

# Adaptive Steffensen-like Methods with Memory for Solving Nonlinear Equations with the Highest Possible Efficiency Indices

M. J. Lalehchini <sup>\*</sup>, T. Lotfi <sup>†‡</sup>, K. Mahdiani <sup>§</sup>

Received Date: 2018-11-14    Revised Date: 2019-05-06    Accepted Date: 2019-07-06

## Abstract

The primary goal of this work is to introduce two adaptive Steffensen-like methods with memory of the highest efficiency indices. In the existing methods, to improve the convergence order applied to memory concept, the focus has only been on the current and previous iteration. However, it is possible to improve the accelerators, considering the time from the first to the current iterations. Therefore, we achieve superior convergence orders and obtain as high efficiency indices as possible. These are the main contributions of this work.

*Keywords* : Nonlinear equations; Iterative methods; Steffensen-like method; Methods with memory; Adaptive methods; R-order.

## 1 Introduction

One of the most important subjects in developing numerical algorithms is to establish optimal algorithms with economic complexity. For example, developing iterative methods for approximating zero(s) of a given nonlinear equation falls within this matter, and many studies have been devoted to it [9, 12]. Inspired by this, we will set up two adaptive Steffensen-like methods with memory which are improvement of the existing methods [1, 5, 6, 12, 16, 22]. To our knowledge, these kinds of adaptive methods have not

been studied in the literature. Traub developed the first method with memory from Steffensen's method [5] as following [11]:

$$\begin{cases} w_k = x_k + \gamma_k f(x_k), \\ x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \\ \gamma_{k+1} = -\frac{1}{N_1'(x_{k+1})}, \end{cases} \quad k = 0, 1, 2, \dots, \quad (1.1)$$

where  $x_0$  and  $\gamma_0$  are given initially suitable values, and  $N_1(t) = f(x_{k+1}) + (t - x_{k+1})f[x_{k+1}, x_k]$  is the linear Newton's interpolation. The convergence order of the with-memory method (1.1) is  $1 + \sqrt{2} \approx 2.414$ . Also, Džunnić and Petković improved Traub's idea, introducing a better accelerator [12]:

$$\begin{cases} w_k = x_k + \gamma_k f(x_k), \\ x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \\ \gamma_{k+1} = -\frac{1}{N_2'(x_{k+1})}, \end{cases} \quad k = 0, 1, 2, \dots, \quad (1.2)$$

<sup>\*</sup>Department of Applied Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran.

<sup>†</sup>Corresponding author. lotfi@iauh.ac.ir, Tel: +98(918)8121361.

<sup>‡</sup>Department of Applied Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran.

<sup>§</sup>Department of Applied Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran.

where  $x_0$  and  $\gamma_0$  are given initially suitable values, and  $N_2(t)$  is the Newton's interpolation polynomial given by  $N_2(t) = f(x_{k+1}) + (t - x_{k+1})f[x_{k+1}, w_k] + (t - x_{k+1})(t - w_k)f[x_{k+1}, w_k, x_k]$ .

The convergence order of the method with memory (1.2) is  $1 + \sqrt{4} = 3$ . Moreover, Džunić added another parameter to the Steffensen's method and obtained a more efficient method with memory [9]:

$$\begin{cases} x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k] + \lambda_k f(w_k)}, \\ \gamma_{k+1} = -\frac{1}{N'_2(x_{k+1})}, \quad k = 0, 1, 2, \dots, \\ w_{k+1} = x_{k+1} + \gamma_{k+1} f(x_{k+1}), \\ \lambda_{k+1} = \frac{-N'_3(w_{k+1})}{2N'_3(w_{k+1})}, \end{cases} \tag{1.3}$$

where  $x_0, \gamma_0$ , and  $\lambda_0$  are given initially suitable values. This method has the convergence order  $\frac{3+\sqrt{17}}{2} \approx 3.56$ .

**Remark 1.1** If  $\gamma$  and  $\lambda$  are constants, then methods (1.2) and (1.3) are without memory methods with convergence order two having the following error equations, respectively:

$$e_{k+1} = c_2(1 + \gamma f'(\alpha))e_k^2 + O(e_k^3), \tag{1.4}$$

and

$$e_{k+1} = (c_2 + \lambda)(1 + \gamma f'(\alpha))e_k^2 + O(e_k^3). \tag{1.5}$$

In this work, we will attempt to carry out two adaptive methods with memory regardless of (1.2) and (1.3), which are superior [2, 14, 20, 24]. To achieve this end, first, the accelerator parameter  $\gamma_k$  is updated with the existing information in the previous and current iterations. We prove that this method has convergence order 3.4 using the same function evaluations as (1.2), so its efficiency index is much better. Similarly, we derive another adaptive method with memory for (1.3) which acquires convergence order 3.9 using the same functional evaluations. Therefore, this method is better than both our adaptive method with one accelerator and all the existing methods.

## 2 Developing adaptive with memory methods

This section deals with two new adaptive methods with memory. To this end, we modify and ex-

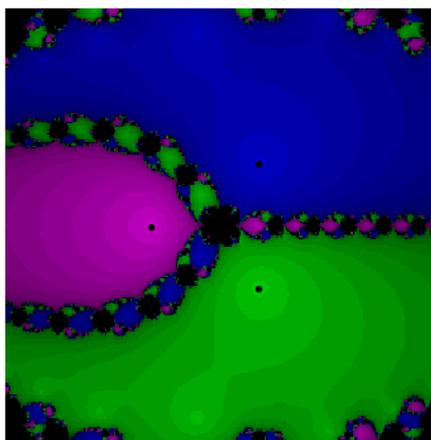
tend methods (1.2) and (1.3) in such a way that they consider all previous information to attain as high as possible convergence order without any new functional evaluation. In this manner, we use the adaptive idea which has not been considered to our best knowledge.

### 2.1 Mono accelerator adaptive with memory method

In (1.2), to update accelerator  $\gamma_k$  in each iteration, we only use the information from the current and previous iterations and reach the convergence order 3. However, the procedure goes ahead, the old information of current and previous steps can be used. In another words, we tend to apply the adaptive idea to construct the new methods with memory. Accordingly, we introduce the following new adaptive method with memory

$$\begin{cases} w_k = x_k + \gamma_k f(x_k), \\ x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad k = 0, 1, 2, \dots, \\ \gamma_{k+1} = -\frac{1}{N'_{2k+2}(x_{k+1})}, \end{cases} \tag{2.6}$$

where  $x_0$  and  $\gamma_0$  are given initially suitable values, and  $N_{2k+2}(t)$  is Newton's interpolation polynomial of degree  $2k + 2$  at the points  $x_{k+1}, w_k, x_k, \dots, w_0, x_0$ . Referring to



**Figure 1:** Dynamical Planes for (2.6) for  $\gamma = 0.1$ .

the Error Equation (1.4), it is observed that if  $1 + \gamma f'(\alpha) = 0$ , then convergence order of the method without memory (2.6), increases if for a moment we supposed that  $\gamma_k$  is fixed [22]. Since  $\alpha$  is unknown, we cannot suppose  $\gamma = -1/f'(\alpha)$ .

**Table 1:** Test functions for  $\gamma = 0.1, \lambda = 0.1$ .

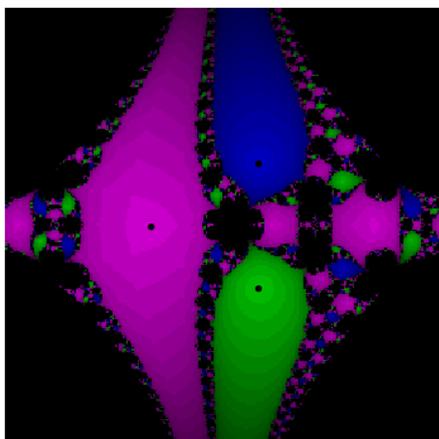
Example	$x_0$	$\alpha$
$f_1(t) = e^{(t^2-4)} + \sin(t-2) - t^4 + 15$	2.50	2.00
$f_2(t) = \frac{1}{t^4} - t^2 - \frac{1}{t} + 1$	2.00	1.00
$f_3(t) = (t-2)(t^{10} + t + 2)e^{-5t}$	2.40	2.00
$f_4(t) = e^{(t^2-3t)} + \sin(t) + \log(t^2 + 1)$	0.45	0.00

**Table 2:** Test functions for  $\gamma = 0.1, \lambda = 0.1$ .

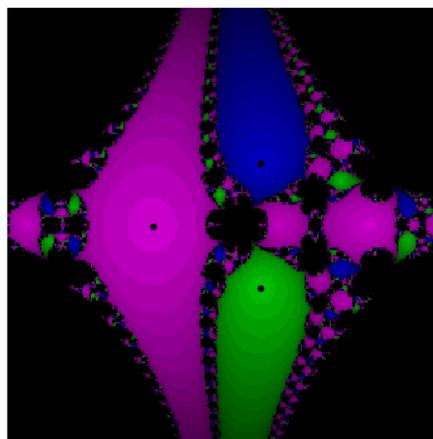
Function.	$ x_1 - \alpha $	$ x_2 - \alpha $	$ x_3 - \alpha $	COC
$f_1$	0.2371(0)	0.9503(-2)	0.1967(-7)	4.0683
$f_2$	0.3417(-1)	0.9531(-3)	0.1021(-8)	3.8404
$f_3$	0.2024(0)	0.2490(-2)	0.8749(-10)	3.9025
$f_4$	0.2277(-1)	0.4709(-3)	0.3300(-9)	3.6537

**Table 3:** Results of (2.25) for different test functions.

Function.	$ x_1 - \alpha $	$ x_2 - \alpha $	$ x_3 - \alpha $	COC
$f_1$	0.2902(0)	0.1028(-1)	0.1581(-7)	4.0078
$f_2$	0.9073(-1)	0.2554(-2)	0.2200(-8)	3.9115
$f_3$	0.1681(0)	0.1006(-2)	0.1582(-11)	3.9603
$f_4$	0.4153(-1)	0.4718(-3)	0.5529(-11)	4.0784



**Figure 2:** Dynamical Planes for (2.6) for  $\gamma = -0.31093$ .



**Figure 3:** Dynamical Planes for (2.6) for  $\gamma = -0.33330$ .

Even if we assumed that  $\alpha$  was known, we could not use it to evaluate  $f'(\alpha)$ , since it increased the functional evaluation, and optimality of the methods would be destroyed. It is assumed that the sequence  $\{x_k\}$  converges to  $\alpha$ . Moreover,  $f'$  is at least continuous, so there is  $\lim_{k \rightarrow \infty} f'(x_k) = f'(\alpha)$  as  $k \rightarrow \infty$ . Thus, we can use  $N'_{2k+2}(x_k)$  instead of  $f'(x_k)$  to our mission, i.e.,  $\gamma_k = -1/N'_{2k+2}(x_k)$ .

To discuss the convergence order of (2.6), we need:

**Lemma 2.1** If  $\gamma_{k+1} = -1/N'_{2k+2}(x_k)$ , and

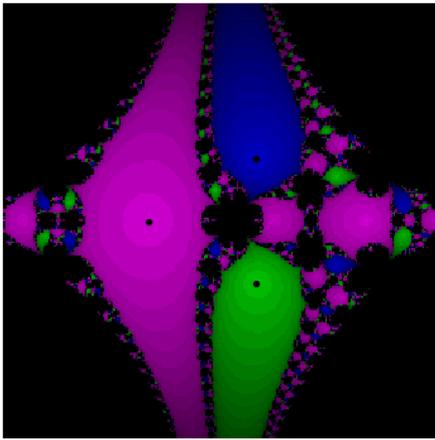


Figure 4: Dynamical Planes for (2.6) for  $\gamma = 0.1$

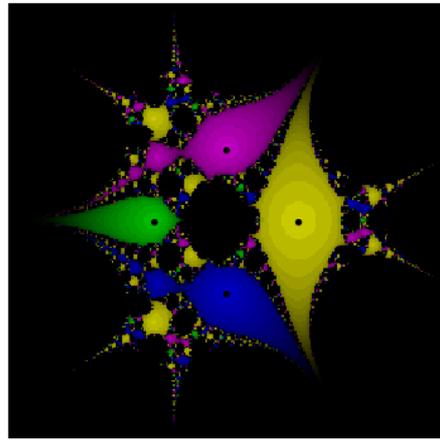


Figure 6: Dynamical Planes for (2.6) for  $\gamma = -0.24975$

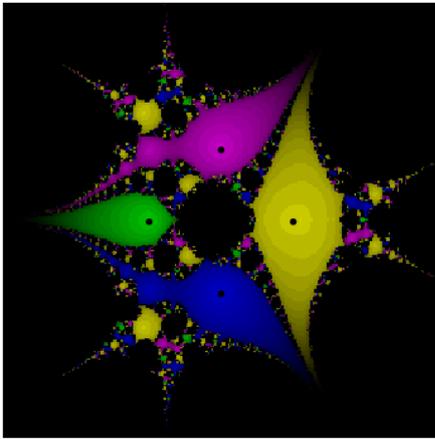


Figure 5: Dynamical Planes for (2.6) for  $\gamma = -0.21435$

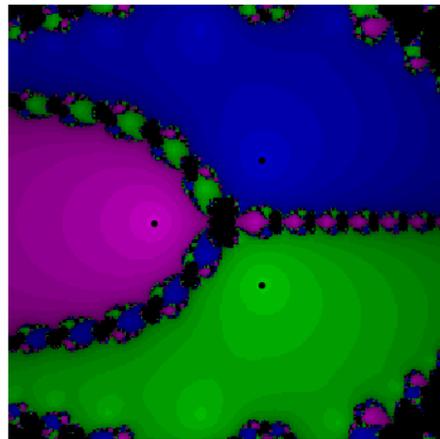


Figure 7: Dynamical Planes for (2.25) for  $\gamma = 0.1, \lambda = 0.1$ .

$\lambda_{k+1} = \frac{-N''_{2k+3}(w_{k+1})}{2N'_{2k+3}(w_{k+1})}$  then

$$1 + \gamma f'(\alpha) \sim \prod_{i=0}^k e_{w,i} e_i, \quad c_2 + \lambda \sim \prod_{i=0}^k e_{w,i} e_i, \tag{2.7}$$

where  $e_i = x_i - \alpha$  and  $e_{w,i} = w_i - \alpha$ .

**Proof.** According to Newton's interpolation formula for nodes  $t_0, t_1, \dots, t_s$ , we have

$$f(t) - N_s(t) = \frac{f^{(s+1)}(\xi)}{(s+1)!} \prod_{i=0}^s (t - t_i). \tag{2.8}$$

where  $s \in [\min \{t_0, t_1, \dots, t_s\}, \max \{t_0, t_1, \dots, t_s\}]$  and

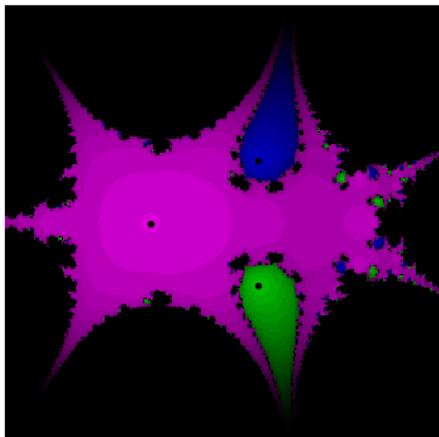
$$\gamma = -\frac{1}{f'(\alpha)} \simeq -\frac{1}{N'_{2k+2}(x_k)} = \gamma_{k+1}. \tag{2.9}$$

By differentiating (2.8) and setting  $t = x_{k+1}$  like proving Lemma 1 in [12], we have

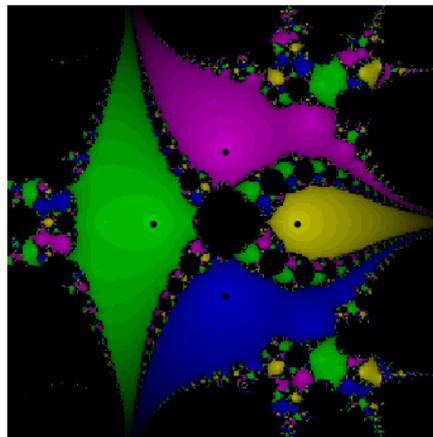
$$\begin{aligned} N'_{2k+2}(x_{k+1}) &= f'(x_{k+1}) - \frac{f^{(2k+3)}(\xi)}{(2k+3)!} \times \\ &\prod_{i=0}^k (x_{k+1} - x_i)(x_{k+1} - w_i) \\ &\sim f'(\alpha) (1 + c_{2k+3} \prod_{i=0}^k e_{w,i} e_i). \end{aligned} \tag{2.10}$$

Consequently,

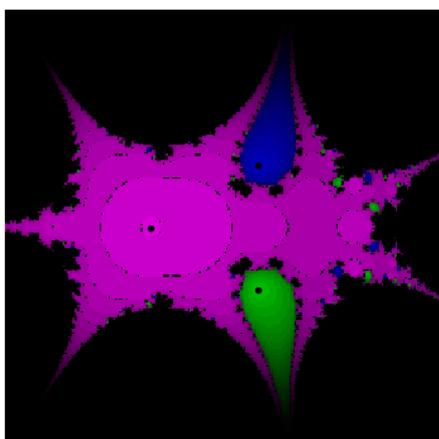
$$\begin{aligned} 1 + \gamma_{k+1} f'(\alpha) &= 1 - \frac{f'(\alpha)}{N'_{2k+2}(x_{k+1})} = \\ 1 - \frac{1}{1 - c_{2k+3} \prod_{i=0}^k e_{w,i} e_i} &\sim \prod_{i=0}^k e_{w,i} e_i. \end{aligned} \tag{2.11}$$



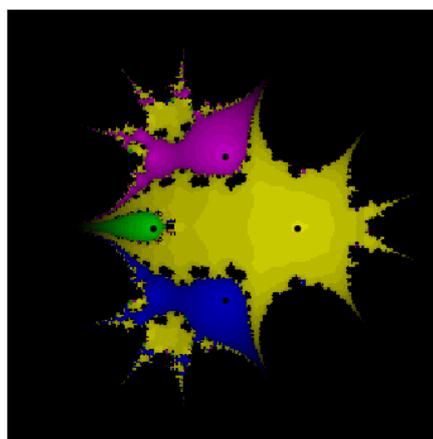
**Figure 8:** Dynamical Planes for (2.25) for  $\gamma = -0.31336, \lambda = 0.99909$ .



**Figure 10:** Dynamical Planes for (2.25) for  $\gamma = 0.1, \lambda = 0.1$



**Figure 9:** Dynamical Planes for (2.25) for  $\gamma = -0.33334, \lambda = 1$ .



**Figure 11:** Dynamical Planes for (2.25) for  $\gamma = -0.21210, \lambda = -1.46997$

To prove the second relation with respect to (2.8),(2.10)and Lemma1 in [9] we have

$$c_2 + \lambda_{k+1} = \frac{f''(\alpha)}{2f'(\alpha)} - \frac{N''_{2k+3}(w_{k+1})}{2N'_{2k+3}(w_{k+1})} \sim \prod_{i=0}^k e_{w,i} e_i \tag{2.12}$$

**Theorem 2.1** *Let the initial approximation  $x_0$  be sufficiently close to the zero  $\alpha$  of  $f$ . Also  $R$  and  $p$  denote the convergence order of the sequences  $\{x_k\}$  and  $\{w_k\}$ , respectively, as obtained in adaptive method with memory (2.6). Then, we will have*

$$\begin{cases} R^k p - R^k - (p + 1) \sum_{i=0}^{k-1} R^i = 0, \\ R^{k+1} - R^k - (p + 1) \sum_{i=0}^{k-1} R^i = 0. \end{cases} \tag{2.13}$$

**Proof.** We can assume

$$e_{k+1} \sim e_k^R. \tag{2.14}$$

Hence,

$$e_{k+1} \sim (e_{k-1}^R)^R = e_{k-1}^{R^2}. \tag{2.15}$$

Inductively,

$$e_{k+1} \sim e_0^{R^{k+1}}. \tag{2.16}$$

Similarly, we have

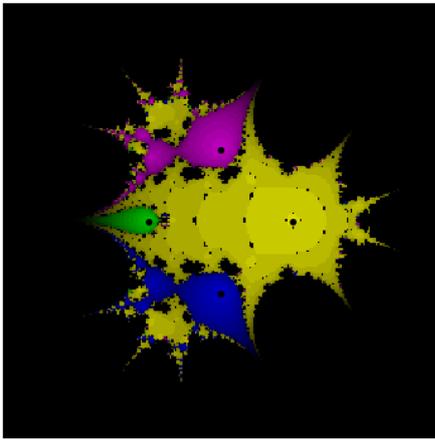
$$e_{w,k} \sim e_k^p = (e_{k-1}^R)^p = e_{k-1}^{R^p}. \tag{2.17}$$

Thus,

$$e_{w,k} \sim e_0^{R^k p}. \tag{2.18}$$

By (2.14) and (2.17), Lemma 2.1 results in

$$1 + \gamma f'(\alpha) \sim e_0^{(p+1) \sum_{i=0}^{k-1} R^i}. \tag{2.19}$$



**Figure 12:** Dynamical Planes for (2.25) for  $\gamma = -0.25005, \lambda = -1.49993$

On the other hand, since  $e_{w,k} \sim (1 + \gamma f'(\alpha))e_k$  and  $e_{k+1} \sim (1 + \gamma f'(\alpha))e_k^2$ , taking into account (2.19), we have

$$e_{w,k} \sim e_0^{R^k + (p+1) \sum_{i=0}^{k-1} R^i}, \tag{2.20}$$

and

$$e_{k+1} \sim e_0^{2R^k + (p+1) \sum_{i=0}^{k-1} R^i}. \tag{2.21}$$

From (2.18)-(2.20) and (2.16)-(2.21), we conclude that

$$e_0^{R^k p} \sim e_0^{R^k + (p+1) \sum_{i=0}^{k-1} R^i}, \tag{2.22}$$

$$e_0^{R^{k+1}} \sim e_0^{2R^k + (p+1) \sum_{i=0}^{k-1} R^i}. \tag{2.23}$$

Consequently,

$$\begin{cases} R^k p - R^k - (p+1) \sum_{i=0}^{k-1} R^i = 0, \\ R^{k+1} - 2R^k - (p+1) \sum_{i=0}^{k-1} R^i = 0. \end{cases} \tag{2.24}$$

### 2.2 Bi accelerators adaptive with memory method

We now introduce bi accelerators adaptive with memory method [7, 14, 19, 20, 21, 24]. Since most of the details are similar to the descriptions of (2.6), we confine ourselves to repeat them. We consider the following two new accelerators adaptive with-memory method:

$$\begin{cases} x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k] + \lambda_k f(w_k)}, \\ \gamma_{k+1} = -\frac{1}{N'_{2k+2}(x_{k+1})}, \quad k = 0, 1, 2, \dots, \\ w_{k+1} = x_{k+1} + \gamma_{k+1} f(x_{k+1}), \\ \lambda_{k+1} = \frac{-N'_{2k+3}(w_{k+1})}{2N'_{2k+3}(w_{k+1})}, \end{cases} \tag{2.25}$$

where  $x_0, \gamma_0$  and  $\lambda_0$  are given suitably. Then, we have

**Theorem 2.2** *Let the initial approximation  $x_0$  be sufficiently close to the zero  $\alpha$  of  $f$ . Also  $R$  and  $p$  denote the convergence order of the sequences  $\{x_k\}$  and  $\{w_k\}$ , respectively, as obtained in adaptive method with memory (2.25). Then, we will have*

$$\begin{cases} R^k p - R^k - (p+1) \sum_{i=0}^{k-1} R^i = 0, \\ R^{k+1} - 2R^k - 2(p+1) \sum_{i=0}^{k-1} R^i = 0. \end{cases} \tag{2.26}$$

**Proof.** We can assume

$$e_{k+1} \sim e_k^R. \tag{2.27}$$

Hence,

$$e_{k+1} \sim (e_{k-1}^R)^R = e_{k-1}^{R^2}. \tag{2.28}$$

Inductively,

$$e_{k+1} \sim e_0^{R^{k+1}}. \tag{2.29}$$

Similarly, we have

$$e_{w,k} \sim e_k^p = (e_{k-1}^R)^p = e_{k-1}^{R p}. \tag{2.30}$$

Thus,

$$e_{w,k} \sim e_0^{R^k p}. \tag{2.31}$$

By (2.14) and (2.17), Lemma 2.1 results

$$(c_2 + \lambda)(1 + \gamma f'(\alpha)) \sim e_0^{2(p+1) \sum_{i=0}^{k-1} R^i}. \tag{2.32}$$

By (1.5) and  $e_{w,k} \sim (1 + \gamma f'(\alpha))e_k$ , taking into account (2.32), we have

$$e_{w,k} \sim e_0^{R^k + (p+1) \sum_{i=0}^{k-1} R^i}, \tag{2.33}$$

and

$$e_{k+1} \sim e_0^{2R^k + 2(p+1) \sum_{i=0}^{k-1} R^i}. \tag{2.34}$$

From (2.31)-(2.33) and (2.29)-(2.34), we conclude that

$$e_0^{R^k p} \sim e_0^{R^k + (p+1) \sum_{i=0}^{k-1} R^i}, \tag{2.35}$$

$$e_0^{R^{k+1}} \sim e_0^{2R^k + 2(p+1) \sum_{i=0}^{k-1} R^i}. \tag{2.36}$$

Consequently,

$$\begin{cases} R^k p - R^k - (p+1) \sum_{i=0}^{k-1} R^i = 0, \\ R^{k+1} - 2R^k - 2(p+1) \sum_{i=0}^{k-1} R^i = 0. \end{cases} \tag{2.37}$$

### 3 Numerical Computations

In this section, to show the efficiency of the new adaptive methods with memory (2.6) and (2.25), we report their numerical results. To this end, among many tested problems, we confine to report the results of four test functions (See Table 1). Moreover, the initial values are given in Tables 1. It should be noted that  $|x_k - \alpha|$  shows the error in each iteration and  $a(b)$  stands for  $a^b$ , and the computational order of convergence (COC) can be approximated by the following formula:

$$COC \approx \frac{\log \frac{|x_{k+1}-\alpha|}{|x_k-\alpha|}}{\log \frac{|x_k-\alpha|}{|x_{k-1}-\alpha|}}.$$

Tables 2 and 3 report the numerical implementations of the adaptive methods with memory in this work. Table 2 shows that the numerical results support the developed theory for method (2.6). As can be seen in Table 3, the (COC) of the tested functions using the method (2.25) is very good, and has the highest amount. Indeed, there is not any work in the literature that could compete with method (2.25). On the other hand, this is the highest and the best efficiency index which comes from the point of the efficiency index. Let us discuss this contribution a little more. It is well known that any general optimal multipoint method without memory, using  $n + 1$  functional evaluations has optimal convergence order  $2^n$ , so it could reach the optimal efficiency index  $E^*(n + 1, 2^n) = \lim_{n \rightarrow \infty} 2^{\frac{n}{n+1}} = 2$ . On the other hand, we have proved that the new adaptive method with memory (2.25) could reach the same efficiency index with only two functional evaluations (See Theorem 2.2). This means that this method competes with any optimal multipoint method without memory.

In what follows, we disclose the mathematica code for the numerical implementations.

### 4 Dynamical Behavior

Here we focus on the stability behavior of the adaptive methods with memory (2.6) and (2.25). To this end, we utilize visual dynamical approach [13, 15, 17, 19, 24]. Although we have tested many examples, we have reached the same conclusion. Therefore, we only report the results for

the function  $p(z) = z^3 + 1$  and  $p(z) = z^4 - 1$ . To show the stability of the one-parameter (2.6) and two-parameter (2.25) methods, we analyze their dynamic properties and focus on the of dynamic planes related to iterative methods. Mathematica software can be used to do the analysis in which the graphic planes are shown in a rectangle of  $[-3, 3] \times [-3, 3]$  dimension along with a 72-pixel resolution. The dynamic planes illustrate the absorption area for polynomials  $p_1(z) = z^3 + 1$  and  $p_2(z) = z^4 - 1$ .  $p_1(z)$  and  $p_2(z)$  have the solution of  $-1, \frac{1}{2}(1 + i\sqrt{3}), \frac{1}{2}(1 - i\sqrt{3})$  and  $-1, 1, -i, i$ , respectively. Each solution is allocated a color. The greater the dark areas, the greater the intensity of unstability. In Figures 1, 2 and 3, the stability of single parameter methods (2.6) is shown for  $\gamma = 0.1, -0.31093$  and  $-0.3333$ , in which  $p_1(z)$  has been used. Also, in Figures 4, 5 and 6, the stability of single parameter method (2.6) is shown for  $\gamma = 0.1, -0.31336$  and  $-0.33334$  in which  $p_2(z)$  is used. Comparing the figures shows that the stability has been reduced. For the two-parameter method (2.25), in Figure 7, the absorption area and stability are shown for  $\gamma$  and  $\lambda$  at 0.1 and 0.1 respectively, in Figure 8 for  $\gamma = -0.31336$  and  $\lambda = 0.99909$  and in Figure 9 for  $\gamma = -0.33334$  and  $\lambda = 1.00000$ , where  $p_1(z)$  is used. Finally, the two-parameter iterative method (2.25) has been taken to compare absorption areas, using  $p_2(z)$  in Figures 10, 11 and 12, for  $\gamma$  and  $\lambda$  at 0.1 and 0.1 in Figure 10,  $-0.21210$  and  $-1.46997$  in Figure (11) and  $-0.25005, -1.49993$  in Figure 12, where  $p_2(z)$  is used. comparing the figures, we witness there is a reduction in stability showing that, despite the increase of convergence order in the two-parameter method (2.25), the stability trend is satisfactory. Both developed methods with memory (2.6) and (2.25) show the same instable behavior. Consequently, though developing methods with memory has the advantage from the view of computational complexity, they represent numerical chaos and numerical instability.

### 5 Conclusion

In this work, we developed two new but very efficient methods with memory to solve a nonlinear equation. We have shown that the methods

could reach the highest possible efficiency indices and can compete with any method with or without memory in the literature. Convergence analysis of the developed methods has been presented, and we have tested some numerical examples to show the practicality of the proposed methods. Though both developed methods have the highest possible efficiency as opposed to any other methods in the literature, we have seen that methods with memory show instability in practice. Therefore, they benefit from computational efficiency and suffer from numerical stability. Finally, we end the conclusion with the following research question: how can a method with memory be developed to show the numerical stability behavior?

## References

- [1] A. Cordero, J. R. Torregrosa, A class of Steffensen type methods with optimal order of convergence, *Applied Mathematics and Computation* 19 (2011) 7653-7659.
- [2] A. Cordero, T. Lotfi, P. Bakhtiari, J. R. Torregrosa, An efficient two-parametric family with memory for nonlinear equations, *Numer. Algorithms* 68 (2015) 323-335.
- [3] H. T. Kung, J. F. Traub, Optimal order of one-point and multipoint iteration, *J. ACM* 21 (1974) 643-651.
- [4] H. Veisheh, T. Lotfi, T. Allahviranloo, A study on the local convergence and dynamics of the two-step and derivative-free Kung Traubs method (2017). <http://dx.doi.org/10.1007/s40314-017-0458-5/>.
- [5] I. F. Steffensen, Remarks on iteration, *Skand. Aktuarietidskr* 16 (1933) 64-72.
- [6] I. K. Argyros, H. Ren, Efficient Steffensen-type algorithms for solving nonlinear equations, *International Journal of Computer Mathematics* 90 (2013) 691-704.
- [7] I. K. Argyros, M. Kansal, V. Kanwar, S. Bajaj, Higher-order derivative-free families of Chebyshev Halley type methods with or without memory for solving nonlinear equations, *Applied Mathematics and Computation* 315 (2017) 224-245.
- [8] I. K. Argyros, S. George, A. A. Magrenan, Local convergence for multi-point-parametric Chebyshev-Halley-type methods of high convergence order, *Journal of Computational and Applied Mathematics* 282 (2015) 215-224.
- [9] J. Dzunic, On efficient two-parameter methods for solving nonlinear equations, *Numerical Algorithms* 63 (2013) 549-569.
- [10] J. Dzunic, M. S. Petkovic, A cubically convergent steffensen-like method for solving nonlinear equations, *Applied Mathematics Letters* 25 (2012) 1881-1886.
- [11] J. F. Traub, Iterative methods for the solution of equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
- [12] M. S. Petkovic, B. Neta, L. D. Petkovic, J. Dzunic, Multipoint methods for solving nonlinear equations: a survey, *Appl. Math. Comput.* 226 (2014) 635-660.
- [13] M. Salimi, T. Lotfi, S. Sharifi, S. Siegmund, Optimal Newton Secant like methods without memory for solving nonlinear equations with its dynamics, *International Journal of Computer Mathematics* 94 (2017) 1759-1777.
- [14] M. Zaka Ullah, S. Kosari, F. Soleymani, F. Khaksar Haghani, A. S. Al-Fhaid, A super-fast tri-parametric iterative method with memory, *Applied Mathematics and Computation* 289 (2016) 486-491.
- [15] P. Bakhtiari, A. Cordero, T. Lotfi, K. Mahdiani, J. R. Torregrosa, Widening basins of attraction of optimal iterative methods, *Nonlinear Dynamics* 87 (2017) 913-938.
- [16] S. Artidiello, A. Cordero, J. Torregrosa, M. Vassileva, Two weighted eight-order classes of iterative root-finding methods, *International Journal of Computer Mathematics* 92 (2015) 1790-1805.
- [17] S. Amat, S. Busquier, C. Bermudez, A. A. Magrenan, On the election of the damped parameter of a two-step relaxed Newton-type method, *Nonlinear Dynamics* 84 (2016) 9-18.

- [18] S. Amat, S. Busquier, J. M. Gutierrez, On the local convergence of secant-type methods, *International Journal of Computer Mathematics* 81 (2004) 1153-1161.
- [19] S. Aridiello, F. Chicharro, A. Cordero, J. Torregrosa, Local convergence and dynamical analysis of a new family of optimal fourth-order iterative methods, *International Journal of Computer Mathematics* 90 (2013) 2049-2060.
- [20] T. Lotfi, F. Suleiman, M. Ghorbanzadeh, P. Assari, On the construction of some tri-parametric iterative methods with memory, *Numer. Algorithms* 70 (2015) 835-845.
- [21] T. Lotfi, F. Soleymani, Z. Noori, A. Kman, F. Khaksar Haghani, Efficient iterative methods with and without memory possessing high efficiency indices, *Discrete Dynamics in Nature and Society* 9 (2014) 20-34.
- [22] T. Lotfi, K. Mahdiani, P. Bakhtiari, F. Soleymani, Constructing two-step iterative methods with and without memory, *Computational Mathematics and Mathematical Physics* 55 (2015) 183-193.
- [23] T. Lotfi, P. Bakhtiari, A. Cordero, K. Mahdiani, J. R. Torregrosa, Some new efficient multipoint iterative methods for solving nonlinear systems of equations, *International Journal of Computer Mathematics* 92 (2015) 1921-1934.
- [24] X. Wang, T. Zhang, Y. Qin, Efficient two-step derivative-free iterative methods with memory and their dynamics, *International Journal of Computer Mathematics* 93 (2016) 1423-1446.
- [25] Y. H. Geum, Y. I. Kim, B. Neta, On developing a higher-order family of double-Newton methods with a bivariate weighting function, *Appl. Math. Comput.* 254 (2015) 277-290.



Mohammad Javad Lalehchini is a PhD candidate in Department of Applied Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran. Her research areas in Numerical analysis include approximating numerical solutions using iterative methods for nonlinear systems of equations.



Taher Lotfi has got MSc degrees in Applied Mathematics from Kharazmi (Tarbiat Moallem) University, and PhD degree from Science and Research Branch, Islamic Azad University, Tehran, Iran. My main research interest include approximating numerical solutions using iterative methods for nonlinear systems of equations, specially for large systems arising from IE, ODE and PDE. Also, interval analysis, reproducing kernel space methods, soft computing based on wavelets and fuzzy concepts, generalized inverses are my current MSc and PhD students research field.



Katayoun Mahdiani has got MSc and PhD degrees in Applied Mathematics from Islamic Azad University, Karaj, Iran. My main research interest include approximating numerical solutions using iterative methods for nonlinear systems of equations, specially for large systems arising from integral equations.