Thai Journal of Mathematics
Volume 12 (2014) Number 2 : 313–327
http://thaijmath.in.cmu.ac.th
ISSN 1686-0209

Some High-Order Iterative Methods for Finding All the Real Zeros

Fazlollah Soleymani

Department of Mathematics, Mashhad Branch
Islamic Azad University, Mashhad, Iran
e-mail : fazlollah.soleymani@gmail.com

Abstract : This paper concerns the numerical solution of one variable nonlinear equations. Some four-point four-step iterative schemes are given. The new methods are attained by a simple but powerful approximation of the first derivative of the function in the fourth step of our cycle, where the first three steps are any of the optimal derivative-involved eighth-order methods. Analytical proof of the main theorem is given to clarify the fourteenth-order convergence. The extension of one high-order method for multiple zeros will be given as well. A hybrid algorithm has also been proposed to extract all the real zeros of nonlinear functions in a given interval. Finally, we furnish numerical comparisons to attest the theoretical results and the fast rate of convergence.

Keywords : iterative methods; error equation; interval methods; computational load.
2010 Mathematics Subject Classification : 41A25.

1 Introduction

Assume that $f : I \subseteq \mathbb{R} \to \mathbb{R}$ is sufficiently smooth and $\alpha \in I$ is its simple zero. There are many high order iterative solvers in the literature. Almost all of them have some drawbacks and strong points. Some include derivative to proceed which called derivative-involved schemes [1] and some not; namely derivative-free methods [2]. In this situation, the measure of efficiency index in comparison of different multi-point schemes have totally been using in the literature. A multi-point scheme without memory [3] for solving nonlinear scalar equations possesses the efficiency index $\sqrt[n]{p}$, where p is the convergence rate and n is the number of (functional) evaluations per full iteration.

This paper studies a new class of four-step four-point methods for solving nonlinear equations. The class is derived from a well-done approximation of the first derivative of the function in the fourth step of a four-step cycle, at which the first three steps are any of the optimal derivative-involved eighth-order iterative methods. Per full cycle, the class adds only one more evaluation of the function to increase the order of convergence from eight to fourteen in contrast to the optimal eighth-order methods. As a result, the efficiency index of the class comes up from 1.68 to 1.69, to also ensure the users for its implementation.

In what follows, first in Section 2 a brief review on some of the common high-order methods in the literature is given. This section is followed by Section 3, whereas the main contribution is provided to boost up the convergence rate and the efficiency index of the existing eighth-order methods. Section 4 illustrates the accuracy of the new obtained four-point fourteenth-order methods by solving and comparing some nonlinear test functions. Section 5 will remind the well-known interval Newton's method as a tool for extracting initial approximations for the real zeros. Finally in Section 6, the conclusions will be drawn.

2 Pointers to the literature

Khattri and Abbasbandy in [4] presented a fourth-order scheme including two derivative- and one function-evaluation per cycle with one free parameter ($a \in \mathbb{R}$) as comes next:

$$\begin{cases} y_n = x_n - \frac{2}{3}\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \Big[1 + \left(\frac{21}{8} - a\right)\left(\frac{f'(y_n)}{f'(x_n)}\right)^1 + \left(-\frac{9}{2} - 3a\right)\left(\frac{f'(y_n)}{f'(x_n)}\right)^2 \\ \qquad\qquad + \left(\frac{15}{8} - 3a\right)\left(\frac{f'(y_n)}{f'(x_n)}\right)^3 + a\left(\frac{f'(y_n)}{f'(x_n)}\right)^4\Big]\frac{f(x_n)}{f'(x_n)}. \end{cases} \tag{2.1}$$

This technique has 1.58 as its efficiency index. Soleymani in [5] proposed a sixth-order variant of Jarratt method using four evaluations per iteration to reach the efficiency index 1.56, which is useful when the initial guesses are in the vicinity of the zeros but not so close

$$\begin{cases} y_n = x_n - \frac{2}{3}\frac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \frac{3f'(y_n)+f'(x_n)}{6f'(y_n)-2f'(x_n)}\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f'(y_n)+2f[z_n,x_n,x_n](z_n-y_n)}. \end{cases} \tag{2.2}$$

Cordero et al. [6] investigated a seventh-order technique consisting of four-evaluation per iteration to reach the index of efficiency 1.62 as follows

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = x_n + \frac{f(x_n)+f(y_n)}{f'(x_n)} - 2\frac{f(x_n)}{f'(x_n)}\frac{f(x_n)}{f(x_n)-f(y_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f[z_n,y_n]+f[z_n,x_n,x_n](z_n-y_n)}. \end{cases} \tag{2.3}$$

Liu and Wang [7] suggested some optimal eighth-order methods using four evaluations per full cycle ($\beta_1, \beta_2 \in \mathbb{R}$) in what follows

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \frac{f(y_n)}{f'(x_n)}\frac{4f(x_n)-f(y_n)}{4f(x_n)-9f(y_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)}\left[\frac{8f(y_n)}{4f(x_n)-11f(y_n)} + \left(1+\frac{f(z_n)}{3f(y_n)-\beta_1 f(z_n)}\right)^3 + \frac{4f(z_n)}{f(x_n)+\beta_2 f(z_n)}\right], \end{cases} \tag{2.4}$$

where the efficiency index is 1.68. [7] also suggested the following three-step approach ($\alpha_1, \alpha_2 \in \mathbb{R}$) with the same number of evaluations and efficiency index

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \frac{f(y_n)}{f'(x_n)}\frac{f(x_n)}{f(x_n)-2f(y_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)}\left[\left(\frac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)}\right)^2 + \frac{f(z_n)}{f(y_n)-\alpha_1 f(z_n)} + \frac{4f(z_n)}{f(x_n)+\alpha_2 f(z_n)}\right]. \end{cases} \tag{2.5}$$

Wang and Liu in [8] applied the method of weight functions to produce other eighth-order schemes ($\alpha \in \mathbb{R}$) as follows

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \frac{f(x_n)}{f'(x_n)}\frac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)}\left[1+\frac{4f(z_n)}{f(x_n)-\alpha f(z_n)}\right]\left[\frac{f(x_n)^2}{f(x_n)^2-2f(x_n)f(y_n)-f(y_n)^2} + \frac{f(z_n)}{f(y_n)}\right], \end{cases} \tag{2.6}$$

and

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \frac{f(x_n)}{f'(x_n)}\frac{4f(x_n)^2-5f(x_n)f(y_n)-f(y_n)^2}{4f(x_n)^2-9f(x_n)f(y_n)}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)}\left[1+4\frac{f(z_n)}{f(x_n)}\right]\left[\frac{8f(y_n)}{4f(x_n)-11f(y_n)} + 1 + \frac{f(z_n)}{f(y_n)}\right], \end{cases} \tag{2.7}$$

where the efficiency index is 1.68.

In such a situation, the concept of optimality, which was given by Kung-Traub in [9] for without memory methods is taken into account. To obtain a better background on iterative methods in zero-finding, we refer the readers to the works [10–13].

3 A new class of fourteenth-order methods

In this section, we give our main contribution by providing a novel class of iterative methods, which possesses the fourteenth-order convergence and 1.69 as the efficiency index. We fulfill our aim by taking an especial look at the computational complexity as well. The existing eighth-order methods have high complexity themselves, therefore for obtaining new schemes with better efficiency index and convergence rate, we should avoid providing more complexity to the techniques. This also ensures the users to apply such obtained techniques from the class as easier as possible. To achieve such a goal, we assume the following structure

$$\begin{cases} \text{any optimal 8th-order method} = \begin{cases} f(x_n) \text{ and } f'(x_n) \text{ are available,} \\ f(y_n) \text{ is available,} \\ f(z_n) \text{ is available,} \end{cases} \\ x_{n+1} = w_n - \frac{f(w_n)}{f'(w_n)}. \end{cases} \tag{3.1}$$

Note that we consider, y_n, z_n and, w_n as the outlets of the first, second and third sub-steps, respectively. It is clear that (3.1) is a sixteenth-order method with six evaluations per full iteration and reach the 1.58 as its efficiency index. Now the main challenge is to approximate $f'(w_n)$ as effectively as possible to achieve our goal, i.e. first the order goes up, second the efficiency index increases in contrast to the optimal eighth-order methods, and third no more computational burden has been applied to the attained techniques. Toward this aim, although we have five known data, namely $f(x_n)$, $f'(x_n)$, $f(y_n)$, $f(z_n)$ and $f(w_n)$, we only use the last three values of the function in the second, third and fourth steps of our cycle. We do this intentionally to reduce the computational load of the attained class only. Thus, for estimating $f'(w_n)$, we consider the following rational linear function

$$p(t) = f(y) + \frac{a + (t - y)}{b(t - y) + c}, \tag{3.2}$$

whence its derivative is in the following form $p'(t) = (c - ab)/(b(t - y) + c)^2$. (3.2) is inspired by Pade approximant. The Pade approximant is a rational function that can be viewed of as a generalization of the Taylor expansion. A rational function is the ratio of polynomials. Because these functions only use the elementary arithmetic operations, they are so easy to evaluate numerically. At this moment, the three unknown coefficients could be attained by substituting the known values in (3.2). That is, by satisfying $p(t)|_{y_n} = f(y_n)$, $p(t)|_{z_n} = f(z_n)$, $p(t)|_{w_n} = f(w_n)$, first we obtain that $a = 0$ and subsequently, we have

$$\begin{cases} b(w_n - y_n) + c = \frac{1}{f[y_n, w_n]}, \\ b(z_n - y_n) + c = \frac{1}{f[y_n, z_n]}, \end{cases} \tag{3.3}$$

where $f[y_n, w_n]$ and $f[y_n, z_n]$ are divided differences. Solving this easy problem gives us

$$\begin{cases} b = \frac{1}{w_n - z_n} \left(\frac{1}{f[y_n, w_n]} - \frac{1}{f[y_n, z_n]} \right), \\ c = \frac{1}{w_n - z_n} \left(\frac{y_n - z_n}{f[y_n, w_n]} - \frac{y_n - w_n}{f[y_n, z_n]} \right). \end{cases} \tag{3.4}$$

Now by substituting the known relations for the unknown parameters in the derivative form of (3.2), we obtain by simplifying $f'(w_n) \approx \frac{f[y_n,w_n]f[z_n,w_n]}{f[y_n,z_n]}$. Using this approximation and (3.1), we obtain the following four-step class

$$\begin{cases} \text{any optimal 8th-order method} = \begin{cases} f(x_n) \text{ and } f'(x_n) \text{ are available,} \\ f(y_n) \text{ is available,} \\ f(z_n) \text{ is available,} \end{cases} \\ x_{n+1} = w_n - \frac{f[y_n,z_n]f(w_n)}{f[y_n,w_n]f[z_n,w_n]}. \end{cases} \tag{3.5}$$

Applying (2.6) in the first three steps of (3.5) leads to the following four-step technique, which possesses the fourteenth-order convergence and 1.69 as the efficiency index

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - \frac{f(x_n)}{f'(x_n)} \frac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)}, \\ w_n = z_n - \frac{f(z_n)}{f'(x_n)} \left[1 + \frac{4f(z_n)}{f(x_n)}\right] \left[\frac{f(x_n)^2}{f(x_n)^2-2f(x_n)f(y_n)-f(y_n)^2} + \frac{f(z_n)}{f(y_n)}\right], \\ x_{n+1} = w_n - \frac{f[y_n,z_n]f(w_n)}{f[y_n,w_n]f[z_n,w_n]}. \end{cases} \tag{3.6}$$

Theorem 3.1. *Let α be a simple root of the sufficiently differentiable function f in an open interval I. If x_0 is sufficiently close to α, then (3.6) is of fourteenth-order and satisfies the error equation*

$$e_{n+1} = (c_2^3 - c_2c_3)^3(4c_2^4 - 7c_2^2c_3 + c_3^2 + c_2c_4)e_n^{14} + O(e_n^{15}), \tag{3.7}$$

where $e_n = x_n - \alpha$, and $c_j = \frac{1}{j!}\frac{f^{(j)}(\alpha)}{f'(\alpha)}$, $j \geq 2$.

Proof. We provide the Taylor expansion of any terms involved in (3.6). By Taylor expanding around the simple root in the *nth* iterate, we have $f(x_n) = f'(\alpha)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + \cdots + O(e_n^{15})]$ and also $f'(x_n) = f'(\alpha)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + \cdots + O(e_n^{14})]$. Using these expansions and the first step of (3.6), we have

$$y_n - \alpha = c_2e_n^2 + (-2c_2^2 + 2c_3)e_n^3 + \cdots + O(e_n^{15}). \tag{3.8}$$

We know that the first three steps are of optimal eighth-order. Taylor expansion in the second step of (3.6) by applying (3.8) yields in

$$z_n - \alpha = (c_2^3 - c_2c_3)e_n^4 - 2(2c_2^4 - 4c_2^2c_3 + c_3^2 + c_2c_4)e_n^5 + \cdots + O(e_n^{15}). \tag{3.9}$$

Note that to cut a long story short, for such a high order expansion, we here only write the 1st or 2nd terms of the error equations and the other terms are denoted by three dots. (3.9) shows that the order is 4 up to the second step. By taking into consideration (3.9) and the third step of (3.6), we attain

$$w_n - \alpha = c_2(c_2^2 - c_3)(4c_2^4 - 7c_2^2c_3 + c_3^2 + c_2c_4)e_n^8 + \cdots + O(e_n^{15}). \tag{3.10}$$

At this time the Taylor expansion of $f(w_n)$ around the simple root in the n-th iterate of (3.6) is required. Therefore, we write

$$f(w_n) = c_2(c_2^2 - c_3)(4c_2^4 - 7c_2^2c_3 + c_3^2 + c_2c_4)f'(\alpha)e_n^8 + \cdots + O(e_n^{15}). \tag{3.11}$$

Subsequently, by applying (3.11) in the last step of (3.6), we have

$$\frac{f[y_n, z_n]f(w_n)}{f[y_n, w_n]f[z_n, w_n]} = c_2(c_2^2 - c_3)(4c_2^4 - 7c_2^2c_3 + c_3^2 + c_2c_4)e_n^8 + \cdots + O(e_n^{15}). \tag{3.12}$$

Now by considering (3.12) in the last step of (3.6), we get that the error equation (3.7), which manifests that (3.6) reaches the fourteenth-order convergence using five evaluations per full cycle. This completes the proof. □

Remark 3.2. *Each method of the class reaches the efficiency index* $\sqrt[5]{14} \approx 1.69$, *which is greater than* $\sqrt[3]{4} \approx 1.58$ *of optimal fourth-order techniques',* $\sqrt[4]{6} \approx 1.56$ *of (2.2)',* $\sqrt[4]{7} \approx 1.62$ *(2.3)' and* $\sqrt[4]{8} \approx 1.68$ *of optimal eighth-order techniques'.*

Remark 3.3. *The introduced approximation of the function in the fourth-step of our cycle can be applied on any optimal eighth-order derivative-involved technique to increase the convergence rate from eight to fourteen and also the efficiency index from 1.68 to 1.69. Hence, the class of methods is efficient, though the optimality conjecture of Kung-Traub is lost. For example, applying (2.5) in (3.5) leads us to*

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \frac{f(y_n)}{f'(x_n)}\frac{f(x_n)}{f(x_n)-2f(y_n)}, \\ w_n = z_n - \frac{f(z_n)}{f'(x_n)}\left[\left(\frac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)}\right)^2 + \frac{f(z_n)}{f(y_n)} + \frac{4f(z_n)}{f(x_n)}\right], \\ x_{n+1} = w_n - \frac{f[y_n,z_n]f(w_n)}{f[y_n,w_n]f[z_n,w_n]}, \end{cases} \tag{3.13}$$

where satisfies the following error equation

$$e_{n+1} = (c_2^3 - c_2c_3)^3(13c_2^4 - 15c_2^2c_3 + c_3^2 + c_2c_4)e_n^{14} + O(e_n^{15}). \tag{3.14}$$

Remark 3.4. *The introduced estimation, i.e.* $f'(w_n) \approx (f[y_n, w_n]f[z_n, w_n]) / f[y_n, z_n]$, *is simple to implement and does not add much computational burden to the class. To discuss more, from the very beginning we wished to construct a class of higher order methods with better efficiency index than the optimal three-point eighth-order methods, which also does not contain high computational amount. The new iterations carry out less computational complexity than the optimal sixteenth-order methods' in [14] and [15].*

Note that in general and by using a similar Taylor expansion, it would be easy to deduce the following theorem regarding the fourteenth order of convergence for the general class (3.5).

Theorem 3.5. *With the same assumptions as in Theorem 3.1 and if the asymptotic error constant of the optimal derivative-involved method (in the first three sub-steps of (3.5)) is denoted by AEC(8), then any method from the new class (3.5) satisfies the error equation*

$$e_{n+1} = (c_2^3 - c_2c_3)^2 AEC(8)e_n^{14} + O(e_n^{15}). \tag{3.15}$$

In case of having multiple zeros for nonlinear functions, the procedure is somehow similar. By applying a transformation on the given function, one may first transform the multiple zero into a simple one. This procedure would add one more derivative evaluation at least automatically.

To illustrate further, we consider the transformation $h(x) := f(x)/f'(x)$, which was attributed (somehow) to [3]. Now by implementing the fourteenth-order iterative method (3.13) on the transformation $h(x)$, we can easily extend it for dealing with multiple roots, when high precision computing alongside high order is needed. By considering $\vartheta_n = \frac{(f(y_n)f'(x_n)-f(x_n)f'(y_n))^2}{(-2f(y_n)f'(x_n)+f(x_n)f'(y_n))^2} + \frac{f(z_n)(4f(y_n)f'(x_n)+f(x_n)f'(y_n))}{f(x_n)f(y_n)f'(z_n)}$, we obtain

$$\begin{cases} y_n = x_n + \frac{f(x_n)f'(x_n)}{-f'(x_n)^2+f(x_n)f''(x_n)}, \\ z_n = y_n + \frac{f(x_n)f(y_n)f'(x_n)^2}{(2f(y_n)f'(x_n)-f(x_n)f'(y_n))(f'(x_n)^2-f(x_n)f''(x_n))}, \\ w_n = z_n - \frac{f(z_n)\vartheta_n}{f'(z_n)\left(1-\frac{f(x_n)f''(x_n)}{f'(x_n)^2}\right)}, \\ x_{n+1} = w_n - \frac{h[y_n,z_n]f(w_n)}{h[y_n,w_n]h[z_n,w_n]f'(w_n)}, \end{cases} \tag{3.16}$$

wherein

$$\begin{cases} h[y_n, z_n] = \frac{\frac{f(y_n)}{f'(y_n)} - \frac{f(z_n)}{f'(z_n)}}{y_n - z_n}, \\ h[y_n, w_n] = \frac{-\frac{f(w_n)}{f'(w_n)} + \frac{f(y_n)}{f'(y_n)}}{-w_n + y_n}, \\ h[z_n, w_n] = \frac{-\frac{f(w_n)}{f'(w_n)} + \frac{f(z_n)}{f'(z_n)}}{-w_n + z_n}. \end{cases} \tag{3.17}$$

Therefore, now we have an efficient method (3.16) of order fourteen for finding the multiple roots. The classical efficiency index of (3.16) is $14^{\frac{1}{9}} \approx 1.340$, which is better than the multiple version of Newton's scheme, i.e. $2^{\frac{1}{3}} \approx 1.259$. For further readings, we refer the readers to [16–18].

4 Numerical comparisons

The reliability of the proposed class is tested in this section by comparing with an existing high order technique. We here remark that, Neta in [19] suggested a four-step method in the following form with fourteenth-order convergence rate

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \frac{f(x_n)+Af(y_n)}{f(x_n)+(A-2)f(y_n)} \frac{f(y_n)}{f'(x_n)}, \ A \in \mathbb{R} \\ w_n = z_n - \frac{f(x_n)-f(y_n)}{f(x_n)-3f(y_n)} \frac{f(z_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} + \theta_1 f^2(x_n) + \theta_2 f^3(x_n) + \theta_3 f^4(x_n), \end{cases} \tag{4.1}$$

wherein $\theta_3 = \frac{\Delta_1-\Delta_2}{F_w-F_y}$, $\theta_2 = -\Delta_1 + \theta_3(F_w + F_z)$, $\theta_1 = \varphi_w + \theta_2 F_w - \theta_3 F_w^2$ with $\Delta_1 = \frac{\varphi_w-\varphi_z}{F_w-F_z}$, $\Delta_2 = \frac{\varphi_y-\varphi_z}{F_y-F_z}$, and

$$\begin{cases} \varphi_w = \frac{1}{F_w}\left(\frac{w_n-x_n}{F_w} - \frac{1}{f'(x_n)}\right), \\ \varphi_y = \frac{1}{F_y}\left(\frac{y_n-x_n}{F_y} - \frac{1}{f'(x_n)}\right), \\ \varphi_z = \frac{1}{F_z}\left(\frac{z_n-x_n}{F_z} - \frac{1}{f'(x_n)}\right), \end{cases} \tag{4.2}$$

with $F_w = f(w_n) - f(x_n)$, $F_y = f(y_n) - f(x_n)$, and $F_z = f(z_n) - f(x_n)$.

The test nonlinear functions are displayed in Table 1. The results of comparisons are given in Table 2. We used 4000 digits floating point arithmetic in the calculations. Notice that using high arithmetic that allows us to dynamically define the number of necessary digits for the computations, is relevant. Each test function is computed for two initial guesses. As Table 2 manifests, our methods of the class are efficient and accurate. In order to investigate the behavior of the new proposed methods, we present a comparison with fourteenth-order method of Neta (4.1) with $A = 0$. (3.6) is almost the best scheme in the numerical reports.

Table 1. Test Functions and their zeros.

Test Functions	Zeros
$f_1 = e^{x^2+7x-30} - 1$	3
$f_2 = xe^{x^2} - (\sin x)^2 + 3\cos x + 5$	$-1.207647827130918927\ldots$
$f_3 = x^3 - 10$	$2.1544346900318837218\ldots$
$f_4 = (\sin x)^2 - x^2 + 1$	$1.404491648215341226\ldots$
$f_5 = 10xe^{-x^2} - 1$	$1.679630610428449940\ldots$
$f_6 = (x-1)^3 - 2$	$2.2599210498948731648\ldots$

5 Interval Newton method in Mathematica

Let $\mathbf{f'(x)}$ be an inclusion monotonic interval extension of $f'(x)$ and consider the algorithm

$$\mathbf{x^{(k+1)}} = \mathbf{x^{(k)}} \cap \mathbf{N}(\mathbf{x^{(k)}}), \qquad (\mathbf{k} = \mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots), \tag{5.1}$$

where

$$\mathbf{N}(\mathbf{x}) = \mathbf{mid}(\mathbf{x}) - \frac{\mathbf{f}(\mathbf{mid}(\mathbf{x}))}{\mathbf{f'(x)}}. \tag{5.2}$$

This is well-known as interval Newton method [20].

Theorem 5.1 (See [20]). *If an interval* $\mathbf{x^{(0)}}$ *contains a zero* x^* *of* $f(x)$*, then so does* $\mathbf{x^{(k)}}$ *for all* $k = 0, 1, 2, \ldots,$ *defined by (5.1). Furthermore, the intervals* $x^{(k)}$ *form a nested sequence converging to* x^* *if* $0 \notin \mathbf{f}'(\mathbf{x^{(0)}})$.

The interval Newton method is asymptotically error squaring. Although there are only some countable works in the literature for finding all the zeros of nonlinear functions in an interval, see e.g. [21–23], the interval schemes along with the use of the programming package Mathematica [24], provide a framework for such a purpose.

Keiper in [25] successfully coded the interval Newton method (5.2) in Mathematica. In what follows, we present that code with some changes in order to provide enough accurate initial guesses for all the real zeros in an interval and then use the new 14th-order scheme for increasing their accuracies up, when high precision computing is required.

The first piece of the code could be written as follows:

Table 2. Results of comparisons for different 14th-order methods.

Iterative methods	(4.1)	(3.6)	(3.13)
$f_1,\ x_0 = 0.5$			
$\lvert f_1(x_2)\rvert$	0.4	0.1e-1	0.1
$\lvert f_1(x_3)\rvert$	0.1e-10	0.2e-30	0.3e-18
$f_1,\ x_0 = 2.95$			
$\lvert f_1(x_2)\rvert$	0.3e-97	0.1e-118	0.3e-92
$\lvert f_1(x_3)\rvert$	0.4e-1368	0.3e-1669	0.7e-1299
$f_2,\ x_0 = -2$			
$\lvert f_2(x_2)\rvert$	0.4e-14	0.8e-28	0.2e-19
$\lvert f_2(x_3)\rvert$	0.8e-218	0.4e-410	0.1e-290
$f_2,\ x_0 = -1$			
$\lvert f_2(x_2)\rvert$	0.1e-132	0.1e-160	0.4e-141
$\lvert f_2(x_3)\rvert$	0.8e-1876	0.7e-2270	0.8e-1997
$f_3,\ x_0 = 4.5$			
$\lvert f_3(x_2)\rvert$	0.5e-50	0.5e-42	0.1e-37
$\lvert f_3(x_3)\rvert$	0.2e-722	0.3e-611	0.5e-550
$f_3,\ x_0 = 1.5$			
$\lvert f_3(x_2)\rvert$	0.2e-75	0.4e-82	0.5e-65
$\lvert f_3(x_3)\rvert$	0.5e-1078	0.3e-1172	0.1e-732
$f_4,\ x_0 = 2.8$			
$\lvert f_4(x_2)\rvert$	0.3e-47	0.2e-49	0.1e-46
$\lvert f_4(x_3)\rvert$	0.3e-671	0.8e-700	0.8e-661
$f_4,\ x_0 = 1.1$			
$\lvert f_4(x_2)\rvert$	0.1e-94	0.5e-99	0.1e-84
$\lvert f_4(x_3)\rvert$	0.8e-1335	0.1e-1395	0.9e-1195

Iterative methods	(4.1)	(3.6)	(3.13)
$f_5,\ x_0 = 2$			
$\|f_5(x_2)\|$	0.6e-71	0.7e-74	0.2e-58
$\|f_5(x_3)\|$	0.1e-1002	0.2e-1043	0.4e-895
$f_5,\ x_0 = 1.1$			
$\|f_5(x_2)\|$	0.4e-86	0.1e-86	0.3e-84
$\|f_5(x_3)\|$	0.8e-1215	0.7e-1220	0.1e-1187
$f_6,\ x_0 = 3.4$			
$\|f_6(x_2)\|$	0.2e-61	0.2e-51	0.2e-46
$\|f_6(x_3)\|$	0.6e-872	0.6e-732	0.1e-661
$f_6,\ x_0 = 2.2$			
$\|f_6(x_2)\|$	0.5e-254	0.8e-260	0.2e-249
$\|f_6(x_3)\|$	0	0	0

```
intervalnewton::rec = "MaxRecursion exceeded."; intnewt[f_, df_, x_,
{a_, b_}, eps_, n_] :=
  Block[{xmid, int = Interval[{a, b}]},
  If[b - a < eps, Return[int]];
   If[n == 0, Message[intervalnewton::rec];
    Return[int]];
   xmid = Interval[SetAccuracy[(a + b)/2, 16]];
   int = IntervalIntersection[int,
     SetAccuracy[xmid - N[f /. x -> xmid]/N[df /. x -> int], 16]];
   (intnewt[f, df, x, #, eps, n - 1]) & /@ (List @@ int)];
Options[intervalnewton] = {MaxRecursion -> 2000};
```

In the above piece of code, the interval $[a, b]$ includes the real zeros, and is refined until all zeros are detected (up to the machine precision). We are now able to apply the interval Newton's method to capture all the intervals having one unique solution in themselves. In these lines `eps` is the tolerance.

We also may note that we are working with double precision in the first step of the hybrid algorithm to rapidly obtain a list of robust initial intervals. In most practical problem $\texttt{eps} = 10^{-4}$ is enough, but in some cases it would be necessary to increase this tolerance to capture all the solutions in a root cluster. Thus, the code would be

```
intervalnewton[f_, x_, int_Interval, eps_, opts___] :=
  Block[{df, n}, n = MaxRecursion /. {opts}
  /. Options[intervalnewton];
   df = D[f, x];
   IntervalUnion @@ Select[Flatten[(intnewt[f, df, x, #, eps, n])
        & /@ (List @@ int)], IntervalMemberQ[f /. x -> #, 0] &]];
```

Now by choosing the tolerance, the lower and upper bounds and the nonlinear function, one may easily attain robust initial approximations in a form of list of numbers each having the accuracy up to at least `eps` decimal places. Let us apply this procedure on the following two very oscillatory nonlinear functions:

$$f(x) = \sin\left(10x^2\right)\cosh(x),\ \ D = [0.2, 3.], \tag{5.3}$$

and also

$$g(x) = \sin(30\sin(x)) + \frac{1}{2},\ \ D = [0., 10.]. \tag{5.4}$$

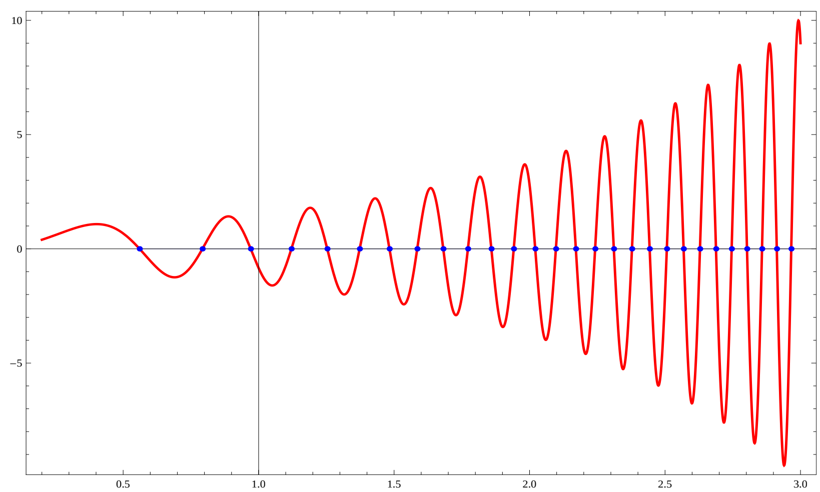

Figure 1: The plot of the function $f(x)$ with finitely many zeros.

Applying the interval Newton scheme, e.g. as follows:

```
IntervalSol = intervalnewton[g[x], x,
            Interval[{0., 10.}], .0001];
setInitial = N[Mean /@ List @@ IntervalSol]

NumberOfGuesses = Length[setInitial]
```

on the above test $g(x)$ could simply and in a short piece of time provide robust list of initial approximations for the zeros of the test nonlinear functions.

For f, we obtain 0.560499, 0.792666, 0.970816, 1.121, 1.25331, 1.37294, 1.48294, 1.58533, 1.6815, 1.77245, 1.85897, 1.94162, 2.02091, 2.0972, 2.1708, 2.242, 2.311, 2.378, 2.44316, 2.50663, 2.56851, 2.62897, 2.68806, 2.74587, 2.80249, 2.858, 2.91243, 2.96588.

For g, we attain: 0.122479, 0.193186, 0.338012, 0.413079, 0.571688, 0.657153, 0.848806, 0.961944, 1.28675, 1.85484, 2.17965, 2.29279, 2.48444, 2.56992, 2.72852, 2.80358, 2.9484, 3.01911, 3.15904, 3.22897, 3.37048, 3.44283, 3.59311, 3.6723, 3.84362, 3.93905, 4.16755, 4.32269, 5.10209, 5.25723, 5.48572, 5.58115, 5.75247, 5.83167, 5.98194, 6.0543, 6.1958, 6.26574, 6.40566, 6.47638, 6.62119, 6.69626,

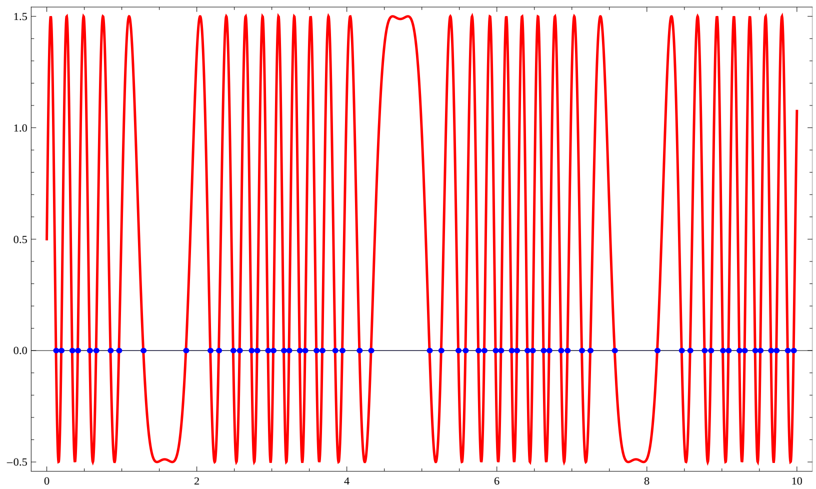

Figure 2: The plot of the function $g(x)$ with finitely many zeros.

6.85487, 6.94034, 7.13199, 7.24513, 7.56994, 8.13802, 8.46283, 8.57597, 8.76762, 8.8531, 9.0117, 9.08677, 9.23159, 9.30232, 9.44223, 9.51216, 9.65366, 9.72602, 9.8763, 9.95549. Note that the zeros have been clarified along the functions on the Figures 1-2.

The attained lists could easily be updated and their accuracies could come up to any desired tolerance using the new efficient high-order methods of this paper, when high precision computing is on the focus. For instance, the following correction part due to (3.6) by using 1500 digits and the stopping criterion $|f(x_n)| < 10^{-500}$ could be written:

```
digits = 1500; For[i = 1, i <= NumberOfGuesses, i++,
   {k = 0; X = SetAccuracy[setInitial[[i]], digits];
    While[Abs[f[X]] > 10^-500 && k <= 10,
     {k = k + 1; fX = SetAccuracy[f[X], digits];
       f1X = SetAccuracy[f'[X], digits];
       di = SetAccuracy[fX/f1X, digits];
       Y = SetAccuracy[X - di, digits];
       fY = SetAccuracy[f[Y], digits];
       Z = SetAccuracy[X - di ((fX - fY)/(fX - 2 fY)), digits];
       fZ = SetAccuracy[f[Z], digits];
       t = SetAccuracy[(fX^2)/(fX^2 - 2 fX*fY - fY^2), digits];
       s = SetAccuracy[fZ/fX, digits];
       u = SetAccuracy[fZ/fY, digits];
       W = SetAccuracy[Z - (fZ/f1X) (1 + 4 s) (t + u), digits];
       fW = SetAccuracy[f[W], digits];
       We2 = SetAccuracy[(((fY - fW)/(Y - W))*((fZ
```

```
            - fW)/(Z - W)))/((fY - fZ)/(Y - Z)), digits];
      X = SetAccuracy[W - fW/We2, digits];};];
  Print[Column[{
      "The number of full iterations is:" k,
      "The zero is:" N[X, 64],
      "The residual norm of the approximate zero is:"
      N[Abs[f[X]], 5]},
    Frame -> All]];
  }]; // AbsoluteTiming
```

Table 3. Comparison of different methods for finding all the simple zeros of f and g

Iterative methods	Newton	(2.6)	(3.6)
Computational time for f (in seconds)	3.45	2.37	2.06
Computational time for g (in seconds)	7.76	5.15	4.43

The results of comparison for some different methods based on the computational time to achieve the desired tolerance have been reported in Table 3. The results support the new scheme (3.6) from the class (3.5). Note that, the computer specifications in this paper are Intel(R) Core(TM) 2 Quad CPU, Q9550 @ 2.83GHz with 2.00GB of RAM.

6 Concluding remarks

In this paper, we have derived a novel class of four-point four-step methods for solving one variable nonlinear equations. The class was attained by approximating the first derivative of the function in the fourth step using quasi-Pade approximant. Each method of the class consists of four evaluations of the function and one evaluation of the first derivative per full cycle to reach the convergence rate 14 and the index of efficiency 1.69. We have applied only three known data in estimating $f'(w_n)$ intentionally to obtain a class, which in addition to have more convergence rate and efficiency index in contrast to the optimal eighth-order techniques, have an acceptable computational load in comparison to the optimal sixteenth-order methods.

The analytical proof for one method of the class was given. The Mathematica 8 program of a hybrid algorithm has been given to illustrate on how to capture all the real zeros of nonlinear functions in an interval, when high precision computing is needed. Finally, we should infer that the new iterative formulas can be used as an alternative to the existing methods or in some cases, where existing methods are not successful.

References

[1] M. Grau-Sanchez, Improvements of the efficiency of some three-step iterative like-Newton methods, Numer. Math. 107 (2007) 131–146.

[2] F. Soleymani, Optimized Steffensen-type methods with eighth-order convergence and high efficiency index, Int. J. Math. Math. Sci., Volume 2012 (2012), Article ID 932420, 18 pages.

[3] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, New Jersey, 1964.

[4] S.K. Khattri, S. Abbasbandy, Optimal fourth order family of iterative methods, Math. Vesnik 63 (2011) 67–72.

[5] F. Soleymani, Revisit of Jarratt method for solving nonlinear equations, Numer. Algor. 57 (2011) 377–388.

[6] A. Cordero, J.L. Hueso, E. Martinez, J.R. Torregrosa, A family of iterative methods with sixth and seventh order convergence for nonlinear equations, Math. Comput. Model. 52 (2010) 1490–1496.

[7] L. Liu, X. Wang, Eighth-order methods with high efficiency index for solving nonlinear equations, Appl. Math. Comput. 215 (2010) 3449–3454.

[8] X. Wang, L. Liu, New eighth-order iterative methods for solving nonlinear equations, J. Comput. Appl. Math. 234 (2010) 1611–1620.

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

[10] A. Iliev, N. Kyurkchiev, Nontrivial Methods in Numerical Analysis: Selected Topics in Numerical Analysis, LAP LAMBERT Academic Publishing, 2010.

[11] N. Kyurkchiev, A. Iliev, A note on the constructing of nonstationary methods for solving nonlinear equations with raised speed of convergence, Serdica J. Comput. 3 (2009) 47–74.

[12] H. Montazeri, F. Soleymani, S. Shateyi, S.S. Motsa, On a new method for computing the numerical solution of systems of nonlinear equations, J. Appl. Math., Volume 2012 (2012), Article ID 751975, 15 pages.

[13] D.K.R. Babajee, A. Cordero, F. Soleymani, J.R. Torregrosa, On a novel fourth-order algorithm for solving systems of nonlinear equations, J. Appl. Math., Volume 2012 (2012), Article ID 165452, 12 pages.

[14] Y.H. Geum, Y.I. Kim, A biparametric family of four-step sixteenth-order root-finding methods with the optimal efficiency index, Appl. Math. Lett. 24 (2011) 1336–1342.

[15] Y.H. Geum, Y.I. Kim, A biparametric family of optimally convergent sixteenth-order multipoint methods with their fourth-step weighting function as a sum of a rational and a generic two-variable function, J. Comput. Appl. Math. 235 (2011) 3178–3188.

[16] F. Soleymani, Letter to the editor regarding the article by Khattri: Derivative free algorithm for solving nonlinear equations, Computing 95 (2013) 159–162.

[17] F. Soleymani, New class of eighth-order iterative zero-finders and their basins of attraction, Afrika Matematika (2012), DOI: 10.1007/s13370-012-0100-z.

[18] F. Soleymani, S. Karimi Vanani, H.I. Siyyam, I.A. Al-Subaihi, Numerical solution of nonlinear equations by an optimal eighth-order class, Ann. Univ. Ferrara (2012), DOI: 10.1007/s11565-012-0165-5.

[19] B. Neta, On a family of multipoint methods for non-linear equations, Int. J. Comput. Math. 9 (1981) 353–361.

[20] R.E. Moore, R.B. Kearfott, M.J. Cloud, Introduction to Interval Analysis, SIAM, 2009.

[21] F. Soleymani, D.K.R. Babajee, S. Shateyi, S.S. Motsa, Construction of optimal derivative-free techniques without memory, J. Appl. Math., Volume 2012 (2012), Article ID 497023, 24 pages.

[22] F. Soleymani, S. Shateyi, Two optimal eighth-order derivative-free classes of iterative methods, Abst. Appl. Anal., Volume 2012 (2012), Article ID 318165, 14 pages.

[23] T. Johnson, W. Tucker, Enclosing all zeros of an analytic function - A rigorous approach, J. Comput. Appl. Math. 228 (2009) 418–423.

[24] R. Hazrat, Mathematica: A Problem-Centered Approach, Springer-Verlag, 2010.

[25] J.B. Keiper, Interval Arithmetic in Mathematica, The Mathematica J. 5 (1995) 66–71.

(Received 10 March 2012)
(Accepted 9 November 2012)