International Journal of
Mathematical Modelling & Computations
Vol. 04, No. 03, Summer 2014, 277- 288

A New Two Step Class of Methods with Memory for Solving Nonlinear Equations with High Efficiency Index

T. Lotfi[a,*] and P. Assari[a]

[a] *Department of Mathematics, Islamic Azad University, Hamedan Branch, Hamedan, Iran.*

Abstract. It is attempted to extend a two-step without memory method to its with memory. Then, a new two-step derivative free class of without memory methods, requiring three function evaluations per step, is suggested by using a convenient weight function for solving nonlinear equations. Eventually, we obtain a new class of methods by employing a self-accelerating parameter calculated in each iterative step applying only information from the current and the previous iterations, defining a with memory class.
Although these improvements are achieved without any additional function evaluations, the R-order of convergence are boosted from 4 to 5.24 and 6, respectively, and it is demonstrated that the proposed with memory classes provide a very high computational efficiency.
Numerical examples are put forward and the performances are compared with the basic two-step without memory methods.

Received: 6 October 2013, Revised: 22 December 2013, Accepted: 15 Janurary 2014.

Keywords: Nonlinear equation, With memory method, R-order of convergence, Self-accelerating parameter.

Index to information contained in this paper

1. Introduction

Finding the roots of nonlinear equations has significant applications in various fields of science and engineering. The main goal in constructing iterative methods for solving nonlinear equations is to attained as high as possible order of convergence with the lowest computational costs. Since 1960s', many methods have been

[*] Corresponding author. Email: lotfitaher@yahoo.com or lotfi@iauh.ac.ir

http://www.ijm2c.ir

constructed for solving nonlinear equations. Newton's method is the most famous one [4]

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

Due to avoid computing the derivative, Steffensen in [8] presents a derivative-free method as follows

$$x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \tag{2}$$

where $w_n = x_n + f(x_n), n = 0, 1, 2, ...$

Recently, many high order derivative-free methods have been built based on Steffensen-type methods inwhich they have an advantage over kinds of methods while derivatives are complicated to compute.

For the first time multi-step methods are appeared in Ostrowski's book [5] with the purpose of higher order convergence and efficiency. Ostrowski proposed the first two-step method as given by

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, & n = 0, 1, 2, ..., \\ x_{n+1} = y_n - \dfrac{f(y_n)}{f(x_n) - 2f(y_n)} \cdot \dfrac{f(x_n)}{f'(x_n)}. \end{cases} \tag{3}$$

A family of fourth-order methods free from any derivatives, satisfying Kung and Traub's conjecture, are established by Ren-Wu-Bi [6]

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \\ x_{n+1} = y_n - \dfrac{f(y_n)}{f[x_n, y_n] + f[y_n, w_n] - f[x_n, w_n] + \alpha(y_n - x_n)(y_n - w_n)}, \end{cases} \tag{4}$$

where $w_n = x_n + f(x_n), n = 0, 1, 2, ...$ and $\alpha \in \mathbb{R}$.

A variant of Steffensen's method of fourth-order convergence for solving nonlinear equations is suggested by Liu et al. [3]

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \\ x_{n+1} = y_n - \dfrac{f[x_n, y_n] - f[y_n, w_n] + f[x_n, w_n]}{f[x_n, y_n]^2} f(y_n), \end{cases} \tag{5}$$

wherein $w_n = x_n + f(x_n), n = 0, 1, 2, ...$

In the case of without memory methods, eventually the two-step class of methods are proposed by Soleymani [7]

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n - \beta f(x_n), \quad \beta \in \mathbb{R} - \{0\}, \\ x_{n+1} = y_n - \Big(\dfrac{f[x_n, y_n] - f[y_n, w_n] + f[x_n, w_n]}{f[x_n, y_n]^2} f(y_n)\Big) G(t) H(\varphi), \end{cases} \tag{6}$$

where $t = \dfrac{f(y_n)}{f(x_n)}$, $\varphi = \dfrac{f(y_n)}{f(w_n)}$, $n = 0, 1, 2, ...,$ and $H(t)$ and $G(\varphi)$ are two real valued weight functions that satisfy the conditions

$$G(0) = 1, G'(0) = 0, |G''(0)| < \infty, H(0) = 1, H'(0) = 0, |H''(0)| < \infty. \tag{7}$$

Moreover, its error equation is given by

$$\begin{aligned} e_{n+1} = x_{n+1} - \alpha = \frac{1}{2c_1^3} c_2(-1 + c_1\beta)(-2c_1c_3(-1 + c_1\beta) \\ + c_2^2(-4 + G''(0) + c_1\beta(2 + (-2 + c_1\beta)G''(0)) + H''(0)))e_n^4 + O(e_n^5), \end{aligned} \tag{8}$$

wherein $c_j = \dfrac{f^{(j)}(\alpha)}{j!}, j \geqslant 1$ and $e_n = x_n - \alpha$.

In the following of this paper, in Section 2 an extension of the family of two-step without memory methods (6) to its with memory is exhibited and then, we present a new optimal fourth-order family with a convenient weight function, deduce its error equation in Section 3, based on this family and by choosing the appropriate values of a self-accelerating parameter, we suggest a new two-step class of with memory methods in Section 4. As the main contribution of this paper, we establish that the R-order (By R-order we mean any real convergence order.) of both corresponding classes of with memory methods are increased from 4 to 5.24 and 6, respectively, using Newton's interpolatory polynomials of third degree. The new methods use only three evaluations of the functions per step, hence possess a very high computational efficiency. In Section 5 numerical examples and comparisons with existing methods are given to confirm the theoretical results.

2. Development of the Methods

We develop the class of two-step with memory methods relied on the two-step methods (6) using parameter $\beta = \beta_n$ by an approximation to $\dfrac{1}{f'(\alpha)}$ applying available data as follows

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n - \beta_n f(x_n), \quad \beta_n \in \mathbb{R} - \{0\}, \\ x_{n+1} = y_n - \Big(\dfrac{f[x_n, y_n] - f[y_n, w_n] + f[x_n, w_n]}{f[x_n, y_n]^2} f(y_n)\Big) G(t_n) H(\varphi_n), \end{cases} \tag{9}$$

where $t_n = \dfrac{f(y_n)}{f(x_n)}$, $\varphi_n = \dfrac{f(y_n)}{f(w_n)}$, $n = 0, 1, 2, ...$

According to the error equation (8), we find out that if the order of convergence of methods (6) wants to rise up, the coefficient of e_n^4 in (8) should be equal to zero. Due to this reason, we force $(-1 + c_1\beta) = 0$, where $c_1 = f'(\alpha)$.
However, since α is unknown, we have no information on the exact $f'(\alpha)$, thus, we approximate $f'(\alpha)$ with $N_3'(x_n)$. Although there are multifarious fashions to approximate $f'(\alpha)$, it seems that $N_3'(x_n)$ is the prime one. For converting methods (6) to with memory class, we consider

$$\beta_n = \frac{1}{f'(\alpha)} = \frac{1}{N_3'(x_n)}, \tag{10}$$

where $N_3(t) = N_3(t; x_n, y_{n-1}, w_{n-1}, x_{n-1})$ is Newton's interpolatory polynomial of third degree, set through four available approximations $(x_n, y_{n-1}, w_{n-1}, x_{n-1})$. The derivative $N_3'(t)$ at $t = x_n$ is calculated by using the following formula:

$$\begin{aligned} N_3'(x_n) = \Big[\frac{d}{dt}N_3(t)\Big]_{t=x_n} &= f[x_n, y_{n-1}] + f[x_n, y_{n-1}, x_{n-1}](x_n - y_{n-1}) \\ &+ f[x_n, y_{n-1}, x_{n-1}, w_{n-1}](x_n - y_{n-1})(x_n - x_{n-1}). \end{aligned} \tag{11}$$

Due to this development, we present following theorem:

Theorem 2.1 *If an initial approximation x_0 is sufficiently close to the zero of $f(x)$ and the parameter β_n in the iterative scheme* (9) *is recursively calculated by the formula given in* (10)*, then, the R-order of convergence of the method* (9) *is at least 5.24.*

Proof.

Let $\{x_n\}$ be a sequence of approximations generated by an iterative with memory method. If this sequence converges to the zero α of f with the R-order r, then we can write

$$e_{n+1} \sim D_n e_n^r, \quad e_n = x_n - \alpha, \tag{12}$$

where D_n tends to the asymptotic error constant D of iterative method as $k \to \infty$. Thus

$$e_{n+1} \sim D_n(D_{n-1}e_{n-1}^r)^r = D_n D_{n-1}^r e_{n-1}^{r^2}. \tag{13}$$

Moreover, we have

$$e_{n,w} = (1 + \beta f'(\alpha))e_n + O(e_n^2), \tag{14}$$

$$e_{n,y} = c_2(1 + \beta f'(\alpha))e_n^2 + O(e_n^3), \tag{15}$$

$$e_{n+1} = A_4(1 + \beta f'(\alpha))e_n^4 + O(e_n^5). \tag{16}$$

Considering the error relations (14)-(16) and (8) with the self-accelerating parameter $\beta = \beta_n$ we can write the corresponding error relations for the with memory method (9)

$$e_{n,w} \sim (1 + \beta_n f'(\alpha))e_n, \tag{17}$$

$$e_{n,y} \sim c_2(1 + \beta_n f'(\alpha))e_n^2, \tag{18}$$

$$e_{n+1} \sim A_4(1 + \beta_n f'(\alpha))e_n^4, \tag{19}$$

where $A_4(\alpha) = \frac{1}{2c_1^3}[-2c_1c_3(-1+c_1\beta)+c_2^2(-4+G''(0)+c_1\beta(2+(-2+c_1\beta)G''(0))+H''(0))]$.

In addition, we have [9]

$$1 + \beta_n f'(\alpha) \sim c_4 e_{n-1}e_{n-1,y}e_{n-1,w}. \tag{20}$$

Assume that the iterative sequences $\{y_n\}$ and $\{z_n\}$ have the R-orders p and q, respectively.
In a similar way of (12), we can write

$$e_{n,y} \sim E_n e_n^p \qquad , \qquad e_{n,w} \sim F_n e_n^q$$

Then, we obtain

$$e_{n,y} \sim E_n e_n^p \sim E_n(D_{n-1}e_{n-1}^r)^p = E_n D_{n-1}^p e_{n-1}^{rp}, \tag{21}$$

$$e_{n,w} \sim F_n e_n^q \sim F_n(D_{n-1}e_{n-1}^r)^q = F_n D_{n-1}^q e_{n-1}^{rq}. \tag{22}$$

Considering (12), (18), (20) and (21), we have

$$\begin{aligned} e_{n,y} &\sim c_2(1+\beta_n f'(\alpha))e_n^2 \sim c_2(c_4 e_{n-1}e_{n-1,y}e_{n-1,w})e_n^2 \\ &\sim c_4c_2 e_{n-1}E_{n-1}e_{n-1}^p F_{n-1}e_{n-1}^q(D_{n-1}e_{n-1}^r)^2 \\ &\sim c_4c_2E_{n-1}F_{n-1}D_{n-1}^2 e_{n-1}^{2r+p+q+1}. \end{aligned} \tag{23}$$

In the same way, by combining (12), (17), (20) and (22), we get

$$\begin{aligned} e_{n,w} &\sim (1+\beta_n f'(\alpha))e_n \sim (c_4 e_{n-1}e_{n-1,y}e_{n-1,w})e_n \\ &\sim c_4 e_{n-1}E_{n-1}e_{n-1}^p F_{n-1}e_{n-1}^q D_{n-1}e_{n-1}^r \\ &\sim c_4E_{n-1}F_{n-1}D_{n-1}e_{n-1}^{r+p+q+1}. \end{aligned} \tag{24}$$

Similarly, from (12), (19) and (20)-(22), we have

$$\begin{aligned} e_{n+1} &\sim A_4(1+\beta_n f'(\alpha))e_n^4 \sim A_4(c_4 e_{n-1}e_{n-1,y}e_{n-1,w})e_n^4 \\ &\sim A_4c_4 e_{n-1}E_{n-1}e_{n-1}^p F_{n-1}e_{n-1}^q(D_{n-1}e_{n-1}^r)^4 \\ &\sim A_4c_4E_{n-1}F_{n-1}D_{n-1}^4 e_{n-1}^{4r+p+q+1}. \end{aligned} \tag{25}$$

Comparing the exponents of e_{n-1} on the right hand sides of (21) - (23), (22) - (24), and (13) - (25), we form the system of three equations in p, q and r

$$\begin{cases} r^2-4r-p-q=1, \\ rp-2r-p-q=1, \\ rq-r-p-q=1. \end{cases}$$

Non-trivial solution of this system is $p = 1+\sqrt{5} \approx 3.24$, $q = \sqrt{5} \approx 2.24$ and $r = 3+\sqrt{5} \approx 5.24$. Eventually, the R-order of the with memory methods (9), when β_n is calculated by (10), is at least 5.24. □

3. New Derivative Free Two-Step Methods

Based on the following two-step optimal method (5)

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

where $w_n = x_n + f(x_n), n = 0, 1, 2, ...$ and its error equation

$$e_{n+1} = \frac{(1 + c_1)c_2((2 + c_1)c_2^2 - c_1(1 + c_1)c_3)}{c_1^3} e_n^4 + O(e_n^5), \tag{26}$$

we present a new class of forth-order methods by using backward approximation with a free parameter and a suitable two-valued weight function.
Therefore, we suggest

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \qquad w_n = x_n - \beta f(x_n), \qquad \beta \in \mathbb{R} - \{0\}, \\ x_{n+1} = y_n - (\dfrac{f[x_n, y_n] - f[y_n, w_n] + f[x_n, w_n]}{f[x_n, y_n]^2} f(y_n))H(t_n, \varphi_n), \end{cases} \tag{27}$$

where $H(t, \varphi)$ is a real two-valued weight function with $t_n = \frac{f(y_n)}{f(x_n)}$ and $\varphi_n = \frac{f(y_n)}{f(w_n)}, n = 0, 1, 2, ...$

Theorem 3.1 *Let $f : D \to \mathbb{R}$ be a sufficiently differentiable function with a simple root $\alpha \in D$ and $D \subseteq \mathbb{R}$ be an open interval. Then the new class of derivative-free methods (27) is fourth-order convergence when $H(0, 0) = 1, H_t(0, 0) = H_\varphi(0, 0) = 0, H_{tt}(0, 0) = H_{\varphi\varphi}(0, 0) = 2, H_{t\varphi}(0, 0) = 0$ and has the following error equation*

$$e_{n+1} = -c_2(1 + \beta f'(\alpha))^2(c_3 + c_2^2\beta f'(\alpha))e_n^4 + O(e_n^5), \tag{28}$$

where $e_n = x_n - \alpha, \quad n = 0, 1, 2, ...$

Proof.

Let $e_n = x_n - \alpha$. Introduce the abbreviations:

$$e_y = y_n - \alpha, \quad m = \beta f'(\alpha), \quad w_n = x_n - \beta f(x_n), \quad c_j = \frac{f^{(j)}(\alpha)}{j!f'(\alpha)} \ (j = 2, 3, ...).$$

Now we want to derive the error equation (28) of the class of two-step methods (27).

Using the Taylor expansion and taking into account $f(\alpha) = 0$, we have

$$f(x_n) = f'(\alpha)(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4) + O(e_n^5), \tag{29}$$

and

$$\begin{aligned} f(w) = f(x_n - \beta f(x_n)) = f'(\alpha)\Big(&(1-m)e_n + c_2(1-3m+m^2)e_n^2 \\ &+ (2c_2^2 m(-1+m) + c_3(1-4m+12m^2-m^3))e_n^3 \\ &+ (c_2^3 m^3 - c_2c_3 m(-1-5m+3m^2) + c_4(1-5m \\ &- 5m^2 + 6m^3 - 4m^4 + m^5))e_n^4\Big) + O(e_n^5). \end{aligned} \tag{30}$$

Then, we find

$$\begin{aligned} e_y = y_n - \alpha = e_n - &\frac{f(x_n)}{f[x_n, w_n]} \\ = c_2&(1-m)e_n^2 + (c_3(2-3m+m^2) - c_2^2(2-2m+m^2))e_n^3 \\ + (c_4&(3-6m-4m^2-m^3) + c_2^3(4-5m+3m^2-m^3) \\ + c_2&c_3(-7+10m-7m^2+2m^3))e_n^4 + O(e_n^5). \end{aligned} \tag{31}$$

By (31) we get

$$\begin{aligned} f(y_n) = f'(\alpha)\Big(&c_2(1-m)e_n^2 + (c_3(2-3m+m^2) - c_2^2(2-2m+m^2))e_n^3 \\ &+ (c_4(3-6m+4m^2-m^3) + c_2^3(5-7m+4m^2-m^3) \\ &+ c_2c_3(-7+10m-7m^2+2m^3))e_n^4\Big) + O(e_n^5). \end{aligned} \tag{32}$$

Therefore, we can write a two-valued function H occurring in (27) by Taylor's series about $(0,0)$ in the form

$$\begin{aligned} H(t_n, \varphi_n) =& H(0,0) + H_t(0,0)t_n + H_\varphi(0,0)\varphi_n + \frac{H_{tt}(0,0)}{2}t_n^2 \\ &+ H_{t\varphi}(0,0)t_n\varphi_n + \frac{H_{\varphi\varphi}(0,0)}{2}\varphi_n^2, \end{aligned} \tag{33}$$

then by considering the conditions $H(0,0) = 1, H_t(0,0) = H_\varphi(0,0) = 0, H_{tt}(0,0) = H_{\varphi\varphi}(0,0) = 2, H_{t\varphi}(0,0) = 0$ for $H(t_n, \varphi_n)$, ultimately, we gain the error relation as
$e_{n+1} = -c_2(1+\beta f'(\alpha))^2(c_3 + c_2^2\beta f'(\alpha))e_n^4 + O(e_n^5)$. □

Some derived methods form our suggested forms of the functions H are given below:
$H_1(t, \varphi) = 1 + t^2 + \varphi^2$

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \qquad w_n = x_n - \beta f(x_n), \qquad \beta \neq 0, \\ x_{n+1} = y_n - \Big(\frac{f[x_n,y_n] - f[y_n,w_n] + f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n)\Big) \\ \qquad \times \Big(1 + \big(\frac{f(y_n)}{f(x_n)}\big)^2 + \big(\frac{f(y_n)}{f(w_n)}\big)^2\Big). \end{cases} \tag{34}$$

$H_2(t, \varphi) = 1 + (t + \varphi)^2$

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \qquad w_n = x_n - \beta f(x_n), \qquad \beta \neq 0, \\ x_{n+1} = y_n - \Big(\dfrac{f[x_n, y_n] - f[y_n, w_n] + f[x_n, w_n]}{f[x_n, y_n]^2} f(y_n)\Big) \\ \qquad \times \Big(1 + (\dfrac{f(y_n)}{f(x_n)} + \dfrac{f(y_n)}{f(w_n)})^2\Big). \end{cases} \tag{35}$$

$H_3(t, \varphi) = 1 + (t - \varphi)^2$

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \qquad w_n = x_n - \beta f(x_n), \qquad \beta \neq 0, \\ x_{n+1} = y_n - \Big(\dfrac{f[x_n, y_n] - f[y_n, w_n] + f[x_n, w_n]}{f[x_n, y_n]^2} f(y_n)\Big) \\ \qquad \times \Big(1 + (\dfrac{f(y_n)}{f(x_n)} - \dfrac{f(y_n)}{f(w_n)})^2\Big). \end{cases} \tag{36}$$

$H_4(t, \varphi) = 1 + \dfrac{t^2}{2} + \varphi^2$

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \qquad w_n = x_n - \beta f(x_n), \qquad \beta \neq 0, \\ x_{n+1} = y_n - \Big(\dfrac{f[x_n, y_n] - f[y_n, w_n] + f[x_n, w_n]}{f[x_n, y_n]^2} f(y_n)\Big) \\ \qquad \times \Big(1 + \dfrac{\big(\frac{f(y_n)}{f(x_n)}\big)^2}{2} + (\dfrac{f(y_n)}{f(w_n)})^2\Big). \end{cases} \tag{37}$$

$H_5(t, \varphi) = \dfrac{1 + t^2}{1 - \varphi^2}$

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \qquad w_n = x_n - \beta f(x_n), \qquad \beta \neq 0, \\ x_{n+1} = y_n - \Big(\dfrac{f[x_n, y_n] - f[y_n, w_n] + f[x_n, w_n]}{f[x_n, y_n]^2} f(y_n)\Big) \\ \qquad \times \Big(\dfrac{1 + (\frac{f(y_n)}{f(x_n)})^2}{1 - (\frac{f(y_n)}{f(w_n)})^2}\Big). \end{cases} \tag{38}$$

$H_6(t, \varphi) = \dfrac{1 - t^2}{1 - \varphi^2}$

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \qquad w_n = x_n - \beta f(x_n), \qquad \beta \neq 0, \\ x_{n+1} = y_n - \Big(\dfrac{f[x_n, y_n] - f[y_n, w_n] + f[x_n, w_n]}{f[x_n, y_n]^2} f(y_n)\Big) \\ \qquad \times \Big(\dfrac{1 - (\frac{f(y_n)}{f(x_n)})^2}{1 - (\frac{f(y_n)}{f(w_n)})^2}\Big). \end{cases} \tag{39}$$

4. A New Class of Two-Step Methods with Memory

It is obvious from (28) that the order of convergence of the class (27) is four when $\beta \neq \frac{-1}{f'(\alpha)}$, if $\beta = \frac{-1}{f'(\alpha)}$, then, the order of the class would enhance. By considering $\beta = \beta_n$, we have

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n,w_n]}, \quad w_n = x_n - \beta_n f(x_n), \quad \beta_n \neq 0, \\ x_{n+1} = y_n - \left(\frac{f[x_n,y_n]-f[y_n,w_n]+f[x_n,w_n]}{f[x_n,y_n]^2} f(y_n)\right) H(t_n,\varphi_n), \end{cases} \tag{40}$$

where $t = \dfrac{f(y_n)}{f(x_n)}$, $\varphi = \dfrac{f(y_n)}{f(w_n)}, n = 0,1,2,....$ As before, we consider $N_3'(x_n)$ for approximating $f'(\alpha)$:

$$\beta_n = -\frac{1}{f'(\alpha)} = -\frac{1}{N_3'(x_n)}, \tag{41}$$

where $N_3(t) = N_3(t; x_n, y_{n-1}, w_{n-1}, x_{n-1})$ is Newton's interpolatory polynomial of third degree, set through four available approximations $(x_n, y_{n-1}, w_{n-1}, x_{n-1})$. That $N_3'(x_n)$ is calculated as (11).
According to this modifications, we bring forward a theorem underneath:

Theorem 4.1 *Let the varying parameter β_n in methods* (40) *is calculated by* (41). *If an initial approximation x_0 is sufficiently close to the zero α of f, then, the R-order of convergence of the with memory class* (40) *is at least 6.*

Proof.
In a similar way of the Theorem 2.1, let

$$e_{n,w} = (1 + \beta f'(\alpha))e_n + O(e_n^2), \tag{42}$$

$$e_{n,y} = c_2(1 + \beta f'(\alpha))e_n^2 + O(e_n^3), \tag{43}$$

$$e_{n+1} = A_4(1 + \beta f'(\alpha))^2 e_n^4 + O(e_n^5), \tag{44}$$

where $A_4 = -\dfrac{c_2}{2c_1^3}(2c_1c_3 + c_2^2(-1 + c_1\beta))$.
Corresponding error relations for the with memory method (40) are

$$e_{n,w} \sim (1 + \beta_n f'(\alpha))e_n, \tag{45}$$

$$e_{n,y} \sim c_2(1 + \beta_n f'(\alpha))e_n^2, \tag{46}$$

$$e_{n+1} \sim A_4(1 + \beta_n f'(\alpha))^2 e_n^4. \tag{47}$$

Let the iterative sequences $\{y_n\}$ and $\{z_n\}$ have the R-orders p and q, respectively, then we have

$$e_{n,y} \sim E_n e_n^p \qquad , \qquad e_{n,w} \sim F_n e_n^q$$

Therefore, we obtain

$$e_{n,y} \sim E_n e_n^p \sim E_n(D_{n-1}e_{n-1}^r)^p = E_n D_{n-1}^p e_{n-1}^{rp}, \tag{48}$$

$$e_{n,w} \sim F_n e_n^q \sim F_n(D_{n-1}e_{n-1}^r)^q = F_n D_{n-1}^q e_{n-1}^{rq}. \tag{49}$$

Using (12), (20), (46) and (48), we have

$$\begin{aligned} e_{n,y} &\sim c_2(1+\beta_n f'(\alpha))e_n^2 \sim c_2(c_4 e_{n-1}e_{n-1,y}e_{n-1,w})e_n^2 \\ &\sim c_4 c_2 e_{n-1}E_{n-1}e_{n-1}^p F_{n-1}e_{n-1}^q (D_{n-1}e_{n-1}^r)^2 \\ &\sim c_4 c_2 E_{n-1}F_{n-1}D_{n-1}^2 e_{n-1}^{2r+p+q+1}. \end{aligned} \tag{50}$$

In the same way, according to (12), (20), (45) and (49), we get

$$\begin{aligned} e_{n,w} &\sim (1+\beta_n f'(\alpha))e_n \sim (c_4 e_{n-1}e_{n-1,y}e_{n-1,w})e_n \\ &\sim c_4 e_{n-1}E_{n-1}e_{n-1}^p F_{n-1}e_{n-1}^q D_{n-1}e_{n-1}^r \\ &\sim c_4 E_{n-1}F_{n-1}D_{n-1}e_{n-1}^{r+p+q+1}. \end{aligned} \tag{51}$$

From (12), (20) and (47)-(49), the error relation is achieved as follows

$$\begin{aligned} e_{n+1} &\sim A_4(1+\beta_n f'(\alpha))^2 e_n^4 \sim A_4(c_4 e_{n-1}e_{n-1,y}e_{n-1,w})^2 e_n^4 \\ &\sim A_4 c_4^2 e_{n-1}^2 (E_{n-1}e_{n-1}^p)^2 (F_{n-1}e_{n-1}^q)^2 (D_{n-1}e_{n-1}^r)^4 \\ &\sim A_4 c_4^2 E_{n-1}^2 F_{n-1}^2 D_{n-1}^4 e_{n-1}^{4r+2p+2q+2}. \end{aligned} \tag{52}$$

Contrast the exponents of e_{n-1} on the right hand sides of (48) and (50), (49) and (51), and then (13) and (52), we organize the system of three equations in p, q and r

$$\begin{cases} r^2-4r-2p-2q=2, \\ rp-2r-p-q=1, \\ rq-r-p-q=1. \end{cases}$$

Non-trivial solution of this system is $p=3$, $q=2$ and $r=6$. Ultimately, the R-order of the with memory method (40), when β_n is calculated by (41), is at least 6. □

5. Performances and Comparisons

Now we show the convergence behavior of the with memory methods (9) and (40). All computations are performed using the programming package *Mathematica*. We calculate the computational order of convergence (r_c) [9], using the formula

$$r_c = \frac{\log|f(x_n)/f(x_{n-1})|}{\log|f(x_{n-1})/f(x_{n-2})|}. \tag{53}$$

For demonstration, we use the following example:

$$f(x) = e^{x^2+x\ \cos x-1}\sin\ \pi x + x\ \log(x\ \sin x+1), \alpha=0, x_0=0.6, \beta_0=-0.1$$

The errors $|x_n-\alpha|$ for the first three iterations are given in Table 1 where the denotation $A(-h)$ means $A\times 10^{-h}$.

Table 1. Numerical results

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	r_c
$G_1(t)=1+t^2,\ H_1(\varphi)=1+\varphi^2$				
Soleymani′s methods (6)	1.27(−2)	3.68(−10)	1.93(−40)	4.000
Development of Soleymani′s methods (9)	1.80(−1)	2.41(−5)	2.43(−28)	6.000
$G_2(t)=1+t^2+t^3,\ H_2(\varphi)=1+\varphi^2+\varphi^3$				
Soleymani′s methods (6)	2.25(−2)	5.45(−9)	9.34(−36)	4.000
Development of Soleymani′s methods (9)	2.08(−1)	3.95(−5)	5.03(−28)	5.966
$G_3(t)=(1+t^2)/(1-t^2),\ H_3(\varphi)=(1+\varphi^2)/(1-\varphi^2)$				
Soleymani′s methods (6)	5.18(−2)	1.80(6)	1.81(−24)	4.017
Development of Soleymani′s methods (9)	7.07(−2)	8.53(−6)	5.33(−26)	5.003

Table 2. Numerical results

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	r_c
$H_1(t,\varphi)=1+t^2+\varphi^2$				
New without memory methods (27)	1.22(−2)	3.10(−10)	9.75(−41)	4.012
New with memory methods (40)	1.85(−1)	2.53(−5)	3.31(−28)	6.038
$H_2(t,\varphi)=1+(t+\varphi)^2$				
New without memory methods (27)	3.92(−3)	8.63(−12)	2.10(−46)	4.000
New with memory methods (40)	1.19(−1)	3.39(−5)	4.23(−27)	6.000
$H_3(t,\varphi)=1+(t-\varphi)^2$				
New without memory methods (27)	2.83(−2)	5.66(−8)	6.08(−31)	4.021
New with memory methods (40)	2.51(−1)	1.28(−4)	5.22(−24)	6.072
$H_4(t,\varphi)=1+\dfrac{t^2}{2}+\varphi^2$				
New without memory methods (27)	1.08(−2)	1.54(−10)	4.95(−42)	4.011
New with memory methods (40)	2.03(−1)	2.43(−5)	3.21(−28)	6.000
$H_5(t,\varphi)=\dfrac{1+t^2}{1-\varphi^2}$				
New without memory methods (27)	1.81(−2)	1.69(−9)	8.67(−38)	4.020
New with memory methods (40)	1.75(−1)	2.91(−5)	5.92(−28)	6.000
$H_6(t,\varphi)=\dfrac{1-t^2}{1-\varphi^2}$				
New without memory methods (27)	1.13(−2)	7.20(−11)	8.09(−44)	4.016
New with memory methods (40)	2.57(−1)	4.21(−5)	5.263(−27)	6.000

Note that in a real situation, α is not available, so one should consider the errors $|x_{n+1}-x_n|$ instead of $|x_n-\alpha|$.

When β_n is calculated by (10) to comparison with basic class (6) the results are shown in Table 1 and the weight functions in this table are selected from [7]. In Table 2, we introduce some proper two-valued weight functions then by applying them in these methods and observing results, afterwards comparing them with the basic without memory class (27) when β_n is computed by (41), the R-order of convergence of the with memory methods (40) are confirmed.

According to the results presented in Table 1 and a number of numerical examples, we can conclude that the convergence behavior of the two-step with memory methods (4), based on the self-acclerating parameter β_n is considerably better than the other (1) without memory.

6. Conclusion

Per iteration the developed methods (9) and (40) require evaluations of only three functions. Consider the efficiency index (e.g [1, 5, 9]) defined as $p^{\frac{1}{w}}$, where p is the order of the method and w is the number of function evaluations per iteration by the method. The efficiency index of the with memory methods (9) and (40) are $(5.24)^{\frac{1}{3}} \approx 1.73961$ and $6^{\frac{1}{3}} \approx 1.81712$, respectively, which are better than those without memory (6) and (27) ($4^{\frac{1}{3}} \approx 1.5874$).

By the theoretical analysis and numerical experiments, we confirm that the new with memory methods achieve higher order of convergence and without any additional function evaluations, the R-order of convergence is boosted at least 50 per cent. It is observed that the new methods have better performances.
It is worth mentioning that the developed methods do not work for multiple root(s) efficiently and likely they have linear behaviour. So, one can consider it as a new research problem. In other words, the proposed methods must be modified in such a way that attain as high as possible convergence order or efficiency index.

Acknowledgements

We wish to sincerely thank the anonymous referee for his/her recommendations, which have helped to the readability of this paper.

References

[1] Gautschi. W, Numerical Analysis: An Introduction, Birkhauser, Boston, 1997.
[2] Kung. H. T and Traub. J. F, Optimal Order of One-Point and Multipoint Iteration, J. ACM. **21** (1974) 643-651.
[3] Liu. Z, Zheng. Q and Zhao. P, A variant of Steffensen's method of fourth-order convergence and its applications, Appl, Math, Comput. **216** (2010) 1978-1983.
[4] Ortega. J. M and W. G. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
[5] Ostrowski. A. M, Solution of Equations and Systems of Equations, Academic Press, New York, 1960.
[6] Ren. H, Wu. Q and Bi. W, A class of two-step Steffensen type methods with fourth-order convergence, Appl, Math, Comput. **209** (2009) 206-210.
[7] Soleymani. F, On a Novel Optimal Quartically Class of Methods, Far East Journal of Mathematical Sciences. **58** (2011) 199-206.
[8] Steffensen. J. F, Remarks on iteration, Skand. Aktuarietidskr. **16** (1933) 64-72.
[9] Traub. J. F, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.