Journal of the Egyptian Mathematical Society (2013) **21**, 133–141

Egyptian Mathematical Society
Journal of the Egyptian Mathematical Society
www.etms-eg.org
www.elsevier.com/locate/joems

ORIGINAL ARTICLE

Some optimal iterative methods and their with memory variants

F. Soleymani *

Department of Mathematics, Islamic Azad University, Zahedan Branch, Zahedan, Iran

Received 18 October 2012; revised 11 December 2012; accepted 5 January 2013
Available online 4 February 2013

KEYWORDS
With memory;
Weight function;
R-order;
High precision computing

Abstract Based on the fourth-order method of Liu et al. [10], eighth-order three-step iterative methods without memory, which are totally free from derivative calculation and reach the optimal efficiency index are presented. The extension of one of the methods for multiple zeros without the knowledge of multiplicity is presented. Further accelerations will be provided through the concept of with memory iteration methods. Moreover, it is shown by way of illustration that the novel methods are useful on a series of relevant numerical problems when high precision computing is required.

2010 MATHEMATICS SUBJECT CLASSIFICATION: 65H05, 41A25

1. Introduction

We are concerned with numerical methods for the solution of non-linear equations in this paper [1]. It is known that the common problems associated with implementation of Newton's iteration are as follows: 1. Difficulty in evaluating derivative of a function. 2. Failure of the method to converge always. 3. Slow convergence order. To remedy these problems, many iterative techniques with/without memory have been presenting.

To overcome on the first difficulty, Steffensen in [2] replaced the first derivative of the function in the Newton's iterate by forward finite difference approximation, and he obtained

$$x_{n+1} = x_n - \frac{f(x_n)^2}{f(x_n + f(x_n)) - f(x_n)}, \quad n = 0, 1, 2, \ldots. \tag{1}$$

This method also possesses the quadratical convergence and the same efficiency $2^{\frac{1}{2}} \approx 1.414$ just like the Newton's. To circumvent on the second drawback of Newton's iterate, Yun and Petkovic [3] presented non-iterative methods, or Soleymani and his co-workers developed some hybrid techniques in [4,5]. And finally, for vanquishing the last problem many developments of different orders have been given to date; see e.g. [6,7] and the references therein.

In this study, we focus on finding new multi-point techniques, in which there is no need of derivative-calculation and also they have optimal order of convergence with high efficiency index according to the hypothesis of Kung and Traub [8] concerning the optimality of multi-point iterations without memory. The methods, which satisfy this conjecture are called optimal methods. That is in this work, we look for techniques that set to rights in the above-mentioned first and third

* Tel.: +98 9151401695.
E-mail address: fazlollah.soleymani@gmail.com

Peer review under responsibility of Egyptian Mathematical Society.

ELSEVIER Production and hosting by Elsevier

1110-256X © 2013 Egyptian Mathematical Society. Production and hosting by Elsevier B.V. Open access under CC BY-NC-ND license.
http://dx.doi.org/10.1016/j.joems.2013.01.002

problems, while applying them in a hybrid environment resolves the second difficulty. To do this, we build optimal eighth-order classes of methods by using weight functions, which includes three steps and agrees in the hypothesis of Kung and Traub. We next extend one of the new methods for multiple zeros and also obtain further accelerations in convergence and computational efficiency index without much more functional evaluation by applying the concept of with memory iteration methods.

In what follows, we shortly discuss some of the derivative-free methods in the literature. Then, Section 2 gives the contributions, where further discussions of the computational efficiency will be presented in Section 3. Section 4 supports the theoretical results by numerical testing. Finally, a short conclusion will be drawn in Section 5.

Zheng et al. in [9] provided a family of third-order derivative-free root solvers as follows

$$\begin{cases} y_n = x_n - \frac{\lambda_n f(x_n)^2}{f(x_n+\lambda_n f(x_n))-f(x_n)}, \\ x_{n+1} = x_n - \frac{\lambda_n f^3(x_n)}{[f(x_n+\lambda_n f(x_n))-f(x_n)][f(x_n)-f(y_n)]}, \end{cases} \tag{2}$$

with three evaluations per iteration wherein $\lambda_n \in \mathbb{R}\setminus\{0\}$. This technique has $3^{\frac{1}{3}} \approx 1.442$ as its efficiency index.

Liu et al. in [10] gave an optimal quartically convergent derivative-free technique in the following structure with three evaluations of the function per iteration

$$\begin{cases} y_n = x_n - \frac{f(x_n)^2}{f(x_n+f(x_n))-f(x_n)}, \\ x_{n+1} = y_n - \frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n), \end{cases} \tag{3}$$

wherein $w_n = x_n + f(x_n)$. We here remark that $f[x_n, y_n]$, $f[y_n, w_n]$ and $f[x_n, w_n]$ are divided differences. This scheme has $4^{\frac{1}{3}} \approx 1.587$ as its efficiency index.

In [11], the authors furnished two non-optimal derivative-free methods of orders four and six. The quartically convergent Cordero et al. method is in the form below

$$\begin{cases} y_n = x_n - \frac{2f(x_n)^2}{f(x_n+f(x_n))-f(x_n-f(x_n))}, \\ x_{n+1} = x_n - \frac{2f(x_n)^2}{f(x_n+f(x_n))-f(x_n-f(x_n))}\,\frac{f(y_n)-f(x_n)}{2f(y_n)-f(x_n)}, \end{cases} \tag{4}$$

where consists of four evaluations of the function and possesses $4^{\frac{1}{4}} \approx 1.414$ as its efficiency index, just the same as Steffensen's or Newton's. Their sixth-order technique which includes five evaluations of the function per iteration to reach the efficiency $6^{\frac{1}{5}} \approx 1.430$ can be defined by

$$\begin{cases} y_n = x_n - \frac{2f(x_n)^2}{f(x_n+f(x_n))-f(x_n-f(x_n))}, \\ z_n = y_n - \frac{y_n-x_n}{2f(y_n)-f(x_n)} f(y_n), \\ x_{n+1} = z_n - \frac{y_n-x_n}{2f(y_n)-f(x_n)} f(z_n). \end{cases} \tag{5}$$

We here remind the well-known family of derivative-free methods which was given by Kung and Traub in [8] as comes next

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]},\ w_n = x_n + \beta f(x_n),\ \beta \in \mathbb{R}\setminus\{0\}, \\ z_n = y_n - \frac{f(y_n)f(w_n)}{[f(w_n)-f(y_n)]f[x_n,y_n]}, \\ x_{n+1} = z_n - \frac{f(y_n)f(w_n)\left[y_n-x_n+\frac{f(x_n)}{f[x_n,z_n]}\right]}{[f(y_n)-f(z_n)][f(w_n)-f(z_n)]} + \frac{f(y_n)}{f[z_n,y_n]}. \end{cases} \tag{6}$$

This family of one-parameter methods possesses the eighth-order convergence utilizing four pieces of information, namely, $f(x_n), f(y_n), f(z_n)$ and $f(w_n)$. Therefore, its classical efficiency index is $8^{\frac{1}{4}} \approx 1.682$.

Li et al. in [12] discussed the performance of derivative-free methods in multiple zero-finding by applying the Schroder transformation, see [13,14], for converting a multiple zero to a simple one. For more information on this field, one may consult the papers [15–17].

2. Development of the methods

Let us take heed of the following three-step without memory cycle in which (3) is in the first two steps

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]}, \\ z_n = y_n - \frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}, \end{cases} \tag{7}$$

wherein $w_n = x_n + f(x_n)$ and we have four evaluations of the function and one evaluation of the first-order derivative. Now, the main challenge is to approximate $f'(z_n)$ as efficiently as possible to gain a novel derivative-free method with better order of convergence and better efficiency index in contrast with the optimal fourth-order schemes and the same as (6).

Hence, we take into consideration an interpolating polynomial as comes next

$$f(t) \approx I(t) = a(t-x_n)^2 + b(t-x_n) + c, \tag{8}$$

whence this approximation polynomial satisfies the interpolation conditions $f(x_n) = I(x_n)$, $f(y_n) = I(y_n)$ and $f(z_n) = I(z_n)$. By substituting the known data in $I(t)$, we have a system of three linear equations with three unknowns. By solving this system and simplifying, we have

$$\begin{cases} a = \frac{(y_n-z_n)f(x_n)+(-x_n+z_n)f(y_n)+(x_n-y_n)f(z_n)}{(x_n-y_n)(x_n-z_n)(y_n-z_n)}, \\ b = \frac{(x_n-z_n)^2(f(x_n)-f(y_n))+(x_n-y_n)^2(-f(x_n)+f(z_n))}{(x_n-y_n)(x_n-z_n)(y_n-z_n)}, \\ c = f(x_n). \end{cases} \tag{9}$$

Due to this, a powerful approximation of the first derivative of the function in the third step is attained as comes next

$$\begin{aligned} f'(z_n) \approx I'(z_n) &= 2a(z_n - x_n) + b \\ &= f[x_n, z_n] + f[z_n, y_n] - f[x_n, y_n]. \end{aligned} \tag{10}$$

We here recall that $f[x_n, x_{n-1}, \ldots, x_{n-i}]$ is the divided differences of $f(x)$. And they can be defined recursively via $f[x_i] = f(x_i)$, $f[x_i, x_j] = \frac{f[x_i]-f[x_j]}{x_i-x_j}$, $x_i \neq x_j$, and for $m > i+1$, via

$$f[x_i, x_{i+1}, \ldots, x_m] = \frac{f[x_i, x_{i+1}, \ldots, x_{m-1}] - f[x_{i+1}, x_{i+2}, \ldots, x_m]}{x_i - x_m}, \quad x_i \neq x_m. \tag{11}$$

Eventually, using (10) in the last step of (7) leads to the following high-order technique

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]},\ w_n = x_n + f(x_n), \\ z_n = y_n - \frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f[x_n,z_n]+f[z_n,y_n]-f[x_n,y_n]}, \end{cases} \tag{12}$$

on which there are four function-evaluation per full cycle and it is totally free from any derivative. Unfortunately, the error equation of this method has turned out to be seven as comes next

$$\begin{aligned} e_{n+1} = &\frac{(1+c_1)^2 c_2^2 c_3(-(2+c_1)c_2^2 + c_1(1+c_1)c_3)}{c_1^5} e_n^7 \\ &+ O(e_n^8), \end{aligned} \tag{13}$$

where $c_k = f^{(k)}(\alpha)/k!$, $\forall k = 1, 2, 3, \ldots$. Obviously, this procedure is in not optimal according to the hypothesis of Kung and Traub [8]. Since, a multi-point method consuming four function evaluations should reach the maximum convergence order eight. To remedy this, we take into account of the weight function approach to give two new classes of optimal local order eight.

Consequently, we consider the following uni-parametric family of iterations, which according to Theorem 2.1. reaches the convergence order eight using four pieces of information per full cycle

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]}, w_n = x_n + f(x_n), \\ z_n = y_n - \frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f[x_n,z_n]+f[z_n,y_n]-f[x_n,y_n]} \left\{ 1 + \frac{f(z_n)}{f(w_n)} + \theta\left(\frac{f(z_n)}{f(y_n)}\right)^2 - (2 + f[x_n,w_n])\left(\frac{f(y_n)}{f(w_n)}\right)^3 \right\}, \end{cases} \tag{14}$$

where $\theta \in \mathbb{R}$. In what follows, we observe that (14) satisfies the conjecture of Kung and Traub. A discussion on how we obtained the weight function in the third step of (14) will be given after the proof.

Theorem 2.1. *Let us consider α as a simple root of the non-linear equation $f(x) = 0$ in the domain D. And assume that $f(x)$ is sufficiently smooth in the neighborhood of the root, i.e. D. Then the derivative-free iterative scheme defined by (14) is of optimal local order eight and has the following error equation*

$$e_{n+1} = -\frac{1}{c_1^7}(1+c_1)c_2((2+c_1)c_2^2 - c_1(1+c_1)c_2)((1 + c_1)c_2((4+c_1)c_2^3 + 2c_1(1+c_1)c_2c_3 - c_1^2(1 + c_1)c_4) + ((2+c_1)c_2^2 - c_1(1+c_1)c_3)^2\theta)e_n^8 + O(e_n^9). \tag{15}$$

Proof. Using Taylor series and symbolic computation, we can determine the asymptotic error constant of the three-step uni-parametric family (14). Furthermore, assume $e_n = x_n - \alpha$ be the error in the nth iterate and take into account $f(\alpha) = 0$, $c_k = f^{(k)}(\alpha)/k!$, $\forall k = 1, 2, 3, \ldots$. Now, we expand $f(x_n)$ around the simple zero α. Hence, we have

$$f(x_n) = c_1e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + \cdots + O(e_n^9). \tag{16}$$

Note that to save the space, we only write some of the obtained terms for the error equations and show the others by $\cdots$. By considering (16) and the first step of (14), we attain

$$x_n - \frac{f(x_n)}{f[x_n,w_n]} = \alpha + \left(1 + \frac{1}{c_1}\right)c_2e_n^2 + \frac{(-(2+(2+c_1)c_1)c_2^2 + c_1(1+c_1)(2+c_1)c_3)}{c_1^2}e_n^3 + \cdots + O(e_n^9). \tag{17}$$

We should expand $f(y_n)$ around the root by using (17). Accordingly, we have

$$f(y_n) = (1+c_1)c_2e_n^2 + \left(-\frac{(2+c_1(2+c_1))c_2^2}{c_1 + (1+c_1)(2+c_1)c_3}\right)e_n^3 + \cdots + O(e_n^9). \tag{18}$$

Applying (17) and (18) in the second step of (14) gives us

$$z_n - \alpha = \frac{(1+c_1)c_2\big((2+c_1)c_2^2 - c_1(1+c_1)c_3\big)e_n^4}{c_1^3} + \cdots + O(e_n^9). \tag{19}$$

On the other hand, we obtain

$$f(z_n) = \frac{(1+c_1)c_2\big((2+c_1)c_2^2 - c_1(1+c_1)c_3\big)}{c_1^2}e_n^4 + \cdots + O(e_n^9). \tag{20}$$

This leads us to find the error equation of the denominator of the last step of (14) as follows

$$f[x_n,z_n] + f[z_n,y_n] - f[x_n,y_n] = c_1 - \frac{(1+c_1)c_2c_3e_n^3}{c_1} + \cdots + O(e_n^9). \tag{21}$$

Dividing (20) by (21) ends in

$$\frac{f(z_n)}{f[x_n,z_n] + f[z_n,y_n] - f[x_n,y_n]} = \frac{(1+c_1)c_2((2+c_1)c_2^2 - c_1(1+c_1)c_3)}{c_1^2}e_n^4 + \cdots + O(e_n^9). \tag{22}$$

Additionally, for the weight function in the last step of (14), we have

$$1 + \frac{f(z_n)}{f(w_n)} + \theta\left(\frac{f(z_n)}{f(y_n)}\right)^2 - (2 + f[x_n,w_n])\left(\frac{f(y_n)}{f(w_n)}\right)^3 = 1 - \frac{(1+c_1)c_2c_3}{c_1^2}e_n^3 + \cdots + O(e_n^9). \tag{23}$$

By considering (22) and (23) in the last step of (14), we attain the error Eq. (15). This manifests that (14) has the optimal eighth-order convergence and ends the proof. □

One might ask that how the weight function

$$1 + \frac{f(z_n)}{f(w_n)} + \theta\left(\frac{f(z_n)}{f(y_n)}\right)^2 - (2 + f[x_n,w_n])\left(\frac{f(y_n)}{f(w_n)}\right)^3, \tag{24}$$

was attained. To respond this, in fact, at the end of the third step of (12), we should consider a weight function by considering the values of the function at the known nodes to increase the order one unit. Toward this end, we had taken into consideration the weight functions as comes next

$$z_n - \frac{f(z_n)}{f[x_n,z_n] + f[z_n,y_n] - f[x_n,y_n]} \times \left\{ G\left(\frac{f(z_n)}{f(w_n)}\right) + H\left(\frac{f(z_n)}{f(y_n)}\right) + L\left(\frac{f(y_n)}{f(w_n)}\right) \right\}, \tag{25}$$

wherein $G(t)$, $H(\sigma)$, and $L(\tau)$ are three weight functions, in which $t = \frac{f(z)}{f(w)}$, $\sigma = \frac{f(z)}{f(y)}$, and $\tau = \frac{f(y)}{f(w)}$, (without the index n). Clearly, these three weight functions should be constructed such that the order of convergence arrives at eight. Taylor's series expansion at the end of the third step shows that

$$\begin{cases} G(0) = G'(0) = 1, \\ H(0) = H'(0) = 0 \quad \text{and } |H''(0)| < \infty, \\ L(0) = L'(0) = L''(0) = 0, L^{(3)}(0) \\ = -6(2 + f[x_n,w_n]), \quad \text{and } |L^{(4)}(0)| < \infty, \end{cases} \tag{26}$$

should be chosen to achieve our goal. One of such cases which satisfy (26) is given in (14).

Eq. (14) has some interesting features in comparison with the existing derivative-free methods in the literature. First, its order is greater than the optimal fourth-order methods. And second, it possesses 1.682 as its classical efficiency index, which is greater than $2^{\frac{1}{2}} \approx 1.414$ of (1) and (4), $3^{\frac{1}{3}} \approx 1.441$ of (2), $4^{\frac{1}{3}} \approx 1.587$ of (3), $6^{\frac{1}{5}} \approx 1.430$ of (5) and is equal to $8^{\frac{1}{4}} \approx 1.682$ of Kung-Traub family (6). Choosing $\theta = 0$ in (14) results in the follow-up optimal eighth-order method

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]}, \quad w_n = x_n + f(x_n), \\ z_n = y_n - \frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f[x_n,z_n]+f[z_n,y_n]-f[x_n,y_n]} \left\{ 1 + \frac{f(z_n)}{f(w_n)} \right. \\ \left. \quad -(2 + f[x_n, w_n]) \left(\frac{f(y_n)}{f(w_n)}\right)^3 \right\}, \end{cases} \tag{27}$$

where its error equation is

$$e_{n+1} = -\frac{(1+c_1)^2 c_2^2(-(2+c_1)c_2^2 + c_1(1+c_1)c_3)(-(4+c_1)c_2^3 - 2c_1(1+c_1)c_2c_3 + c_1^2(1+c_1)c_4)}{c_1^7} e_n^8 + O(e_n^9). \tag{28}$$

Up to now, we in fact gave a novel three-step four-point without memory class of iterations as comes next (with forward finite difference approximation)

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]}, \quad w_n = x_n + f(x_n), \\ z_n = y_n - \frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f[x_n,z_n]+f[z_n,y_n]-f[x_n,y_n]} \\ \quad \times \left\{ G(\frac{f(z_n)}{f(w_n)}) + H\left(\frac{f(z_n)}{f(y_n)}\right) + L(\frac{f(y_n)}{f(w_n)}) \right\}, \end{cases} \tag{29}$$

where the weight functions satisfy (26). Another similar class of optimal eighth-order derivative-free methods can be constructed as comes next as the second contribution of this paper (with backward finite difference approximations)

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]}, \quad w_n = x_n - \beta f(x_n), \\ z_n = y_n - \frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f[x_n,z_n]+f[z_n,y_n]-f[x_n,y_n]} \\ \quad \times \left\{ M(\frac{f(y_n)}{f(w_n)}) + K\left(\frac{f(z_n)}{f(w_n)}\right) \right\}, \end{cases} \tag{30}$$

where $\beta \in \mathbb{R} \setminus \{0\}$ and the weight function should satisfy

$$\begin{cases} M(0) = 1, \quad M'(0) = M''(0) = 0, \\ M^{(3)}(0) = -12 + 6\beta f[x_n, w_n], \quad \text{and } |M^{(4)}(0)| < \infty, \\ K(0) = 0, \quad K'(0) = 1, \end{cases} \tag{31}$$

with the following error equation

$$\begin{aligned} e_{n+1} = &-(1/(24c_1^7))(c_2^2(-1+c_1\beta)(c_1c_3(1-c_1\beta)+c_2^2(-2+c_1\beta))(24(-1 \\ &+c_1\beta) \times (c_1^2c_4(1-c_1\beta)+c_2^3(-4+c_1\beta)+2c_1c_2c_3(-1+c_1\beta)) \\ &+c_2^3M^{(4)}(0)))e_n^8+O(e_n^9). \end{aligned} \tag{32}$$

An efficient example from our new class (30), (31) can be the following

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]}, \quad w_n = x_n - f(x_n), \\ z_n = y_n - \frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f[x_n,z_n]+f[z_n,y_n]-f[x_n,y_n]} \\ \quad \times (1 - (2 - f[x_n, w_n]) \left(\frac{f(y_n)}{f(w_n)}\right)^3 - \frac{1}{24} \left(\frac{f(y_n)}{f(w_n)}\right)^4 + \frac{f(z_n)}{f(w_n)}), \end{cases} \tag{33}$$

where its error equation is

$$\begin{aligned} e_{n+1} = &-(1/(24c_1^7))((-1+c_1)c_2^2((-2+c_1)c_2^2 - (-1 \\ &+ c_1)c_1c_3)((95 + 24(-5+c_1)c_1)c_2^3 + 48(-1 \\ &+ c_1)^2c_1c_2c_3 - 24(-1+c_1)^2c_1^2c_4))e_n^8 + O(e_n^9). \end{aligned} \tag{34}$$

Many similar methods from our proposed classes (29) and (30) can be produced in which the highest possible order of convergence is attained by using the smallest possible number of function evaluations. Each member from the classes is optimal and reaches the optimal efficiency index $8^{\frac{1}{4}} \approx 1.682$. Note that (30) is a uni-parametric class in which whatever the smaller positive value of β be chosen, then the numerical results will be much better.

Some other examples from the new class are

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]}, \quad w_n = x_n - f(x_n), \\ z_n = y_n - \frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f[x_n,z_n]+f[z_n,y_n]-f[x_n,y_n]} \\ \quad \times \left(1 - (2 - f[x_n, w_n]) \left(\frac{f(y_n)}{f(w_n)}\right)^3 + \frac{f(z_n)}{f(w_n)}\right), \end{cases} \tag{35}$$

where its error equation is

$$\begin{aligned} e_{n+1} = &(-1/c_1^7)(-1+c_1)^2c_2^2((-2+c_1)c_2^2 - (-1 \\ &+ c_1)c_1c_3)((-4+c_1)c_2^3 + 2(-1+c_1)c_1c_2c_3 - (-1+c_1)c_1^2c_4)e_n^8 \\ &+ O(e_n^9), \end{aligned} \tag{36}$$

and also

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]}, \quad w_n = x_n - \beta f(x_n), \\ z_n = y_n - \frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f[x_n,z_n]+f[z_n,y_n]-f[x_n,y_n]} \\ \quad \times (1 - (2 - \beta f[x_n, w_n]) \left(\frac{f(y_n)}{f(w_n)}\right)^3 + \frac{f(z_n)}{f(w_n)}). \end{cases} \tag{37}$$

The free non-zero parameter β in (37) plays an important role in the implementation of the new methods. It is also called as the self-accelerating parameter, which will used in the next section.

Note that for tackling multiple roots, one may apply a transformation on the given function, to make the multiple zero into a simple one. This procedure would add one more derivative evaluation at least automatically. That is to say, a first-order derivative will be involved.

To illustrate further, we consider the transformation $h(x) := f(x)/f'(x)$, which was attributed to [13,14]. Now by implementing the optimal eighth-order family (37) on the transformation $h(x)$, we can extend it for dealing with multiple roots.

By considering $\beta \in \mathbb{R} \setminus \{0\}$, $w_n = x_n - (\beta f(x_n))/f'(x_n)$ and $\mathcal{FD} = \frac{1-(f(w_n)f'(x_n))/(f(x_n)f'(w_n))}{\beta}$, we obtain

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{\mathcal{FD} f'(x_n)}, \\ z_n = y_n - \frac{(\mathcal{FD}+h[x_n,y_n]-h[y_n,w_n])f[y_n]}{h[x_n,y_n]^2 f'[y_n]}, \\ x_{n+1} = z_n - \frac{f[z_n]\left(1+\frac{(-2+\beta\mathcal{FD})f[y_n]^3 f'[w_n]^3}{f[w_n]^3 f'[y_n]^3}+\frac{f[z_n]f'[w_n]}{f[w_n]f'[z_n]}\right)}{(-h[x_n,y_n]+h[x_n,z_n]+h[z_n,y_n])f'[z_n]}, \end{cases} \tag{38}$$

wherein

$$\begin{cases} h[x_n, y_n] = \frac{\frac{f[x_n]}{f'[x_n]} - \frac{f[y_n]}{f'[y_n]}}{x_n - y_n}, \\ h[y_n, w_n] = \frac{-\frac{f[w_n]}{f'[w_n]} + \frac{f[y_n]}{f'[y_n]}}{-w_n + y_n}, \\ h[x_n, z_n] = \frac{\frac{f[x_n]}{f'[x_n]} - \frac{f[z_n]}{f'[z_n]}}{x_n - z_n}, \\ h[z_n, y_n] = \frac{-\frac{f[y_n]}{f'[y_n]} + \frac{f[z_n]}{f'[z_n]}}{-y_n + z_n}. \end{cases} \tag{39}$$

3. Further acceleration

As was mentioned in the previous section, the free non-zero parameter has a very important role in further acceleration of convergence for the new family of methods, e.g. in (37). Traub in [14] discussed on how to improve and present iteration methods with memory in details using an approximation for the non-zero parameter of Steffensen's scheme by a technique called as Scant approach.

As a matter of fact, based on the presented family without memory (37), we can present new iterative methods with memory in this section. Accelerations of convergence speed are obtained in this way, by using the self-accelerating parameter β. This self-accelerating parameter is applied to improve the order of convergence. To discuss more, we remind the following important remarks.

Remark 1. Generally speaking, highest possible orders via methods with memory could be constructed out of optimal methods without memory, i.e. an efficient procedure launches n-step Steffensen-type methods with memory, with the order up to $2^n + 2^{n-1}$, (50% of an improvement) requiring the same computational cost to the corresponding families without memory. Note that by using only one accelerator.

Remark 2. With the choice $\beta = -1/f'(\alpha)$ when forward finite difference is used throughout the cycle and $\beta = 1/f'(\alpha)$ when backward finite difference approximation has been used throughout the cycle, it can be proved that the order of the optimal Steffensen-type methods without memory would exceed the optimal bound. However, the exact value of $f'(\alpha)$ is not available in practice, and such acceleration of convergence cannot be realized. But, we could approximate this parameter by an iteration via the existing data per computing step.

Remark 3. Following Remarks 1–2, basically one has two techniques at hand to attain the highest possible convergence R-order for with memory methods with one accelerator throughout the cycle only. That is, using an interpolation passing through $n + 2$ nodes for an n-step optimal without memory Steffensen-type method of the degree $n + 1$, the maximal R-order could be achieved. For example, for (37) and by applying an interpolation of degree 4 passing through five nodes, x_{old}, w, y, z, x_{new}, (per computing step) one may obtain the highest possible convergence R-order.

Remark 4. The typical interpolating way is taking into account the Newton interpolation polynomial of the degree $n + 1$, and the second way is the rational interpolation, also known as Pade interpolation of an appropriate degree. Note that higher R-order is equal to higher computational burden per computing step, though the efficiency index comes up dramatically.

For example, using (37) (by replacing β with β_n)

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n - \beta_n f(x_n), \\ z_n = y_n - \frac{f[x_n, y_n] - f[y_n, w_n] + f[x_n, w_n]}{f[x_n, y_n]^2} f(y_n), \\ x_{n+1} = z_n - \frac{f(z_n)}{f[x_n, z_n] + f[z_n, y_n] - f[x_n, y_n]} \\ \quad \times \left(1 - (2 - \beta_n f[x_n, w_n]) \left(\frac{f(y_n)}{f(w_n)}\right)^3 + \frac{f(z_n)}{f(w_n)}\right), \end{cases} \tag{40}$$

and considering the interpolation polynomial as in (8)–(10) passing through three nodes only (even though we have five possible nodes per computing step of (40)), we can increase the convergence R-order from 8 to 9.58. That is, we could obtain

$$1/f'(\alpha) \approx 1/(f[y, x_{new}] + f[x_{new}, z] - f[y, z]). \tag{41}$$

The increase of convergence R-order is attained without any additional calculations so that the novel method with memory possesses a very high computational efficiency index.

In fact, the parameter β can be computed by using information available from the current and previous iteration such that the eighth-order asymptotic convergence constant to be zero in the error equation for the family (40).

Theorem 3.1. *Let the function $f(x)$ be sufficiently differentiable in a neighborhood of its simple zero α. If an initial approximation x_0 is sufficiently close to α and the parameter β_n in (40) is recursively calculated by (41). Then, the R-order of convergence of the three-step method (40) is at least 9.58.*

Proof. We now obtain the order of convergence of the family of methods with memory (40), where β_n is calculated from (41). The error relations with the self-accelerating parameter $\beta = \beta_n$ for (40) are in what follows (assuming this time $c_k = f^{(k)}(\alpha)/(k!f'(\alpha))$, $k = 2, 3, \ldots$)

$$\hat{e}_n = w_n - \alpha \sim (-1 + \beta_n f'(\alpha)) e_n, \tag{42}$$
$$\tilde{e}_n = y_n - \alpha \sim c_2(-1 + \beta_n f'(\alpha)) e_n^2, \tag{43}$$
$$\bar{e}_n = z_n - \alpha \sim c_{n,4}(-1 + \beta_n f'(\alpha)) e_n^4, \tag{44}$$
$$e_{n+1} = x_{n+1} - \alpha \sim c_{n,8}(-1 + \beta_n f'(\alpha))^2 e_n^8. \tag{45}$$

In order to find the error relation for (40) and (41), we need to find the expression for $-1 + \beta_n f'(\alpha)$. Using a symbolic software such as Mathematica with the use of (41), we attain that

$$-1 + \beta_n f'(\alpha) \sim c_3 \tilde{e}_{n-1} \bar{e}_{n-1}. \tag{46}$$

Substituting the value of $-1 + \beta_n f'(\alpha)$ from (46) in (45), one may obtain

$$e_{n+1} \sim c_{n,8}(c_3 \tilde{e}_{n-1} \bar{e}_{n-1})^2 e_n^8, \tag{47}$$
$$e_{n+1} \sim c_{n,8} c_3^2 \tilde{e}_{n-1}^2 \bar{e}_{n-1}^2 e_n^8. \tag{48}$$

From (44), we can write

$$\bar{e}_{n-1} \sim c_{n-1,4}(-1 + \beta_{n-1} f'(\alpha)) e_{n-1}^4. \tag{49}$$

Using (49) in (48) and further simplifying, we get that

$$e_{n+1} \sim c_{n,8} c_3^2 (c_2(-1 + \beta_{n-1} f'(\alpha)) \times e_{n-1}^2)^2 (c_{n-1,4}(-1 + \beta_{n-1} f'(\alpha)) e_{n-1}^4)^2 e_n^8. \tag{50}$$

And thus

$$e_{n+1} \sim c_{n,8}c_2^2c_3^2c_{n-1,4}^2(-1+\beta_{n-1}f'(\alpha))^4e_{n-1}^{12}e_n^8. \quad (51)$$

According to (45), we can write

$$e_n \sim c_{n-1,8}(-1+\beta_{n-1}f'(\alpha))^2e_{n-1}^8. \quad (52)$$

Combining (51) and (52), we obtain

$$e_{n+1} \sim \frac{e_n^{10}c_2^2c_3^2c_{n-1,4}^2c_{n,8}}{e_{n-1}^4c_{n-1,8}^2}. \quad (53)$$

Note that in general we know that the error equation should read $e_{n+1} \sim Ae_n^p$, where A and p are to be determined. Hence, one has $e_n \sim Ae_{n-1}^p$, and subsequently $e_{n-1} \sim A^{-1/p}e_n^{1/p}$. Thus, it is easy to obtain

$$e_n^p \sim \frac{A^{-1+\frac{4}{p}}c_2^2c_3^2c_{n-1,4}^2c_{n,8}}{c_{n-1,8}^2}e_n^{10-\frac{4}{p}}, \quad (54)$$

which results in the equation $p = 10 - \frac{4}{p}$, with two solutions $\{0.417424, 9.58258\}$. Clearly the value for $p = 9.58258$ is acceptable and would be the convergence R-order of the family (40) with memory. The proof is complete. □

The computational efficiency index of the family (40) with memory is 1.7594 which is even better than optimal sixteenth-order methods without memory. Clearly this R-order for (40) is not the optimal bound, and considering a much enriched approximation for $f'(\alpha)$ per computing step by applying all five involved nodes results in the highest possible convergence R-order.

Remark 5. An important remark that must be exposed is the fact regarding the form of the error equation in the optimal eighth-order derivative-free methods, when finding the maximum convergence R-order for with memory methods. That is to say, for families without memory with the error equation (for eighth-order methods as an example) $e_{n+1} = x_{n+1} - \alpha \sim c_{n,8}(-1+\beta_nf'(\alpha))^4e_n^8$, the maximum R-order would be 12 by applying all five nodes (50% improvement), while there is *no* such thing, when the error equation is of the form $e_{n+1} = x_{n+1} - \alpha \sim c_{n,8}(-1+\beta_nf'(\alpha))^2e_n^8$. To discuss further, in such a case there would be 25% of improvement, when the family becomes with memory by applying all five nodes.

Thus the maximum R-order totally depends on the form of the error equation in general. A comparison between the without memory and with memory methods in terms of the maximum convergence order and the maximum efficiency index alongside the number of steps per cycle are given in Figs. 1 and 2. Taking into consideration the above remarks, we have different ways to produce with memory iterations, that only some of them reach the highest possible with memory bound. Considering Remark 5, the maximum convergence R-order for the family (40), would be 10.

Theorem 3.2. *Let the function $f(x)$ be sufficiently differentiable in a neighborhood of its simple zero α. If an initial approximation x_0 is sufficiently close to α and the parameter β_n in (40) is recursively calculated by (55). Then, the R-order of convergence of the three-step method (40) is at least 10.*

Proof. The proof of this theorem is similar to the proof of Theorem 3.1, hence it is omitted. □

Using the Newton interpolation passing through five active nodes $(x_{old}, f(x_{old}))$, $(w, f(w))$, $(y, f(y))$, $(z, f(z))$, $(x_{new}, f(x_{new}))$, per computing step to form an interpolation of degree four gives us the highest possible R-order for (40). Note that many authors tried to form such interpolating polynomials based on divided differences, though it is correct, we believe that built-in algorithms in the programming package Mathematica [18,19], are mostly better in terms of computational time for operations. Such a goal is simply coded in what follows:

```
f[t_] :=InterpolatingPolynomial[{{xold, fxold},
  {w, fw}, {y, fy}, {z, fz},
       {xnew, fxnew}}, t] // Simplify
(1/f'[t]) /. Thread[t-> xnew] // FullSimplify
```

which simply provides the following approximation (much more easier than the closed form of divided differences approach)

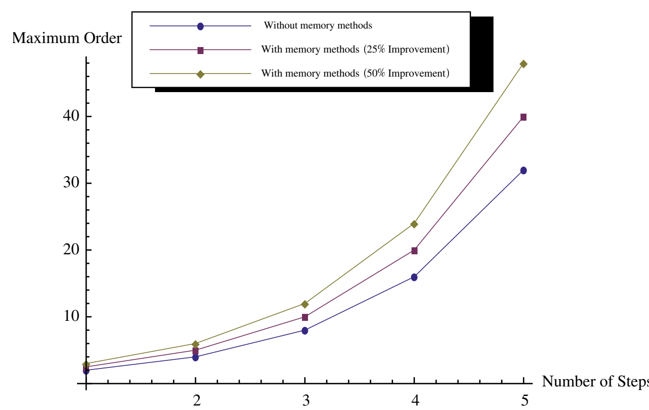

Figure 1 Comparison of methods without memory and with memory (25% and 50% of improvements) in terms of highest possible convergence order.

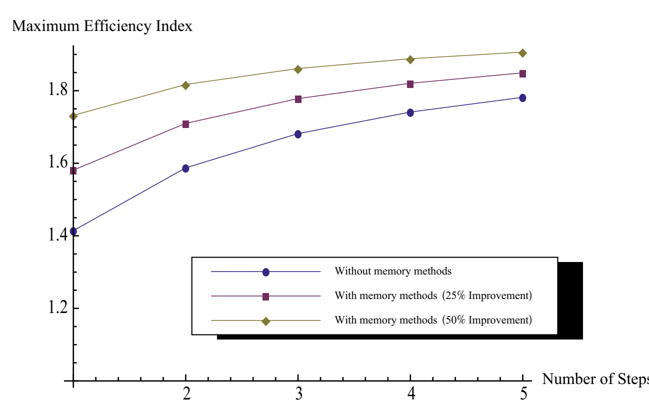

Figure 2 Comparison of methods without memory and with memory (25% and 50% of improvements) in terms of highest possible efficiency index.

Table 1 The examples considered in this study.

Test functions	Zeros
$f_1(x) = (\sin x)^2 + x$	$\alpha_1 = 0$
$f_2(x) = (1 + x^3)\cos(\frac{\pi x}{2}) + \sqrt{1 - x^2} - \frac{2(9\sqrt{2}+7\sqrt{3})}{27}$	$\alpha_2 = 1/3$
$f_3(x) = (\sin x)^2 - x^2 + 1$	$\alpha_3 \approx 1.404491648215341$
$f_4(x) = e^{-x} + \sin(x) - 1$	$\alpha_4 \approx 2.076831274533113$
$f_5(x) = xe^{-x} - 0.1$	$\alpha_5 \approx 0.111832559158963$
$f_6(x) = \sqrt{x^4 + 8}\sin(\frac{\pi}{x^2+2}) + \frac{x^3}{x^4+1} - \sqrt{6} + \frac{8}{17}$	$\alpha_6 = -2$
$f_7(x) = \sqrt{x^2 + 2x + 5} - 2\sin(x) - x^2 + 3$	$\alpha_7 \approx 2.331967655883964$
$f_8(x) = \arcsin(x^2 - 1) - \frac{x}{2} + 1$	$\alpha_8 \approx 0.594810968398369$
$f_9(x) = (\sin(x) - \frac{2}{2})(x + 1)$	$\alpha_9 \approx 0.785398163397448$
$f_{10}(x) = x^5 + x^4 + 4x^2 - 15$	$\alpha_{10} \approx 1.347428098968304$

Table 2 Results of convergence under fair circumstances for different derivative-free methods.

f, Guess	(1)	(3)	(4)	(5)	(6)	(27)	(35)
f_1, 0.6							
IT	8	4	4	3	3	3	3
TNE	16	12	16	15	12	12	12
$\vert f \vert$	0.2e−113	0.1e−90	0.4e−141	0.2e−93	0.1e−182	0.1e−200	0.3e−317
f_2, 0.8							
IT	8	4	4	3	3	3	3
TNE	16	12	16	15	12	12	12
$\vert f \vert$	0.5e−166	0.5e−104	0.1e−35	0.5e−28	0.2e−113	0.3e−146	0.3e−98
f_3, 2							
IT	8	4	4	3	3	3	3
TNE	16	12	16	15	0	12	12
$\vert f \vert$	0.4e−62	0.1e−82	0.9e−180	0.2e−109	Div.	0.4e−111	0.1e−154
f_4, 2.8							
IT	8	4	4	3	3	3	3
TNE	16	12	16	15	12	12	12
$\vert f \vert$	0.6e−276	0.1e−184	0.9e−128	0.1e−94	0.3e−398	0.3e−405	0.4e−219
f_5, −0.7							
IT	8	4	4	3	3	3	3
TNE	16	12	16	15	12	12	12
$\vert f \vert$	0.3e−8	0.9e−31	Div.	Div.	0.4e−53	0.3e−112	0.2e−117
f_6, −1.7							
IT	8	4	4	3	3	3	3
TNE	16	12	16	15	12	12	12
$\vert f \vert$	0.4e−221	0.3e−134	0.6e−119	0.2e−91	0.5e−231	0.4e−210	0.1e−81
f_7, 1.6							
IT	8	4	4	3	3	3	3
TNE	16	12	16	15	12	12	12
$\vert f \vert$	0.1e−238	0.2e−178	0.7e−210	0.2e−156	0.1e−371	0.1e−433	0.1e−400
f_8, 0.3							
IT	8	4	4	3	3	3	3
TNE	16	12	16	15	12	12	12
$\vert f \vert$	0.2e−243	0.7e−226	0.4e−246	0.1e−195	0.3e−466	0.1e−515	0.5e−884
f_9, 0.4							
IT	8	4	4	3	3	3	3
TNE	16	12	16	15	12	12	12
$\vert f \vert$	0.3e−84	0.3e−81	0.3e−143	0.2e−103	0.1e−144	0.6e−117	0.6e−665
f_{10}, 1.32							
IT	8	4	4	3	3	3	3
TNE	16	12	16	15	12	12	12
$\vert f \vert$	0.8e−3	0.1e−139	0.4e−26	Div.	0.6e−229	0.5e−247	0.4e−424

Table 3 Results of the family (40) with memory using (41).

Number of full iterations	COC	Zero	$\lvert f \rvert$
3	9.58678	0.56049912163979286993112824338688	1.2310×10^{-3274}

Table 4 Results of the family (40) with memory using (55).

Number of full iterations	COC	Zero	$\lvert f \rvert$
3	10.0028	0.56049912163979286993112824338688	2.7299×10^{-3606}

$$\begin{aligned}1/f'(\alpha) \approx\ & 1/(f(x_{new})(1/(-w+x_{new})+1/(x_{new}-x_{old})+1/(x_{new}\\ & -y)+1/(x_{new}-z))+(f(y)(w-x_{new})^2(w-x_{old})(x_{new}-x_{old})^2\\ & \times(w-z)(x_{new}-z)^2(x_{old}-z)-(x_{new}-y)^2(f(z)(w-x_{new})^2\\ & \times(w-x_{old})(x_{new}-x_{old})^2(w-y)(x_{old}-y)+(x_{new}-z)^2(y-z)\\ & \times(f(x_{old})(w-x_{new})^2(w-y)(w-z)+f(w)(x_{new}-x_{old})^2\\ & \times(x_{old}-y)(-x_{old}+z))))/((w-x_{new})(w-x_{old})(x_{new}-x_{old})(w-y)\\ & \times(x_{new}-y)(x_{old}-y)(w-z)(x_{new}-z)(x_{old}-z)(y-z))).\end{aligned} \tag{55}$$

Note that in this case the computational efficiency index would be $10^{1/4} \approx 1.7782$.

4. Numerical results

In this section, we check the effectiveness of the novel derivative-free classes (29) and (30), and also the family (40) with memory. Due to this, we have compared (27) and (35) with Steffensen's method (1), Liu et al. scheme (3), the fourth-order method of Cordero et al. (4), the sixth-order technique of Cordero et al. (5), and the optimal eighth-order family of Kung and Traub (6) with $\beta = 1$, using the examples listed in Table 1.

The results of comparisons are given in Table 2 in terms of the number significant digits for each test function after the specified number of iterations, that is, e.g. $0.1e-200$ shows that the absolute value of the given non-linear function (f_1) after three iterations is zero up to 200 decimal places.

In Table 2, Div. represents that the corresponding iterative method is divergent for the initial guess. As can be seen, numerical results are in concordance with the theory developed in this paper. In all the examples, the new methods improve the corresponding classical methods. Moreover, when we fix the same convergence criterion for all methods, the number of iterations and the number of functional evaluations are almost always better in the proposed eighth-order modified methods.

In Table 2, TNE denotes the Total Number of Evaluations required for a method to do the specified iterations. The new methods inherit the merit of the optimal fourth-order two-step methods without memory with regards to application of divided differences and high efficiency index, which is confirmed by the results in Table 2. According to Table 2, under a fair comparison structure, the proposed methods perform well.

Another observation from Table 2 is that, methods (4) and (5) are not efficient when the non-linear test functions are hard, their accuracy is not good as well as they need high number of function evaluations, in fact, although they consist of less operations (multiplication, addition, etc.) per full cycle, they include more function evaluations which is not at all good when the test non-linear functions have complicated structures.

It could also be inferred from Table 2 that the contributed methods are better than the existing optimal eighth-order derivative-free family (6).

It is clear the family (40) with memory with two forms (41) and (55) is better than the methods without memory. Hence, we only test it on a well-known oscillatory example $f(x) = (\sin(10x^2))\cosh(x)$, and report the results on Tables 3 and 4 to support the theoretics of Section 3. The initial approximation is chosen as 0.560507 while the stopping criterion is $\lvert f(x_n) \rvert \leqslant 10^{-1000}$, the initial guess for β_0 is 0.01, and the computational order of convergence (COC) has been computed by $COC = \frac{\ln \lvert f(x_n)/f(x_{n-1}) \rvert}{\ln \lvert f(x_{n-1})/f(x_{n-2}) \rvert}$.

5. Concluding comments

In this research, we have given two simple yet powerful three-step schemes without memory for solving non-linear equations, in which there is no need of derivative-calculation per cycle. The novel classes of iterative methods, which were obtained by considering the method of Liu et al. (3) in the first two-step of a three-step cycle reach the optimal efficiency index 1.682.

In comparisons, each test function was computed to establish the effectiveness of our contributed methods. Table 2 in Section 4, have manifested that the new derivative-free methods are robust for good initial approximations.

We have extended one of the schemes for multiple zeros and discusses on the with memory variants of the new families. The corespondent R-order for methods with memory have been found theoretical and established dramatically improvement in the computational efficiency index. 1.778 has been obtained as the highest possible computational efficiency index for the new methods with memory.

Acknowledgments

The author expresses his sincere thanks to the referee for careful reading and several helpful suggestions.

References

[1] T. Sauer, Numerical Analysis, second ed., Pearson, Boston, 2012.
[2] J.F. Steffensen, Remarks on iteration, Skand. Aktuarietidskr 16 (1933) 64–72.
[3] B.I. Yun, M.S. Petkovic, Iterative methods based on the Signum function approach for solving nonlinear equations, Numer. Algor. 52 (2009) 649–662.

[4] F. Soleymani, S. Shateyi, Two optimal eighth-order derivative-free classes of iterative methods, Abst. Appl. Anal. vol. 2012, Article ID 318165, 14 pp.
[5] F. Soleymani, D.K.R. Babajee, S. Shateyi, S.S. Motsa, Construction of optimal derivative-free techniques without memory, J. Appl. Math., vol. 2012, Article ID 497023, 24 pp.
[6] B. Neta, M.S. Petkovic, Construction of optimal order nonlinear solvers using inverse interpolation, Appl. Math. Comput. 217 (2010) 2445–2448.
[7] F. Soleymani, S. Shateyi, H. Salmani, Computing simple roots by an optimal sixteenth-order class, J. Appl. Math., vol. 2012, Article ID 958020, 13 pp.
[8] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, J. ACM 21 (1974) 643–651.
[9] Q. Zheng, P. Zhao, Z. Li, W. Ma, Variants of Steffensen-secant method and applications, Appl. Math. Comput. 216 (2010) 3486–3496.
[10] Z. Liu, Q. Zheng, P. Zhao, A variant of Steffensen's method of fourth-order convergence and its applications, Appl. Math. Comput. 216 (2010) 1978–1983.
[11] A. Cordero, J.L. Hueso, E. Martinez, J.R. Torregrosa, Steffensen type methods for solving nonlinear equations, J. Comput. Appl. Math. 236 (2012) 3058–3064.
[12] X. Li, C. Mu, J. Ma, L. Hou, Fifth-order iterative method for finding multiple roots of nonlinear equations, Numer. Algor. 57 (2011) 389–398.
[13] E. Schroder, Uber unendlich viele algorithmen zur Auflösung der Gleichungen, Math. Ann. 2 (1870) 317–365.
[14] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice Hall, New York, 1964.
[15] F. Soleymani, Novel computational iterative methods with optimal order for nonlinear equations, Adv. Numer. Anal., 2011, Article ID 270903, 10 pp.
[16] F. Soleymani, S.K. Khattri, S. Karimi Vanani, Two new classes of optimal Jarratt-type fourth-order methods, Appl. Math. Lett. 25 (2012) 847–853.
[17] F. Soleymani, S. Karimi Vanani, Optimal Steffensen-type methods with eighth order of convergence, Comput. Math. Appl. 62 (2011) 4619–4626.
[18] S. Wagon, Mathematica in Action, third ed., Springer, 2010.
[19] M. Trott, The Mathematica GuideBook for Numerics, Springer, 2006.