlinear equations* at International conference *Computational Methods in Applied Mathematics* (Berlin, 2012). However, due to the similarity to the iterative formula (4.91) in the book [14, p 151], the author did not publish (23). Other possible sources of (23) are not known to the authors.

4.4 *A modification of the family* (12)

Since

$$w = \Big(\frac{f(z)}{f(x)}\Big)^{1/m} = \Big(\frac{f(y)}{f(x)}\Big)^{1/m}\Big(\frac{f(z)}{f(y)}\Big)^{1/m} = u \cdot v,$$

the weight function $L(w)$ can be omitted in (12). Then we can construct in an easy way the following family of three-point methods for finding multiple zeros involving two parametric functions

$$
\begin{cases}
y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\[2mm]
z_k = y_k - mu_k H(u_k)\dfrac{f(x_k)}{f'(x_k)}, \\[2mm]
x_{k+1} = z_k - mu_k v_k(1+2u_k v_k)P(u_k)G(v_k)\dfrac{f(x_k)}{f'(x_k)}.
\end{cases} \tag{26}
$$

Making suitable changes in the above program in *Mathematica* (see § 4.2), we prove that the method (26) is of order eight under the following conditions:

$$
\begin{aligned}
&H(0) = 1, \quad H'(0) = 2, \quad H''(0) = m + 9, \\
&P(0) = 1, \quad P'(0) = 2, \quad P''(0) = m + 11, \quad P'''(0) = 30 + 6m + H'''(0), \\
&G(0) = 1, \quad H'(0) = 1.
\end{aligned}
$$

4.5 *Fourth order methods for multiple zeros*

Chun constructed in [15] the following two-point family of iterative methods for finding simple zeros

$$
\begin{cases}
y_k = x_k - \dfrac{f(x_k)}{f'(x_k)} \\[2mm]
x_{k+1} = y_k - \dfrac{f(y_k)}{h(u_k)f'(x_k)}, \quad u_k = \dfrac{f(y_k)}{f(x_k)},
\end{cases} \tag{27}
$$

where $h(u)$ is the weight function. The method (27) is of fourth order under the condition $h(0) = 1,\ h'(0) = -1,\ |h''(0)| < +\infty$. To generate suitable two-point methods, sometimes it is necessary to develop the function h into Taylor's or geometric series.

A slightly more direct approach without altering the weight function, based on Chun's idea, was given in [16] in the form

$$
\begin{cases}
y_k = x_k - \dfrac{f(x_k)}{f'(x_k)}, \\[2mm]
x_{k+1} = y_k - p(u_k)\dfrac{f(y_k)}{f'(x_k)}, \quad u_k = \dfrac{f(y_k)}{f(x_k)}.
\end{cases} \tag{28}
$$

The family (28) possesses the optimal order four if $p(0) = 1,\ p'(0) = 2$ and $|p''(0)| < +\infty$.

Proceeding in the same way as in the case of the family of three-point methods (16) and using (15), we obtain from (28) the fourth order two-point family of iterative methods for finding multiple zeros

$$
\begin{cases}
y_k = x_k - m\dfrac{f(x_k)}{f'(x_k)}, \\[2mm]
x_{k+1} = y_k - mu_k P(u_k)\dfrac{f(x_k)}{f'(x_k)}, \quad u_k = \left(\dfrac{f(y_k)}{f(x_k)}\right)^{1/m},
\end{cases} \tag{29}
$$

where $P(u)$ is the weight function which satisfies

$$
P(0) = 1, \quad P'(0) = 2, \tag{30}
$$

that is, its truncated Taylor series is

$$
P(u) = 1 + 2u.
$$

This result is expected since the iterative formula (29) coincides with the first two steps of the family (18). Explicit expression of $\hat{\varepsilon} = \hat{x} - \alpha$ (regarding (29)) is given by (21) (for ε_z).

Several examples of functions that satisfy (30) are listed below.

$$P(u) = \frac{1+\beta u}{1+(\beta-2)u} \quad \text{(of King's type, see [17])}, \quad P(u) = \left(1 + \frac{2u}{r}\right)^r,$$
$$P(u) = \frac{1+\gamma u^2}{1-2u}, \quad P(u) = \frac{1}{1-2u+au^2}, \quad P(u) = \frac{u^2+(c-2)u-1}{cu-1}.$$

where $r \in \boldsymbol{Q}$ and $\gamma,\ a,\ c \in \boldsymbol{R}$ are arbitrary parameters.

4.6 *Equivalence of methods*

Some words about the equivalence of the methods presented in Section 2 and 3, and the methods derived by Schröder's approach in Section 4.

Equivalence 1. In Section 2 it was shown that the simplified methods (2) and (4) (assuming slight simplifications by taking $\lambda = 0$ in (2) and $\gamma = 0$ in (4)) are special cases of the method (1) proposed by Zhou et al. [1]. Comparing iterative methods (1) and (29) it is evident that both formulas are equivalent, which is easily obtained taking $uP(u) \equiv G(u)$. Moreover, the family (29) requires only two conditions $P(0) = 1,\ P'(0) = 2$ compared to three conditions $G(0) = 0,\ G'(0) = 1,\ G''(0) = 4$. It is important to note that the derivation of the family (1) and latter convergence analysis are more complicated than for the family (29). Furthermore, the derivation of (29), based on Schröder's approach, is entirely natural and crystal clear.

Equivalence 2. The family (14), obtained by natural choice $a_1 = 1,\ a_2 = 0$ in the family (9) (proposed in [6]), is equivalent to the family (18), which is evident setting $S(u) \equiv mP(u)$ and $R(u,v) \equiv mQ(u,v)$. The family (9) (and, consequently, (14)) was presented without derivation and motivation, while the family (18) was derived on an easy and obvious way using Schröder's approach.

Remark 7. Apart from the family (6), other optimal multipoint methods for finding a simple zero (some of them are presented in the book [14]) can be transformed by introducing $F(x) = f(x)^{1/m}$ to multipoint methods for approximating a multiple zero keeping optimal order of convergence. This subject is left to readers. However, the authors of this paper do not expect new papers in this direction since such methods are pretty expensive. This is the subject of the next section.

5 Root-ratio methods are not competitive

In Section 4 we have demonstrated a general procedure for constructing multipoint methods for multiple zeros using basic iterative formulas for simple zeros and Schröder's $f^{1/m}$-approach. We observe that all previously presented methods deal with real or complex values of the forms

$$\left(\frac{f(y_k)}{f(x_k)}\right)^{1/m}, \quad \left(\frac{f(z_k)}{f(x_k)}\right)^{1/m}, \quad \left(\frac{f(z_k)}{f(y_k)}\right)^{1/m}.$$

Computer algebra systems and computer arithmetics of digital computers often meet the problem of finding the m-th root for arbitrary m. In the case of specific values of m they find the *principal value* of the m-root $z^{1/m}$ (m is natural number given as numerical entry) among m values of the sought root in the form

$$z^{1/m} = |z|^{1/m}\Big(\cos\frac{\theta}{m} + i\,\sin\frac{\theta}{m}\Big), \quad \theta = \operatorname{Arg} z \in (-\pi, \pi]. \tag{31}$$

From (31) it is clear that the computation of the m-root consumes a lot of CPU time. The following test, implemented on PC with Intel-i7 processor and clock speed 2.8 GHz, has given CPU times (expressed in μsec) for different values of m in calculation of $\big((a+ib)/(c+id)\big)^{1/m}$. For the authenticity of the test, one million experiments have been performed taking random numbers for a, b, c, d in each cycle to eliminate the use of possibly memorized data from previous cycles. In the real case we set $b = 0,\ d = 0$. The experiments were realized in computer algebra system *Mathematica* in multi-precision arithmetic with 34 significant decimal digits, which corresponds to quadruple-precision of IEEE 754 floating-point arithmetic. The average CPU times for one evaluation are given in Table 1.

m	1	2	3	4	5	6	7
CPU (in μsec), real case	5.25	22.7	32.24	31.86	32.25	32.03	33.1
CPU (in μsec), complex case	13.77	54.22	64.42	64.07	65.83	68.05	66.3

Table 1: CPU times in the calculation of the m-th root of real and complex numbers

In our experiments we observed that the CPU times for $m \geq 3$ almost do not change in the real as well as the complex case (see Table 1). In this way we are able to find reliable ratio of computation times for $m \geq 3$ (multiple zeros) and $m = 1$ (simple zeros):

$$\text{Real case:} \quad \frac{\mathrm{CPU}_{(m\geq 3)}}{\mathrm{CPU}_{(m=1)}} \approx 6.2, \qquad \text{Complex case:} \quad \frac{\mathrm{CPU}_{(m\geq 3)}}{\mathrm{CPU}_{(m=1)}} \approx 4.8. \tag{32}$$

Remark 8. For comparison purpose, we performed one million calculations of the value of a polynomial of degree 20 with real coefficients and complex argument, both chosen randomly in each cycles (Horner's scheme was used). Average CPU times for one evaluation was 116 μsec, which is only two times slower than CPU time in calculation of m-th root in the case of complex numbers (see Table 1).

Remark 9. We emphasize that the ratio of CPU times strongly depends on the used computing platform (usually the hardware or the operating system) and implemented software (assuming the use of quadruple precision). It may differ from the values given by (32), but not too much, the ratios are certainly higher than 3 if $m \geq 3$. This fact can be observed from Table 11 in the paper [5]. We also mention that the execution CPU time of the square root operation ($m = 2$) requires significantly lesser CPU time since this operation is realized by special (more effective) algorithms, see the book *Modern Computer Arithmetic* by Brent and Zimmermann [18].

Having in mind Remark 8 and 9 we conclude that root-ratio methods are expensive from a computational point of view. From (32) and Remark 8 we can draw trustworthy conclusion that root-ratio multipoint methods, such as (1), (2), (6), (9), (10), (12), (14), (25), (26) and other non-listed methods (if there exist), are inefficient. As mentioned in Remark 7, further work on the construction of root-ratio methods is pointless and does not make an advance in the topic. Combining various weight functions in order to derive "new methods" is rather a kind of play and inevitably leads to minor modifications without a proper importance. This fact was emphasized in [19] but the construction of modest modifications of original contributions has continued. Different iterative formula does not mean automatically that a contribution to the topic was achieved.

Is there a good alternative multi-method for multiple zeros which is efficient and convenient for applications? The answer is yes, and it is very likely known to many authors who work in the area of iterative processes. Li et al. proposed in [20] the following (optimal) two-point method of order four:

$$\begin{cases} y_k = x_k - \frac{2m}{m+2} \cdot \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = x_k - \dfrac{\frac{1}{2}m(m-2)A_m t_k - \frac{m^2}{2}}{1 - R_m t_k} \cdot \dfrac{f(x_k)}{f'(x_k)}, \quad t_k = \dfrac{f'(y_k)}{f'(x_k)},\ A_m = \left(\frac{m+2}{m}\right)^m. \end{cases} \tag{33}$$

Zhou et al. [21] later proposed the generalization of the method (33) in the form

$$\begin{cases} y_k = x_k - \frac{2m}{m+2} \cdot \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = x_k - \phi(t_k) \cdot \dfrac{f(x_k)}{f'(x_k)}, \quad t_k = \dfrac{f'(y_k)}{f'(x_k)}, \end{cases} \tag{34}$$

which has the order four under the specific conditions for the weight function ϕ. See, also, [22]. The notion of "convenient" is explained in Remark 11.

We conclude this paper with two remarks of general interest.

Remark 10. The presence of arbitrary parameters in any zero-finding iterative formula makes sense only if these parameters improve characteristics of presented methods (such as acceleration of convergence, wider domain of convergence, more accurate approximations, lower computational cost, etc.). Otherwise, from an algorithmic point of view, free parameters should be chosen so that an iterative formula is as simple as possible – numerical analysts and programmers will always choose the simpler formula in such a way that the best characteristics of the employed methods are maintained. Inserting numerous useless parameters does not make a method better or more general in the genuine sense. Unfortunately, many authors construct "novel" iterative formulas by adding parameters in an artificial way or by varying different weight functions. In essence, such methods are only modest modifications of existing methods and offer a little contribution to the topic. Discussions on choosing the parameters that improve results to some extent can be found in the paper [9].

Remark 11. It should be emphasized that very high accuracy of solutions of nonlinear equations, provided by root-solvers of order eight or more, is not needed for solving a huge number of practical problems; fourth order methods (such as (33) and (34)) produce quite satisfactory results in practice. The question *"how many decimals of zero approximations do we really need in practice?"* is equivalent to the question *"how many decimals of* π *do we really need in practice"?"* A pretty convincing answer can be found in the issue of *NASA/JPL Edu*, March 16, 2016:

• For interplanetary navigation with spacecraft Voyager 1 (launched in 1977, distant from Earth about 22 billion km), Jet Propulsion Labaratory (California Institute of California, Pasadena, USA) and NASA use very accurate calculations involving π with most 15 decimal digits! Not more! The distance error is about 5 cm!

• The radius of the visible universe is about 46 billion light years. To express the circumference of a circle with this radius via the diameter a hydrogen atom (the simplest atom) we need at most 40 decimal digits of π!

Acknowledgement. This work was supported by the Serbian Ministry of Education and Science under Grant 174022.

References

[1] X. Zhou, X. Chen, Y. Song, Families of third and fourth order methods for multiple roots of nonlinear equations, Appl. Math. Comput. 219 (2013) 6030–6038.

[2] M. Y. Lee, Y. I. Kim, Á. A. Magreñán, On the dynamics of tri-parametric family of optimal fourth-order multiple-zero finders with a weight function of the principal mth root of a function-to-function ratio, Appl. Math. Comput. 315 (2017) 564–590.

[3] F. Zafar, A. Cordero, J.R. Torregrosa, Stability analysis of a family of optimal fourth-order methods for multiple roots, Numer Algor (2018), doi.org/10.1007/s11075-018-0577-0 (online).

[4] B. Liu, X. Zhou, A new family of fourth-order methods for multiple roots of nonlinear equations, Nonlinear Anal. Model. Control 18 (2013) 143–152.

[5] Y.H. Geum, Y.I. Kim, B. Neta, A family of optimal quartic-order multiple-zero finders with a weight function of the principal kth root of a derivative to derivative ratio and their basins of attraction, Math. Comput. Simulation 136 (2017) 1–21.

[6] R. Behl, A. Cordero, S. Motsa, J. R. Torregrosa, An eighth-order family of optimal multiple root finders and its dynamics, Numer. Algor. 77 (2018) 1249–1272.

[7] F. Zafar, A. Cordero, R. Quratulain, J. R. Torregrosa, Optimal iterative methods for finding multiple roots of nonlinear equations using free parameters, J. Math. Chem. 56 (2018) 1884–1901.

[8] F. Zafar, A. Cordero, S. Sultana, J. R. Torregrosa, Optimal iterative methods for finding multiple roots of nonlinear equations using weight functions and dynamics, J. Comput. Appl. Math. 345 (2018) 352–374.

[9] Y.H. Geum, Y.I. Kim, B. Neta, Constructing a family of optimal eighth-order modified Newton-type multiple-zero finders along with the dynamics behind their purely imaginary extraneous fixed points, J. Comput. Appl. Math. 333 (2018) 131–156.

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

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

[12] G.W. Stewart, On infinitely many algorithms for solving equations, drum.lib.umd.edu/handle/1903/577 (CS-TR-2990.ps) (University of Maryland Library).

[13] J. Džunić, M. S. Petković, L. D. Petković, A family of optimal three-point methods for solving nonlinear equations using two parametric functions, Appl. Math. Comput. 217 (2011) 7612–7619.

[14] M. S. Petković, B. Neta, L. D. Petković, J. Džunić, Multipoint Methods for Solving Nonlinear Equations, Elsevier, Amsterdam-Boston-Heidelberg-London-New York, 2013.

[15] C. Chun, Some fourth-order iterative methods for solving nonlinear equations, Appl. Math. Comput. 195 (2008) 454–459.

[16] M. S. Petković, L. D. Petković, Families of optimal multipoint methods for solving nonlinear equations: a survey, Appl. Anal. Discrete Math. 4 (2010) 1–22.

[17] R. F. King, A family of fourth order methods for nonlinear equations, SIAM J. Numer. Anal. 10 (1973) 876–879.

[18] R. Brent, P. Zimmermann, Modern Computer Arithmetic, Cambridge University Press, Cambridge 2011.

[19] 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.

[20] 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.

[21] X. Zhou, X. Chen, Y. Song, Construction of higher order methods for multiple roots of nonlinear equations, J. Comput. Appl. Math. 235 (2011) 4199–4206.

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

[23] J. Džunić, M. S. Petković, L. D. Petković, A family of optimal three-point methods for solving nonlinear equations using two parametric functions, Appl. Math. Comput. 217 (2011) 7612–7619.