Computers and Mathematics with Applications 63 (2012) 764–774

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

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

Finding the solution of nonlinear equations by a class of optimal methods

M. Sharifi [a,∗], D.K.R. Babajee [b], F. Soleymani [a]

[a] Department of Mathematics, Islamic Azad University, Zahedan Branch, Zahedan, Iran
[b] Department of Mathematics, Port Louis North State Secondary School, Cite la Cure, Port-Louis, Mauritius

ARTICLE INFO

Article history:
Received 12 September 2011
Received in revised form 20 November 2011
Accepted 21 November 2011

Keywords:
Multiple roots
Order of convergence
Multipoint iterations
Jarratt method
Optimal order

ABSTRACT

This paper is devoted to the study of an iterative class for numerically approximating the solution of nonlinear equations. In fact, a general class of iterations using two evaluations of the first order derivative and one evaluation of the function per computing step is presented. It is also proven that the class reaches the fourth-order convergence. Therefore, the novel methods from the class are Jarratt-type iterations, which agree with the optimality hypothesis of Kung–Traub. The derived class is further extended for multiple roots. That is to say, a general optimal quartic class of iterations for multiple roots is contributed, when the multiplicity of the roots is available. Numerical experiments are employed to support the theory developed in this work.

1. Introduction

Finding roots of a nonlinear equation $f(x) = 0$ is a significant problem in science and engineering, [1–3]. As the order of an iterative method without memory increases, so does the number of (functional) evaluations per step. The efficiency index gives a measure of the balance between those quantities, based on the formula $p^{1/n}$, where p is the order of convergence of the method and n the number of evaluations per full cycle, [4–7]. Herein, another eye-catching finding plays a crucial role as follows. Kung and Traub conjectured in [8] that the order of convergence of any multipoint without memory method with n evaluations cannot exceed the bound 2^{n-1}, called the optimal order. Thus, the optimal order for a method with three evaluations per iteration would be 4. Jarratt method [9] is an example of the optimal fourth-order methods, because it only uses three evaluations per cycle. For further reading, one may refer to [10–13].

We here also remark that iterations without memory using one evaluation of the function and two evaluations of the first order derivative per iteration are called as Jarratt-type methods in the literature.

In this paper, we present multipoint methods for solving nonlinear equations, constructed by the weight function approach. These methods will be referred to Jarratt-type without memory iterative methods. More precisely, in Section 2, we construct a two-point class of methods without memory of the order of convergence four. Numerical examples are given in Section 3 to illustrate convergence behavior of the new methods for simple roots. We also discuss that the quartic method of Jarratt is a special case of our contributed class. It will be seen from these examples that a special choice of initial approximations provides considerably good accuracy of approximations to the roots, obtained by the proposed methods. Furthermore, the derived class will be generalized for the case of multiple roots when the multiplicity $m > 1$ is known. In fact, a general two-point class of multiple root-finders will be contributed and analyzed in Section 4. Numerical comparison in this regard, will be furnished in Section 5. Finally, Section 6 gives the concluding comments of this research article.

∗ Corresponding author.
E-mail addresses: iau.sharifi@yahoo.com (M. Sharifi), dkrbabajee@gmail.com (D.K.R. Babajee), fazlollah.soleymani@gmail.com (F. Soleymani).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.11.040

2. New findings

The main goal of this section is to attain as fast as possible (local) order of convergence with minimal computational costs and three functional evaluations in constructing iterative methods without memory for solving nonlinear equations. Therefore, we aim at presenting a new general way in constructing without memory iterations. We take into account a third-order iteration, which is known as Heun's iteration [14,15] in the literature

$$\begin{cases} y_n = x_n - \dfrac{2}{3}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{f(x_n)}{4}\left(\dfrac{1}{f'(x_n)} + \dfrac{3}{f'(y_n)}\right). \end{cases} \tag{1}$$

The without memory iteration (1) reads the error equation $e_{n+1} = \frac{2c_2^2}{3}e_n^3 + O(e_n^4)$, where $e_n = x_n - \alpha$ and $c_k = \left(\frac{1}{k!}\right)\frac{f^{(k)}(\alpha)}{f'(\alpha)}$, $k \geq 2$. Therefore, its efficiency index is 1.442. We herein precisely aim at using three evaluations per cycle, i.e. two evaluations of the first order derivative and one evaluation of the function to reach the order of convergence four. Motivated by the scheme (1) and the use of weight function, we consider the following iteration scheme

$$\begin{cases} y_n = x_n - \dfrac{2}{3}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{f(x_n)}{4}\left(\dfrac{1}{f'(x_n)} + \dfrac{3}{f'(y_n)}\right)(G(t_n) + H(\tau_n)), \end{cases} \tag{2}$$

where $G(t_n)$ and $H(\tau_n)$ are two weight functions with $t_n = \frac{f'(y_n)}{f'(x_n)}$ and $\tau_n = \frac{f(x_n)}{f'(y_n)}$. Thus, the scheme (2) defines a new class of multipoint methods with two weight functions. To obtain the solution of any nonlinear equations by the new two-point without memory class, we must set a particular initial approximation x_0, ideally close to the simple root. In numerical mathematics, it is very essential to know the behavior of an approximate method. Therefore, we are about to prove the order of convergence of the new class. In fact, it is proven below that how we can obtain fourth-order optimal methods out of (2).

Theorem 1. *Let a sufficiently smooth function $f : D \subseteq \mathbb{R} \to \mathbb{R}$ has a simple root α in the open interval D. Then the class of methods without memory (2) can be of fourth-order convergence, when the weight functions $G(t_n)$ and $H(\tau_n)$ are chosen as discussed below.*

Proof. Let $e_n = x_n - \alpha$ be the error in the nth iterate. By using symbolic computation, writing Taylor's series expansion for any term of (2), and using $f(\alpha) = 0$, we have

$$f(x_n) = f'(\alpha)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)], \tag{3}$$

wherein $c_k = \left(\frac{1}{k!}\right)\frac{f^{(k)}(\alpha)}{f'(\alpha)}$, $k \geq 2$. Also for the first derivative of the function in the first step of our cycle, we have

$$f'(x_n) = f'(\alpha)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + O(e_n^4)]. \tag{4}$$

Using (3), (4) and the first step of (2), we have $x_n - \frac{2}{3}\frac{f(x_n)}{f'(x_n)} - \alpha = \frac{e_n}{3} + \frac{2c_2e_n^2}{3} - \frac{4}{3}(c_2^2 - c_3)e_n^3 + \frac{2}{3}(4c_2^3 - 7c_2c_3 + 3c_4)e_n^4 + O(e_n^5)$. Similarly by Taylor's series expanding we have

$$\frac{1}{f'(x_n)} + \frac{3}{f'(y_n)} = \frac{4}{f'(\alpha)} - \frac{4c_2e_n^1}{f'(\alpha)} + \frac{4(c_2^2 - 3c_3)e_n^2}{3f'(\alpha)} + \frac{4(10c_2^3 + 3c_2c_3 - 10c_4)}{9f'(\alpha)}e_n^3 + O(e_n^4). \tag{5}$$

Furthermore using (5), we get that

$$x_n - \frac{f(x_n)}{4}\left(\frac{1}{f'(x_n)} + \frac{3}{f'(y_n)}\right) = \alpha + \frac{2c_2^2e_n^3}{3} + \frac{1}{9}(-13c_2^3 + 15c_2c_3 + c_4)e_n^4 + O(e_n^5). \tag{6}$$

Again by Taylor's series expanding around the simple root in the last step and using (6), we have

$$\begin{aligned} x_n - \frac{f(x_n)}{4}\left(\frac{1}{f'(x_n)} + \frac{3}{f'(y_n)}\right)(G(t_n) + H(\tau_n)) - \alpha &= (1 - G(1) - H(0))e_n^1 + \left(\frac{4}{3}c_2G'(1) - H'(0)\right)e_n^2 \\ &+ \frac{1}{18}(48c_3G'(1) - 6c_2H'(0) + 4c_2^2(3(G(1) + H(0) - 6G'(1)) - 4G''(1)) - 9H''(0))e_n^3 + \frac{1}{162}(360c_2^2H'(0) \\ &+ 18c_2(15c_3(G(1) + H(0) - 8G'(1)) - 32c_3G''(1) - 3H''(0)) + c_2^3(-234G(1) - 234H(0) + 1584G'(1) \\ &+ 864G''(1) + 64G^{(3)}(1)) + 3(6c_4(G(1) + H(0)) + 208c_4G'(1) - 9(4c_3H'(0) + H^{(3)}(0))))e_n^4 + O(e_n^5). \end{aligned} \tag{7}$$

This reveals that the weight functions in (2) should be chosen as comes next to make the order optimal

$$\begin{cases} G(1)=1, \quad G'(1)=0, \quad G''(1)=\dfrac{3}{4}, \quad |G^{(3)}(1)|<\infty, \\ H(0)=H'(0)=H''(0)=0, \quad |H^{(3)}(0)|<\infty. \end{cases} \tag{8}$$

Thus using (8), we can have the following general error equation, which reveals the fourth-order convergence for our contributed class

$$e_{n+1}=\left(-c_2c_3+\frac{c_4}{9}+\frac{1}{81}c_2^3(207+32G^{(3)}(1))-\frac{1}{6}H^{(3)}(0)\right)e_n^4+O(e_n^5). \tag{9}$$

This concludes the proof. □

It is obvious that our novel class of iterations requires three evaluations per iteration, i.e. two first order derivative and one function evaluations. Thus, it is a new Jarratt-type optimal scheme, which agrees with the still unproved conjecture of Kung and Traub [8]. We herein also pull the attention toward an open problem in root finding that reveals: *no* iterative without memory three-step method with two evaluations of the function and two evaluations of the first order derivative, which reaches the convergence order eight has been contributed in the literature.

Now by choosing appropriate weight functions as presented in (8), one can give optimal two-step methods, such as

$$\begin{cases} y_n=x_n-\dfrac{2}{3}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1}=x_n-\dfrac{f(x_n)}{4}\left(\dfrac{1}{f'(x_n)}+\dfrac{3}{f'(y_n)}\right)\left(1+\dfrac{3}{8}\left(\dfrac{f'(y_n)}{f'(x_n)}-1\right)^2-\dfrac{69}{64}\left(\dfrac{f'(y_n)}{f'(x_n)}-1\right)^3+\left(\dfrac{f(x_n)}{f'(y_n)}\right)^4\right), \end{cases} \tag{10}$$

where $e_{n+1}=(-c_2c_3+\frac{c_4}{9})e_n^4+O(e_n^5)$ is its error equations. We can also have the following efficient solver

$$\begin{cases} y_n=x_n-\dfrac{2}{3}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1}=x_n-\dfrac{f(x_n)}{4}\left(\dfrac{1}{f'(x_n)}+\dfrac{3}{f'(y_n)}\right)\left(1+\dfrac{3}{8}\left(\dfrac{f'(y_n)}{f'(x_n)}-1\right)^2+\left(\dfrac{1}{81}\right)\left(\dfrac{f(x_n)}{f'(y_n)}\right)^3\right), \end{cases} \tag{11}$$

with the following error equation: $e_{n+1}=\left(-\frac{1}{81}+\frac{23c_2^3}{9}-c_2c_3+\frac{c_4}{9}\right)e_n^4+O(e_n^5)$.

Remark 1. Choosing suitable weight functions in the last step of (2) will result in the follow-up iteration

$$\begin{cases} y_n=x_n-\dfrac{2}{3}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1}=x_n-\dfrac{f(x_n)}{4}\left(\dfrac{1}{f'(x_n)}+\dfrac{3}{f'(y_n)}\right)\left(1+\dfrac{3}{8}\left(\dfrac{f'(y_n)}{f'(x_n)}-1\right)^2-\dfrac{21}{32}\left(\dfrac{f'(y_n)}{f'(x_n)}-1\right)^3+\left(\dfrac{f(x_n)}{f'(y_n)}\right)^4\right), \end{cases} \tag{12}$$

which is in fact the fourth-order Jarratt method, [9]. Note that the fourth-order Jarratt method is defined by

$$\begin{cases} y_n=x_n-\dfrac{2}{3}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1}=x_n-\dfrac{3f'(y_n)+f'(x_n)}{6f'(y_n)-2f'(x_n)}\dfrac{f(x_n)}{f'(x_n)}. \end{cases} \tag{13}$$

Both schemes (12) and (13) read the same error equation as follows: $e_{n+1}=\left(c_2^3-c_2c_3+\frac{c_4}{9}\right)e_n^4+O(e_n^5)$. Therefore, Jarratt well known without memory iteration is a special case from our suggested class. This shows the generality of the proposed class as well.

In terms of computational point of view (if we suppose all functions and derivative evaluations have the same computational cost [16]), the efficiency index of our class of derivative-involved without memory methods (2) is 1.587, which is greater than that of Newton's and Steffensen's, i.e. 1.414, and the same with Jarratt method and King's family.

We here note that we tried to obtain the class (2), as efficiently as possible by adding a few number of arithmetic evaluations over (1). In fact, (2) can further be improved to be a more general class. That is, we can also suggest the following very general class

$$\begin{cases} y_n=x_n-\dfrac{2}{3}\dfrac{f(x_n)}{f'(x_n)}\{L(\rho_n)\}, \\ x_{n+1}=x_n-\dfrac{f(x_n)}{4}\left(\dfrac{1}{f'(x_n)}+\dfrac{3}{f'(y_n)}\right)\{G(t_n)+H(\tau_n)\}, \end{cases} \tag{14}$$

Table 1
Interesting choices of $L''(0)$, $G^{(3)}(1)$ and $H^{(3)}(0)$ in (16), which provide efficient optimal root solvers based on the structure (14).

Method	$L''(0)$	$G^{(3)}(1)$	$H^{(3)}(0)$	Error equation
1	0	0	0	$e_{n+1}=\frac{1}{9}(23c_2^3-9c_2c_3+c_4)e_n^4+O(e_n^5)$
2	1	1	1	$e_{n+1}=\left(-\frac{1}{6}+\frac{239c_2^3}{81}-c_2c_3+\frac{c_4}{9}\right)e_n^4+O(e_n^5)$
3	1	$-\frac{207}{32}$	0	$e_{n+1}=\left(-\frac{c_2}{2}-c_2c_3+\frac{c_4}{9}\right)e_n^4+O(e_n^5)$
4	-2	$-\frac{207}{32}$	0	$e_{n+1}=\left(c_2-c_2c_3+\frac{c_4}{9}\right)e_n^4+O(e_n^5)$
5	-2	$-\frac{207}{32}$	6	$e_{n+1}=\left(-1+c_2-c_2c_3+\frac{c_4}{9}\right)e_n^4+O(e_n^5)$

where $G(t_n)$, $H(\tau_n)$ and $L(\rho_n)$ are three real valued weight functions with $t_n=\frac{f'(y_n)}{f'(x_n)}$ and $\tau_n=\frac{f(x_n)}{f'(y_n)}$ and $\rho_n=\frac{f(x_n)}{f'(x_n)}$. It is proven below that how we can obtain fourth-order optimal methods out of (14).

Theorem 2. *Let a sufficiently smooth function $f: D\subseteq\mathbb{R}\to\mathbb{R}$ has a simple root α in the open interval D. Then the class of methods without memory (14) can be of fourth-order convergence, when the weight functions $G(t_n)$, $H(\tau_n)$ and $L(\rho_n)$ have the following conditions:*

$$\begin{cases} L(0)=1, \quad L'(0)=0, \quad |L''(0)|<\infty, \\ G(1)=1-H(0), \quad G'(1)=0, \quad G''(1)=\frac{3}{4}, \quad |G^{(3)}(1)|<\infty, \\ H'(0)=H''(0)=0, \quad |H^{(3)}(0)|<\infty. \end{cases} \tag{15}$$

Proof. The proof of this theorem is similar to the proof of Theorem 1. Therefore, we only give its error equation

$$e_{n+1}=\left(\frac{c_4}{9}-\frac{1}{2}c_2(2c_3+L''(0))+\frac{1}{81}c_2^3(207+32G^{(3)}(1))-\frac{1}{6}H^{(3)}(0)\right)e_n^4+O(e_n^5). \tag{16}$$

This shows that (14)–(15) reaches the convergence rate four using only three evaluations. And it is observable that (16) is more general than (9). Thus, the proof is complete. □

Other than the efficient methods (10) and (11) of optimal local order four in the sense of Kung–Traub, many more two-step without memory iterations can be constructed, i.e. by suitably changing (15) in (14). Thus now, in order to save the space and also give some of the other such optimal fourth-order methods according to (14)–(16), we list the interesting ones in Table 1. In Table 1, the last column gives the error equations obtained by varying the three factors involved in (16).

3. Numerical reports for simple roots

To demonstrate the performance of the new fourth-order methods, we take many particular nonlinear scalar equations for comparison. They are shown in Table 2. The simple zeros of each of them are listed in front of the nonlinear test functions. We shall determine the consistency and stability of results by examining the convergence of the new iterative without memory methods. The findings are generalized by illustrating the effectiveness of the quartic methods for determining the simple root of a nonlinear equation.

Now, we shortly mention some of the existing methods for comparison. Steffensen's second-order method is defined by

$$x_{n+1}=x_n-\frac{f(x_n)}{f[x_n,w_n]}, \quad w_n=x_n+f(x_n). \tag{17}$$

Derivative-free uni-parametric family ($\beta\in\mathbb{R}-\{0\}$) of Kung–Traub [8] could be given as comes next

$$\begin{cases} y_n=x_n-\dfrac{f(x_n)}{f[x_n,w_n]}, \quad w_n=x_n+\beta f(x_n), \\ x_{n+1}=y_n-\dfrac{f(y_n)f(w_n)}{(f(w_n)-f(y_n))f[x_n,y_n]}. \end{cases} \tag{18}$$

For a comparison here, we have chosen the Steffensen's scheme (17), Kung–Traub fourth-order family with $\beta=0.01$ (18), the quartic method of Jarratt (13), and the fourth-order methods (10) and (11). Note again that (13) is an element of the contributed class (2) as discussed in Remark 1. The comparison results are given in Table 3 in terms of the number of significant digits for each test function after the specified number of iterations, that is, e.g. 0.5e−173 shows that the absolute value of the given nonlinear function f_1 after four full iterations is zero up to 173 decimal places. The stopping criterion is $|f(x_n)|<1.E-1200$. We employed the computer algebra system MATLAB 7.6 with multiple-precision arithmetic. We observe from Table 3 that the two-point methods (10) and (11) produce approximations of good accuracy compared to the one- and two-point methods of different orders. Regarding these two methods, it is evident that the new class (2) gives acceptable accurate approximations in the tested examples. We used F as the notation for Failure, when the root solver for

Table 2
The test examples in this study for simple root case.

Test functions	Simple zeros
$f_1 = (\sin x)^2 + x$	$\alpha_1 = 0$
$f_2 = (1 + x) + \cos\left(\frac{\pi x}{2}\right) - \sqrt{1 - x^2}$	$\alpha_2 \approx -0.728584046444826\ldots$
$f_3 = (\sin x)^2 - x^2 + 1$	$\alpha_3 \approx 1.404491648215341\ldots$
$f_4 = e^{-x} + \sin(x) - 2$	$\alpha_4 \approx -1.0541271240912128\ldots$
$f_5 = xe^{-x} - 0.1$	$\alpha_5 \approx 0.111832559158963\ldots$
$f_6(x) = x^5 + x^3 - 1$	$\alpha_6 \approx 0.837619774826962\ldots$
$f_7(x) = \sqrt{x^2 + 2x + 5} - 2\sin(x) - x^2 + 3$	$\alpha_7 = 2.33196765588396401\ldots$
$f_8(x) = \sin^{-1}(x^2 - 1) - \frac{x}{2} + 1$	$\alpha_8 \approx 0.594810968398369\ldots$
$f_9(x) = \left(\sin(x) - \frac{\sqrt{2}}{2}\right)(x + 1)$	$\alpha_9 \approx 0.785398163397448\ldots$
$f_{10}(x) = x - \sin(\cos(x)) + 1$	$\alpha_{10} \approx -0.1660390510510295\ldots$
$f_{11}(x) = x^5 + 17x$	$\alpha_{11} = 0$
$f_{12}(x) = \sin(x) + \cos(x) + x$	$\alpha_{12} \approx -0.45662470456763\ldots$
$f_{13}(x) = x^3 - x^2 - 2x - \cos(x) + 2$	$\alpha_{13} \approx 0.498542523582153\ldots$
$f_{14}(x) = \sqrt{x^3} + \sin(x) - 30$	$\alpha_{14} \approx 9.716501993365200\ldots$
$f_{15}(x) = \tan^{-1}(x^2 - x)$	$\alpha_{15} = 1$
$f_{16}(x) = \sin^{-1}(x^2) - 2x$	$\alpha_{16} = 0$

a specific guess, requires more number of evaluations to find the root, or diverges, or finds another root. In Table 3, in some cases (18) gives better results, this is a simple consequence, when we choose very small value for β in (18), as we have done and chosen $\beta = 0.01$. We have computed the zeros of each test nonlinear function by three different initial guesses to completely reveal that although methods of the same order and same structure have somewhat similar results, there is no winner among the methods compared under a fair comparison situation. For one guess, one iterative method is better while for another guess (for the same nonlinear function) another method is superior. In Table 3, IT and TNE stand for number of iterations and total number of (functional) evaluations, respectively.

4. Extension for multiple roots

Finding multiple roots for a given nonlinear equation deserves further attention and therefore in this section we aim at generalizing the scheme (2) for multiple roots. Let D be an interval of $\mathbb{R}$, and furthermore, let $f(x)$ be a function from D into $\mathbb{R}$, and $\alpha \in D$ be a zero of f, i.e. a point such that $f(\alpha) = 0$. The point α is said a *zero of multiplicity m* of $f(x)$, if there exists a real number $l \neq 0$ such that

$$\lim_{x \to \alpha} \frac{|f(x)|}{|x - \alpha|^m} = l. \tag{19}$$

It is important to find robust and efficient methods in the case of multiple roots. When the multiplicity of the root is given or approximated, then we should extend simple root solvers for multiple roots. Now we contribute one class of two-step Jarratt-type methods according to (2) for multiple roots. The obtained results are rather interesting in view of the fact that usual one-point or multipoint iterative methods e.g., Newton method, Jarratt method (12), Kung–Traub method (18) etc. show linear convergence in the case of multiple roots. For the purpose of demonstration, the iteration scheme (2) meant for simple zero can be extended for multiple zeros of nonlinear functions. Numerical experiments for various test equations in the case of multiple roots will be given in Section 5 and confirm the validity of convergence and asymptotic error constants for the developed methods.

We herein start by extending first the cubical method of Heun (1). Heun's method is a member $\beta = \frac{2}{3}$ of the Nedzhibov et al. [17] family of third-order methods:

$$x_{n+1} = x_n - \frac{2\beta - 1}{2\beta}\frac{f(x_n)}{f'(x_n)} - \frac{1}{2\beta}\frac{f(x_n)}{f'(x_n - \beta u(x_n))}, \quad \beta \neq 0. \tag{20}$$

Scheme (2) is an improvement of Heun's method (1) so that it becomes fourth-order. We now suggest the multiple root version of third-order Heun's method (1) as comes next:

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n + \dfrac{1}{4}m(m^2 + 2m - 4)\dfrac{f(x_n)}{f'(x_n)} - \dfrac{1}{4}m(m+2)^2\left(\dfrac{m}{m+2}\right)^m \dfrac{f(x_n)}{f'(y_n)}, \end{cases} \tag{21}$$

where its error equation is given by

$$e_{n+1} = \frac{2}{m^2(m+2)}C_1^2 e_n^3 + O(e_n^4), \tag{22}$$

with $C_j = \frac{m!}{(m+j)!}\frac{f^{(m+j)}(\alpha)}{f^{(m)}(\alpha)}$, $j \geq 1$. We summarize the extension and the proof of multiple form of (2) in the following theorem.

Table 3
Results of convergence under fair circumstances for different methods.

f	Guess		(17)	(18)	(13)	(10)	(11)
f_1		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	−0.2	$\lvert f\rvert$	0.8e−81	0.5e−126	0.8e−161	0.2e−124	0.3e−109
	0.4	$\lvert f\rvert$	0.1e−90	0.4e−122	0.9e−139	0.4e−180	0.2e−120
	0.2	$\lvert f\rvert$	0.4e−132	0.5e−177	0.2e−198	0.5e−257	0.5e−173
f_2		IT	9	4	4	4	4
		TNE	18	12	12	12	12
	−0.5	$\lvert f\rvert$	0.9e−409	0.4e−211	0.6e−193	0.8e−261	0.6e−192
	−0.1	$\lvert f\rvert$	F	0.1e−216	0.3e−91	0.3e−64	0.1e−77
	−0.4	$\lvert f\rvert$	0.6e−156	0.1e−216	0.7e−187	0.2e−155	0.1e−189
f_3		IT	9	4	4	4	4
		TNE	18	12	12	12	12
	1.7	$\lvert f\rvert$	0.6e−199	0.8e−166	0.9e−187	0.8e−211	0.1e−159
	1.1	$\lvert f\rvert$	0.1e−290	0.3e−111	0.5e−147	0.1e−83	0.3e−89
	1.6	$\lvert f\rvert$	0.1e−298	0.1e−203	0.1e−226	0.1e−291	0.4e−196
f_4		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	−1.1	$\lvert f\rvert$	0.2e−329	0.1e−355	0.5e−388	0.9e−420	0.4e−343
	−0.7	$\lvert f\rvert$	0.6e−140	0.1e−100	0.1e−140	0.6e−55	0.2e−75
	−1	$\lvert f\rvert$	0.1e−319	0.2e−330	0.1e−364	0.1e−367	0.3e−316
f_5		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	−0.1	$\lvert f\rvert$	0.5e−114	0.1e−165	0.2e−199	0.2e−241	0.1e−157
	0	$\lvert f\rvert$	0.1e−178	0.3e−225	0.2e−265	0.1e−356	0.6e−220
	0.15	$\lvert f\rvert$	0.3e−289	0.4e−337	0.4e−327	0.1e−367	0.1e−324
f_6		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	0.7	$\lvert f\rvert$	F	0.1e−118	0.2e−168	0.3e−92	0.6e−100
	0.9	$\lvert f\rvert$	0.3e−69	0.6e−230	0.1e−270	0.5e−319	0.1e−223
	1.3	$\lvert f\rvert$	F	0.5e−52	0.3e−78	0.2e−92	0.8e−52
f_7		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	2	$\lvert f\rvert$	0.1e−333	0.4e−325	0.2e−389	0.3e−195	0.2e−281
	2.6	$\lvert f\rvert$	0.2e−364	0.4e−321	0.1e−318	0.1e−226	0.1e−293
	2.9	$\lvert f\rvert$	0.4e−282	0.5e−244	0.8e−229	0.1e−104	0.2e−213
f_8		IT	9	4	4	4	4
		TNE	18	12	12	12	12
	0.9	$\lvert f\rvert$	0.7e−307	0.6e−263	0.2e−246	0.1e−184	0.1e−284
	1.3	$\lvert f\rvert$	F	0.1e−172	0.7e−102	0.1e−77	0.6e−124
	0.3	$\lvert f\rvert$	0.2e−487	0.6e−275	0.5e−284	0.1e−201	0.4e−287
f_9		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	0.3	$\lvert f\rvert$	F	0.2e−132	0.2e−146	0.8e−169	0.3e−131
	1.4	$\lvert f\rvert$	F	0.9e−80	0.4e−95	0.2e−72	0.4e−70
	0.6	$\lvert f\rvert$	0.1e−340	0.4e−277	0.1e−285	0.1e−259	0.7e−291
f_{10}		IT	9	4	4	4	4
		TNE	18	12	12	12	12
	0.6	$\lvert f\rvert$	0.1e−216	0.1e−164	0.8e−193	0.6e−55	0.1e−175
	−0.8	$\lvert f\rvert$	0.6e−92	0.1e−86	0.5e−111	0.2e−31	0.3e−68
	0.3	$\lvert f\rvert$	0.3e−313	0.9e−209	0.2e−221	0.4e−133	0.1e−208
f_{11}		IT	8	3	4	4	4
		TNE	16	9	12	12	12
	−0.3	$\lvert f\rvert$	F	0.8e−477	0.9e−601	0.2e−326	0.6e−285
	−0.8	$\lvert f\rvert$	F	0.1e−177	0.1e−470	0.4e−55	0.2e−178
	0.2	$\lvert f\rvert$	0.1e−87	0.5e−605	0.6e−710	0.1e−436	0.8e−349
f_{12}		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	−0.9	$\lvert f\rvert$	0.2e−203	0.1e−325	0.2e−334	0.4e−148	0.1e−244
	−1	$\lvert f\rvert$	0.1e−148	0.4e−296	0.8e−300	0.1e−114	0.2e−226
	−0.2	$\lvert f\rvert$	0.2e−234	0.4e−297	0.4e−308	0.2e−233	0.4e−279

(*continued on next page*)

Table 3 (*continued*)

f	Guess		(17)	(18)	(13)	(10)	(11)
f_{13}		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	0	$\|f\|$	0.3e−158	0.5e−297	0.7e−232	0.3e−107	0.1e−217
	0.3	$\|f\|$	0.1e−269	0.2e−232	0.1e−237	0.1e−269	0.5e−227
	0.4	$\|f\|$	0.6e−352	0.7e−295	0.3e−302	0.4e−329	0.4e−289
f_{14}		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	10.5	$\|f\|$	0.5e−127	0.1e−279	0.1e−242	0.1e−49	0.3e−183
	9	$\|f\|$	0.1e−194	0.1e−310	0.2e−316	0.1e−69	0.2e−196
	9.6	$\|f\|$	0.1e−386	0.4e−461	0.2e−455	0.8e−351	0.4e−394
f_{15}		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	1.6	$\|f\|$	F	0.8e−97	0.6e−104	0.2e−59	0.2e−93
	1.3	$\|f\|$	0.4e−240	0.4e−150	0.7e−165	0.3e−247	0.4e−154
	0.9	$\|f\|$	0.1e−162	0.4e−208	0.6e−237	0.2e−244	0.2e−198
f_{16}		IT	8	4	4	4	4
		TNE	16	12	12	12	12
	−0.6	$\|f\|$	0.1e−94	0.1e−145	0.1e−161	0.5e−158	0.1e−129
	−0.1	$\|f\|$	0.9e−327	0.7e−315	0.1e−337	0.1e−652	0.1e−303
	0.2	$\|f\|$	0.7e−267	0.4e−216	0.7e−243	0.6e−477	0.1e−201

Theorem 3. *Let $f : D \subseteq \mathbb{R} \to \mathbb{R}$ be a function for an open interval $D \subseteq \mathbb{R}$. Let $f(x)$ has a multiple root, say $\alpha \subseteq D$ with multiplicity $m > 1$ and x_0 is an initial guess to the multiple root. Assume that f is sufficiently differentiable in D, then the iterative scheme*

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n + \left(\dfrac{1}{4}m(m^2+2m-4)\dfrac{f(x_n)}{f'(x_n)} - \dfrac{1}{4}m(m+2)^2\left(\dfrac{m}{m+2}\right)^m \dfrac{f(x_n)}{f'(y_n)}\right)(G(t_n)+H(\tau_n)), \end{cases} \tag{23}$$

will have local fourth-order convergence in the vicinity of α, if the weight functions in (23) be chosen as discussed below. Note that again $G(t_n)$ and $H(\tau_n)$ are two (real valued) weight functions with $t_n = \frac{f'(y_n)}{f'(x_n)}$ and $\tau_n = \frac{f(x_n)}{f'(y_n)}$.

Proof. Since $f(x)$ is a sufficiently differentiable function, therefore expanding $f(x_n)$ around $x = \alpha$ by Taylor's expansion and using $f(\alpha) = f'(\alpha) = \cdots = f^{(m-1)}(\alpha) = 0$ and $f^{(m)}(\alpha) \neq 0$ (a condition for $x = \alpha$ to be a root of multiplicity m), we have

$$f(x_n) = \frac{f^{(m)}(\alpha)}{m!}e_n^m\left(1 + \sum_{j=1}^{\infty} C_j e_n^j\right), \tag{24}$$

and also $f'(x_n) = \frac{f^{(m)}(\alpha)}{(m-1)!}e_n^{m-1}\left(1 + \sum_{j=1}^{\infty}\frac{m+j}{m}C_j e_n^j\right)$, so that using symbolic calculations, we have

$$x_n - \frac{2m}{m+2}\frac{f(x_n)}{f'(x_n)} - \alpha = \frac{m}{m+2}e_n + \frac{2C_1}{m(m+2)}e_n^2 - \frac{((m+1)C_1^2 - 2mC_2)}{m^2(m+2)}e_n^3 + \frac{2((-3m^2-4m)C_1C_2 + (m^2+2m+1)C_1^3 + 3m^2C_3)}{m^3(m+2)}e_n^4 + O(e_n^5), \tag{25}$$

and

$$x_n + \left(\frac{1}{4}m(m^2+2m-4)\frac{f(x_n)}{f'(x_n)} - \frac{1}{4}m(m+2)^2\left(\frac{m}{m+2}\right)^m\frac{f(x_n)}{f'(y_n)}\right) - \alpha = \frac{2}{m^2(m+2)}C_1^2e_n^3 + \left(\frac{(3m^5+48m^2-12m^4+12m^3)C_1C_2 + (m^5+2m^3+6m^4-16m^2-24m-8)C_1^3}{3m^4(m+2)} + \frac{mC_3}{(m+2)}\right)e_n^4 + O(e_n^5), \tag{26}$$

which results in the cubical method of Heun. We now consider the achievement of quartic convergence according to (23) with appropriate weight function in it. Using Taylor expansion yields

$$t_n = \frac{f'(y_n)}{f'(x_n)} = p^{m-1} - \frac{4p^m}{m^3}C_1e_n - \left(\frac{4(m^2+2)p^m}{m^5}C_1^2 - \frac{8p^m}{m^3}C_2\right)e_n^2 + \left(-\frac{8(m^4-m^3+5m^2+m+6)p^m}{3m^7}C_1^3 + \frac{8(m^2+4)p^m}{m^5}C_1C_2 - \frac{8(m^2+6m+6)p^m}{m^3(m+2)}C_3\right)e_n^3 + O(e_n^4), \tag{27}$$

where $u = p^{m-1}$, $t_n = u + v_n$ and $p = \frac{m}{m+2}$. Then, the remainder $v_n = t_n - u$ is infinitesimal with the same of the order of e_n. Thus, we can perform a Taylor expansion around u, [19], so that

$$G(t_n) = G(u) + G'(u)v_n + \frac{1}{2}G''(u)v_n^2 + \frac{1}{3!}G'''(u)v_n^3 + O(e_n^4). \tag{28}$$

Similarly, a Taylor expansion yields

$$\tau_n = \frac{f(x_n)}{f'(y_n)} = \frac{1}{(m+2)p^m}e_n - \frac{(m^2+2m-4)}{m^2(m+2)^2p^m}C_1e_n^2 + \left(\frac{(m^4+5m^3+4m^2-8m-16)}{m^3(m+2)^3p^m}C_1^2 - \frac{2(m^2+2m-4)}{m^2(m+2)^2p^m}C_2\right)e_n^3 + O(e_n^4). \tag{29}$$

τ_n is infinitesimal with the same order of e_n and we can perform a Taylor expansion around 0 so that

$$H(\tau_n) = H(0) + H'(0)\tau_n + \frac{1}{2}H''(0)\tau_n^2 + \frac{1}{3!}H'''(0)\tau_n^3 + O(e_n^4). \tag{30}$$

Using Eqs. (26) and (28)–(30) and the last step of (23) ends in

$$e_{n+1} = x_{n+1} - \alpha = (1 - G(u) - H(0))e_n + \left(\frac{4p^mC_1}{m^3}G'(u) - \frac{1}{(m+2)p^m}H'(0)\right)e_n^2 + \left(\frac{m^2+2m-4}{m^2(m+2)^2p^m}H'(0)C_1 + \frac{8p^m}{m^3}G'(u)C_2 - \frac{1}{2(m+2)^2p^{2m}}H''(0) + \frac{2C_1^2}{m^6(m+2)} \times \left\{-2p^m(m+2)(m^2+2)G'(u) - 4p^{2m}(m+2)G''(u) + m^4(H(0)+G(u))\right\}\right)e_n^3 + O(e_n^4). \tag{31}$$

To make the order optimal in eq. (31), we choose

$$\begin{cases} G(u) = 1, \quad G'(u) = 0, \quad G''(u) = \dfrac{m^4}{4(m+2)p^{2m}}, \quad |G'''(u)| < \infty, \\ H(0) = H'(0) = H''(0) = 0, \quad |H'''(0)| < \infty. \end{cases} \tag{32}$$

Therefore, we attain the following error equation of scheme (23) for multiple roots:

$$e_{n+1} = \left[\frac{mC_3}{(m+2)^2} - \frac{C_1C_2}{m} - \frac{H'''(0)}{6(m+2)^3p^{3m}} + \frac{C_1^3}{3m^4} \times \left(\frac{32G'''(u)p^{3m}}{m^5} + \frac{m^5+6m^4+14m^3+8m^2+40}{(m+2)^2}\right)\right]e_n^4 + O(e_n^5). \tag{33}$$

This concludes the proof for the quartic convergence of the general multiple root-finder (23). As can be observed from the iteration (23), it is a general class of two-step two-point methods without memory consuming three evaluations per computing step, i.e. one function and two first derivative evaluations. This also reveals the consistency of our contribution (23) with the conjecture of Kung–Traub. □

The simple choices of $G(t_n)$ and $H(\tau_n)$, which satisfy (32) are

$$G(t_n) = 1 + \frac{m^4}{8(m+2)p^{2m}}(t_n - u)^2, \quad \text{and} \quad H(\tau_n) = \tau_n^3. \tag{34}$$

Thus, the multiple root version of scheme (10) is given by

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n + \left(\dfrac{1}{4}m(m^2+2m-4)\dfrac{f(x_n)}{f'(x_n)} - \dfrac{1}{4}m(m+2)^2\left(\dfrac{m}{m+2}\right)^m \dfrac{f(x_n)}{f'(y_n)}\right) \\ \quad \times \left(1 + \dfrac{m^4}{8(m+2)\left(\frac{m}{m+2}\right)^{2m}}\left(\dfrac{f'(y_n)}{f'(x_n)} - \left(\dfrac{m}{m+2}\right)^{m-1}\right)^2 - \dfrac{69}{64}\left(\dfrac{f'(y_n)}{f'(x_n)} - \left(\dfrac{m}{m+2}\right)^{m-1}\right)^3 + \left(\dfrac{f(x_n)}{f'(y_n)}\right)^4\right), \end{cases} \tag{35}$$

with the following error equation

$$e_{n+1} = \left[\frac{mC_3}{(m+2)^2} - \frac{C_1C_2}{m} + \frac{C_1^3}{3m^4}\left(\frac{207\left(\frac{m}{m+2}\right)^{3m}}{m^5} + \frac{m^5+6m^4+14m^3+8m^2+40}{(m+2)^2}\right)\right]e_n^4 + O(e_n^5). \tag{36}$$

The multiple root version of scheme (11) is presented by

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n + \left(\dfrac{1}{4}m(m^2+2m-4)\dfrac{f(x_n)}{f'(x_n)} - \dfrac{1}{4}m(m+2)^2\left(\dfrac{m}{m+2}\right)^m \dfrac{f(x_n)}{f'(y_n)}\right) \\ \quad \times \left(1 + \dfrac{m^4}{8(m+2)\left(\frac{m}{m+2}\right)^{2m}}\left(\dfrac{f'(y_n)}{f'(x_n)} - \left(\dfrac{m}{m+2}\right)^{m-1}\right)^2 + \dfrac{1}{81}\left(\dfrac{f(x_n)}{f'(y_n)}\right)^3\right), \end{cases} \tag{37}$$

with error equation

$$e_{n+1} = \left[\frac{mC_3}{(m+2)^2} - \frac{C_1C_2}{m} - \frac{1}{81(m+2)^3\left(\frac{m}{m+2}\right)^{3m}} + \frac{C_1^3}{3m^4}\left(\frac{m^5+6m^4+14m^3+8m^2+40}{(m+2)^2}\right)\right]e_n^4 + O(e_n^5). \tag{38}$$

The multiple root version of scheme (12) can be constructed as comes next

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n + \left(\dfrac{1}{4}m(m^2+2m-4)\dfrac{f(x_n)}{f'(x_n)} - \dfrac{1}{4}m(m+2)^2\left(\dfrac{m}{m+2}\right)^m \dfrac{f(x_n)}{f'(y_n)}\right) \\ \quad \times \left(1 + \dfrac{m^4}{8(m+2)\left(\frac{m}{m+2}\right)^{2m}}\left(\dfrac{f'(y_n)}{f'(x_n)} - \left(\dfrac{m}{m+2}\right)^{m-1}\right)^2 - \dfrac{21}{32}\left(\dfrac{f'(y_n)}{f'(x_n)} - \left(\dfrac{m}{m+2}\right)^{m-1}\right)^3 + \left(\dfrac{f(x_n)}{f'(y_n)}\right)^4\right), \end{cases} \tag{39}$$

where its error relation is

$$e_{n+1} = \left[\frac{mC_3}{(m+2)^2} - \frac{C_1C_2}{m} + \frac{C_1^3}{3m^4}\left(-\frac{126\left(\frac{m}{m+2}\right)^{3m}}{m^5} + \frac{m^5+6m^4+14m^3+8m^2+40}{(m+2)^2}\right)\right]e_n^4 + O(e_n^5). \tag{40}$$

It should be reminded an open problem in multipoint iterations without memory reveals that in order to obtain the quartic order for multiple roots, one may use two first-order derivative and one function evaluations per full cycle, i.e. no iterative without memory scheme with optimal order four for multiple roots with two function and one first-order derivative has yet been contributed in the literature.

5. Numerical reports for multiple roots

Herein also, all numerical computations have been carried out in a MATLAB 7.6 environment using 1200-digit floating-point arithmetic. The following test problems have been used with the stopping criterion $|f(x_n)| < 1.E - 1200$, where α is a root of $f(x)$ with multiplicity m now. The test instances in this case are given in Table 4.

In [18], the authors developed an optimal multiple root solver as comes next

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{m}{8}\left((m^3-4m+8) - (m+2)^2\left(\dfrac{m}{m+2}\right)^m \dfrac{f'(x_n)}{f'(y_n)} \right. \\ \quad \left. \times \left(2(m-1) - (m+2)\left(\dfrac{m}{m+2}\right)^m \dfrac{f'(x_n)}{f'(y_n)}\right)\right)\dfrac{f(x_n)}{f'(x_n)}. \end{cases} \tag{41}$$

Table 4
The test examples for multiple roots comparison.

Test functions	Multiplicity	Multiple zeros
$f_1 = ((\sin x)^2 + x)^5$	5	$\alpha_1 = 0$
$f_2 = \left((1+x) + \cos\left(\frac{\pi x}{2}\right) - \sqrt{1-x^2}\right)^3$	3	$\alpha_2 \approx -0.728584046444826\ldots$
$f_3 = ((\sin x)^2 - x^2 + 1)^4$	4	$\alpha_3 \approx 1.404491648215341\ldots$
$f_4 = (e^{-x} + \sin(x) - 2)^2$	2	$\alpha_4 \approx -1.0541271240912128\ldots$

Table 5
Results of convergence under fair circumstances for different multiple root solvers.

f	Guess		(41)	(42)	(35)	(37)
f_1	0.3	$\|f(x_1)\|$	0.9e−11	0.9e−11	0.4e−11	0.7e−12
		$\|f(x_2)\|$	0.1e−41	0.2e−41	0.9e−43	0.2e−46
		$\|f(x_3)\|$	0.1e−164	0.3e−118	0.1e−169	0.2e−184
		$\|f(x_4)\|$	0.1e−656	0.3e−195	0.1e−589	0.2e−634
f_1	0.2	$\|f(x_1)\|$	0.2e−13	0.2e−13	0.1e−13	0.3e−14
		$\|f(x_2)\|$	0.6e−59	0.1e−51	0.6e−53	0.1e−55
		$\|f(x_3)\|$	0.7e−206	0.1e−128	0.4e−210	0.9e−222
		$\|f(x_4)\|$	0.7e−822	0.1e−205	0.2e−711	0.2e−746
f_2	−0.6	$\|f(x_1)\|$	0.1e−9	0.1e−9	0.1e−9	0.1e−10
		$\|f(x_2)\|$	0.6e−38	0.7e−38	0.2e−38	0.5e−42
		$\|f(x_3)\|$	0.2e−151	0.4e−151	0.1e−152	0.1e−167
		$\|f(x_4)\|$	0.1e−604	0.9e−604	0.1e−610	0.4e−670
f_2	−0.8	$\|f(x_1)\|$	0.5e−9	0.5e−9	0.5e−9	0.4e−9
		$\|f(x_2)\|$	0.1e−35	0.2e−35	0.1e−35	0.4e−36
		$\|f(x_3)\|$	0.7e−142	0.3e−141	0.3e−142	0.5e−144
		$\|f(x_4)\|$	0.6e−567	0.1e−564	0.2e−568	0.7e−576
f_3	1.3	$\|f(x_1)\|$	0.8e−13	0.1e−12	0.7e−13	0.6e−14
		$\|f(x_2)\|$	0.2e−56	0.7e−56	0.9e−57	0.2e−62
		$\|f(x_3)\|$	0.2e−230	0.1e−228	0.2e−232	0.2e−265
		$\|f(x_4)\|$	0.1e−926	0.1e−919	0.1e−934	0.6e−1032
f_3	2	$\|f(x_1)\|$	0.1e−4	0.1e−4	0.3e−6	0.7e−6
		$\|f(x_2)\|$	0.1e−23	0.3e−23	0.3e−30	0.6e−60
		$\|f(x_3)\|$	0.3e−99	0.5e−98	0.2e−126	0.2e−126
		$\|f(x_4)\|$	0.3e−402	0.4e−397	0.1e−510	0.3e−512
f_4	−1	$\|f(x_1)\|$	0.3e−9	0.7e−9	0.3e−9	0.1e−8
		$\|f(x_2)\|$	0.4e−40	0.2e−38	0.6e−40	0.1e−36
		$\|f(x_3)\|$	0.1e−163	0.1e−156	0.6e−163	0.3e−149
		$\|f(x_4)\|$	0.1e−657	0.9e−629	0.6e−655	0.2e−599
f_4	−1.4	$\|f(x_1)\|$	0.1e−3	0.2e−3	0.2e−4	0.1e−2
		$\|f(x_2)\|$	0.1e−17	0.2e−16	0.8e−21	0.3e−13
		$\|f(x_3)\|$	0.4e−74	0.3e−68	0.1e−86	0.1e−55
		$\|f(x_4)\|$	0.1e−299	0.1e−275	0.4e−349	0.2e−224

This iteration requires three evaluations of the function and the same order as our contributed methods for multiple cases when the multiplicity of the zero is available. For the purpose of comparison, we also mention the very recent root solver which is optimal for multiple roots as comes next [19]:

$$\begin{cases} y_n = x_n - \dfrac{2m}{m+2}\dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \dfrac{m}{8}\left(m^3\left(\dfrac{m+2}{m}\right)^{2m}\left(\dfrac{f'(y_n)}{f'(x_n)}\right)^2 - 2m^2(m+3) \right. \\ \quad \left. \times \left(\frac{m+2}{m}\right)^m\left(\frac{f'(y_n)}{f'(x_n)}\right) + (m^3+6m^2+8m+8)\right)\frac{f(x_n)}{f'(x_n)}. \end{cases} \tag{42}$$

Now, we employ the presented multiple fourth-order class (23), including (35), and (37), to solve some nonlinear equations and compare them with multiple root solvers (41) and (42). From Table 5, we can see that, even with fewer iterative steps, the presented methods without memory for multiple roots can obtain high-precision solutions, and thus they are suitable for high-precision computation. The numerical results in Table 5 coincide with the theoretical discussion of Theorem 3.

6. Conclusions

Multipoint iterative methods have become an interesting and challenging task at the beginning of the 21st century. The highest possible computational efficiency of these methods is closely connected to the hypothesis of Kung and Traub from 1974, discussed in Section 1. In this paper, a general without memory class of two-step two-point derivative-involved methods has suggested for simple roots. The theoretical proof of the contribution was presented and it was seen that any method from the novel class reaches the convergence order 4 and subsequently the optimal efficiency index 1.587. The class is free from second order derivative. Furthermore, numerical examples were employed to find the approximate solutions of a lot of nonlinear scalar equations. The numerical results were also corroborating the underlying theory developed in this paper.

We then have extended our technique for multiple roots positively. That is to say, a general two-step class of multiple root finders with fourth-order using three evaluations has been contributed in Section 4. Consequently, when the multiplicity of the roots is available or estimated, one may use the derived multiple root solvers (35), (37) and (39) for achieving the aim. We end this paper by reminding another open problem in root-finding topic, which indicates *no* optimal eighth-order method in the sense of Kung–Traub using two evaluations of the function and two evaluations of the first order derivative, has yet been contributed. Trying to solve this open problem, or constructing quartic iterations for the systems of nonlinear equations according to (2) can be considered as future research in this active topic of study.

References

[1] L.D. Petkovic, M.S. Petkovic, A note on some recent methods for solving nonlinear equations, Applied Mathematics and Computation 185 (2007) 368–374.
[2] A. Iliev, N. Kyurkchiev, Nontrivial Methods in Numerical Analysis, in: Selected Topics in Numerical Analysis, Lambert Academic Publishing, 2010.
[3] P. Sargolzaei, F. Soleymani, Accurate fourteenth-order methods for solving nonlinear equations, Numerical Algorithms 58 (2011) 513–527.
[4] F. Soleymani, S. Karimi Vanani, Optimal Steffensen-type methods with eighth order of convergence, Computers and Mathematics with Applications (2011) doi:10.1016/j.camwa.2011.10.047.
[5] F. Soleymani, On a bi-parametric class of optimal eighth-order derivative-free methods, International Journal of Pure and Applied Mathematics 72 (2011) 27–37.
[6] F. Soleymani, M. Sharifi, On a general efficient class of four-step root-finding methods, International Journal of Mathematics and Computers in Simulation 5 (2011) 181–189.
[7] D.K.R. Babajee, Analysis of higher order variants of Newton's method and their applications to differential and integral equations and in ocean acidation, Ph.D. Thesis, University of Mauritius, December 2010.
[8] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, Journal of the ACM 21 (1974) 643–651.
[9] P. Jarratt, Some efficient fourth order multipoint methods for solving equations, BIT 9 (1969) 119–124.
[10] F. Soleymani, New optimal iterative methods in solving nonlinear equations, International Journal of Pure and Applied Mathematics 72 (2011) 195–202.
[11] F. Soleymani, S.K. Khattri, S. Karimi Vanani, Two new classes of optimal Jarratt-type fourth-order methods, Applied Mathematics Letters (2011) doi:10.1016/j.aml.2011.10.030.
[12] F. Soleymani, M. Sharifi, B.S. Mousavi, An improvement of Ostrowski's and King's techniques with optimal convergence order eight, Journal of Optimization Theory and Applications (2011) doi:10.1007/s10957-011-9929-9.
[13] F. Soleymani, S. Karimi Vanani, M. Khan, M. Sharifi, Some modifications of King's family with optimal eighth order of convergence, Mathematical and Computer Modelling (2011) doi:10.1016/j.mcm.2011.10.016.
[14] M. Grau-Sánchez, J.L. Díaz-Barrero, Zero-finder methods derived using Runge–Kutta techniques, Applied Mathematics and Computation 217 (2011) 5366–5376.
[15] K. Heun, Neue methode zur approximativen integration der differentialgleichungen einer unabhängigen variablen, Zeitschrift für Angewandte Mathematik und Physik ZAMP 45 (1900) 23–38.
[16] J.F. Traub, Iterative Methods for the Solution of Equations, Chelsea Publishing Company, New York, 1982.
[17] G.H. Nedzhibov, V.I. Hasanov, M.G. Petkov, On some families of multipoint iterative methods for solving nonlinear equations, Numerical Algorithms 42 (2006) 127–136.
[18] J.R. Sharma, R. Sharma, Modified Jarratt method for computing multiple roots, Applied Mathematics and Computation 217 (2010) 878–881.
[19] X. Zhou, X. Chen, Y. Song, Constructing higher-order methods for obtaining the multiple roots of nonlinear equations, Journal of Computational and Applied Mathematics (2011) doi:10.1016/j.cam.2011.03.014.