Journal of Mathematical Extension
Journal Pre-proof
ISSN: 1735-8299
URL: http://www.ijmex.com
Original Research Paper

Convergence and Dynamics of a Family of Secant-Like Methods with Memory to Solve Nonlinear Equations

V. Torkashvand*
Farhangian University
ShQ.C., Islamic Azad University

M. A. Fariborzi Araghi
CT.C., Islamic Azad University

Abstract. In This work, the without-memory methods based on the weight functions are considered, in this work. Also, by entering a self-accelerating parameter and approximating it using the Secant-like techniques and Newton interpolation polynomials, with-memory methods with convergence order $2+\sqrt{5}$, $\frac{1}{2}(5+\sqrt{13})$, $\frac{1}{2}(5+\sqrt{17})$, $3+\sqrt{5}$ and 6 are proposed to solve nonlinear equations. It should be noted that the use of these new techniques to approximate the self-accelerator parameter does not increase computational costs. Several examples are given to illustrate the efficiency and performance of these new methods. We have investigated the basins of attraction of the given weight functions from the proposed method. Hence, the second to fourth-degree polynomial equations for selecting the most appropriate ones have been used.

AMS Subject Classification: MSC 65H05; MSC 65B99.
Keywords and Phrases: With-memory method, Accelerator parameter, Basin of attraction.

Received: October 2024; Accepted: March 2025
*Corresponding Author

1 Introduction

1.1 Literature

We will encounter nonlinear equations in many physical, chemical, financial and mathematical problems, many of which are impossible to solve analytically. A powerful tool is the use of iterative methods. We have used multi-step methods that they were more efficient than the one-step methods. Therefore, we use the without-memory methods which are based on the weight functions to solve them. In the following, we turn these methods into single-parameter with memory methods. The accelerating parameter has been approximated by using the Secant method, quasi-Secant and Newton's interpolatory polynomial. We have constructed the new family of with-memory methods by 50% convergence improvement in the third section.We review the two-step iterative methods as follows.

1.2 Existing iterative methods

In 2007, Kou et al. [18] proposed the following quasi-Halley quadratic method:

$$\begin{cases} w_n = x_n - \frac{f(x_n)}{f'(x_n)},\ n = 0, 1, 2, \cdots, \\ x_{n+1} = w_n - (1 + \frac{2f(w_n)}{f(x_n) - \alpha f(w_n)})\frac{f(w_n)}{f'(x_n)}. \end{cases} \tag{1}$$

In 2008, Chun [9] introduced the first optimal two-step method using the weight function

$$\begin{cases} w_n = x_n - \frac{f(x_n)}{f'(xn)},\ t_n = \frac{f(w_n)}{f(x_n)}, \\ H(0) = 1,\ H'(0) = -2,\ H''(0) < \infty, \\ x_{n+1} = w_n - \frac{f(w_n)}{H(t_n)f'(x_n)}. \end{cases} \tag{2}$$

M.S. Petković and Petković in [29] presented a fourth-order method as follows

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)},\ s_n = \frac{f(y_n)}{f(x_n)}, \\ H(0) = 1,\ H'(0) = 2,\ H''(0) < \infty, \\ x_{n+1} = y_n - H(s_n)\frac{f(y_n)}{f'(x_n)}. \end{cases} \tag{3}$$

Soleymani in [32] proposed the following three-parameters two-point methods

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)},\ t_n = \frac{f(y_n)}{f(x_n)},\ u_n = \frac{f(y_n)}{f'(x_n)},\ v_n = \frac{f(x_n)}{f'(x_n)}, \\ G(0) = 1,\ G'(0) = 0,\ G''(0) < \infty,\ H(0) = 1,\ H'(0) < \infty, \\ K(0) = 1,\ K'(0) = 0,\ K''(0) < \infty, \\ x_{n+1} = y_n - G(t_n)H(u_n)K(v_n)\frac{f(x_n)^2+\beta f(x_n)f(y_n)+\gamma f(y_n)^2}{f(x_n)^2+(\beta-2)f(x_n)f(y_n)+\lambda f(y_n)^2}\frac{f(y_n)}{f'(x_n)}. \end{cases} \tag{4}$$

The fourth-order Soleymani et al.'s method [33] is defined by

$$\begin{cases} w_n = x_n - \frac{2}{3}\frac{f(x_n)}{f'(x_n)},\ t_n = \frac{f(x_n)}{f'(x_n)},\ u_n = \frac{f'(w_n)}{f'(x_n)}, \\ Q(0) = 1,\ Q'(0) = 0,\ Q''(0) = 0,\ P(1) = 1,\ P'(1) = \frac{-1}{4}, \\ P''(1) = \frac{3}{2},\ P'''(1) < \infty, Q'''(0) < \infty, \\ x_{n+1} = x_n - Q(t_n)P(u_n)\frac{2f(x_n)}{f'(x_n)+f'(w_n)}. \end{cases} \tag{5}$$

In 2012, Lee and his colleague Kim [22] proposed a family of two-step methods based on a two-variable function

$$\begin{cases} y = x_n + \beta f(x_n)^k,\ z_n = y_n - \frac{f(y_n)}{f[x_n,y_n]},\ v_n = \frac{f(z_n)}{f(y_n)}, \\ w_n = \frac{f(z_n)}{f(x_n)},\ n = 0,1,2,\cdots,\ k \geq 5, \\ H00 = 1,\ H01 = 2 - H10,\ x_{n+1} = z_n - H(v_n,w_n)\frac{f(z_n)}{f[x_n,y_n]}. \end{cases} \tag{6}$$

Six years after Chun, Lotfi [23] presented the following two-step without-memory method for solving nonlinear equations.

$$\begin{cases} w_n = x_n - \frac{2}{3}\frac{f(x_n)}{f'(x_n)},\ s_n = \frac{f'(y_n)}{f'(x_n)}, \\ H(0) = 2,\ H'(0) = \frac{-7}{4},\ H''(0) = \frac{3}{2},\ H'''(0) < \infty, \\ x_{n+1} = x_n - H(s_n)\frac{2f(x_n)}{f'(x_n)+f'(y_n)}. \end{cases} \tag{7}$$

In 2016, the two-point iterative method that constructed by Kansal et al. [17]

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\dfrac{125f(x_n)^3 + 25f(x_n)^2f(y_n) - 150f(x_n)f(y_n)^2 - 96f(y_n)^3}{125f(x_n)^3 - 100f(x_n)^2f(y_n) - 300f(x_n)f(y_n)^2 - 156f(y_n)^3}. \end{cases} \tag{8}$$

In 2019, Junjua et al. [16] proposed the following two-point method based on inverse interpolation

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ g_1 = \frac{1}{(f(y_n)-f(x_n))f[x_n,y_n]} - \frac{1}{(f(y_n)-f(x_n))f'(x_n)}, \\ x_{n+1} = R(0) = y_n + g_1 f(x_n)^2. \end{cases} \tag{9}$$

Cordero et al. [12] , in 2021, solved the nonlinear systems by usining the following parametric family of iterative schemes

$$\begin{aligned} & y^{(n)} = z^{(n)} - \alpha[f'(z^{(n)})]^{-1} f(z^{(n)}), \\ & z^{(n+1)} = z^n - (\beta[f'(y^{(n)})]^{-1} f'(z^{(n)})[f'(y^{(n)})]^{-1} + \gamma[f'(y^{(n)})]^{-1} \\ & + \mu[f'(z^{(n)})]^{-1} + \delta[f'(z^{(n)})]^{-1} f'(y^{(n)})[f'(z^{(n)})]^{-1}) f(z^{(n)}). \end{aligned} \tag{10}$$

Torkashvand et al. [39], in 2023, found the following iterative methods based on with-memory method

$$\begin{cases} w_0 = x_0 - \gamma_0 f(x_0), \gamma_k = \frac{1}{N_3'(x_k)}, \ k = 1, 2, 3, \cdots, \\ g(0) = g'(0) = 1, \ |g''(0)| < \infty, \\ y_k = x_k - \frac{f(x_k)}{f[x_k,w_k]+(w_k-x_k)f(w_k)}, \ k = 0, 1, 2, \cdots, \\ t_k = \frac{f(y_k)}{f(x_k)}, \ x_{k+1} = y_k - g(t_k)\frac{f(y_k)}{f[y_k,w_k]+(y_k-x_k)f(y_k)}. \end{cases} \tag{11}$$

1.3 Motivation and organization

Among many indices for comparison of different methods such as the index of efficiency, radius of convergence, improvement convergence order, etc.; here, we try to build a family of with-memory methods that fits better to improving the degree of convergence. For this reason, in this paper, we firstly propose an efficient fourth-order without-memory method. Then, we have converted the new two-step methods to the with-memory methods. It has shown that the proposed with-memory schemes mostly efficiency index better than the optimal method second-, fourth-, eighth-, sixteenth-order.

The rest of this paper has been prepared as follows. In Section 2, the construction of the new without-memory schemes has been offered. Section 3 includes its analysis of convergence of some with-memory methods and it shows the with-memory proposed methods have fifth- sixth-order. Numerical examples and comparisons of the proposed methods with others with- and without-memory methods have been given in Section 4. In Section 5, the basins of attraction of the given weight functions from the proposed method. We have used the second and third-degree polynomial equations for selecting the most appropriate ones. Finally, we have illustrated conclusions.

2 The Methods and Analysis of Convergence

Firstly, suppose the double Newton's method as follows

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)}. \end{cases} \tag{12}$$

The error equation of the method (12) is as follows

$$e_{n+1} = c_2^3 e_n^4 + O(e_n^5). \tag{13}$$

As can be seen, this method has four evaluations of the functions $f(x_n)$, $f'(x_n)$, $f(y_n)$ and $f'(y_n)$. Also, its convergence order is equal to four. Therefore, this method is not optimal in terms of Kung-Traub conjecture for multi-step methods without memory. Hence, one of the function evaluations should be omitted. Besides, to eliminate the calculation of the derivative, we will approximate $f'(x_n)$ and $f'(y_n)$ in terms of previous information and the use of the weight function as follows

$$\begin{cases} f'(y_n) \approx \frac{f[y_n,x_n]}{H(t_n)},\ t_n = \frac{f(y_n)}{f(x_n)}, \\ w_n = x_n + \beta f(x_n,\ f'(x_n) \approx f[w_n, x_n] = \frac{f(w_n)-f(x_n)}{w_n - x_n}. \end{cases} \tag{14}$$

$\beta \in \mathbb{R}$. In the following, we will specify the conditions of the weight function. Therefore, we start from the scheme (12), the approximations (14) and state the following two-point method

$$\begin{cases} w_n = x_n + \beta f(x_n),\ y_n = x_n - \frac{f(x_n)}{f[w_n,x_n]}, \\ t_n = \frac{f(y_n)}{f(w_n)},\ x_{n+1} = y_n - H(t_n)\frac{f(y_n)}{f[y_n,w_n]}. \end{cases} \tag{15}$$

The following theorem illustrates that under what conditions on weight function, the convergence order of two-step family (15) will arrive at the optimal level 4.

Theorem 2.1. *Let $H,\ f : D \subset \mathbb{R} \to \mathbb{R}$ have a single root $\alpha \in D$, for an open interval D. And the f(x) is sufficiently differentiable. If the initial point x_0 is sufficiently close to α, then the sequence x_m generated by any method of the family (15) converges to α. If H is any function with $H(0) = 1,\ H'(0) = 1,\ |H''(0)| < \infty$ and $\beta \neq 0$ then the methods defined by (15) have convergence of order at least 4.*

Proof. By using Taylor's expansion of $f(x)$ about α and taking into calculation that $f(\alpha) = 0$, we earn

$$f(x_n) = f'(\alpha)(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)). \tag{16}$$

Then, computing $e_{n,w} = w_n - \alpha$, we attain $w_n = x_n + \beta f(x_n)$

$$e_{n,w} = e_n + e_n\beta f'(\alpha)(1 + e_n(c_2 + e_n(c_3 + e_nc_4))) + O(e_n^5), \tag{17}$$

and

$$\begin{aligned} f(w_n) =& f'(\alpha)(e_n + e_n\beta f'(\alpha)(1 + e_n(c_2 + e_n(c_3 + e_nc_4))) \\ &+ c_2(e_n + e_n\beta f'(\alpha)(1 + e_n(c_2 + e_n(c_3 + e_nc_4))))^2 \\ &+ c_3(e_n + e_n\beta f'(\alpha)(1 + e_n(c_2 + e_n(c_3 + e_nc_4))))^3 \\ &+ c_4(e_n + e_n\beta f'(\alpha)(1 + e_n(c_2 + e_n(c_3 + e_nc_4))))^4). \end{aligned} \tag{18}$$

Considering $f[x,y] = \frac{f(x)-f(y)}{x-y}$ is Newton's first order divided difference. We get

$$\begin{aligned} f[x_n, w_n] =& -1/(e_n\beta f'(\alpha)(1+e_n(c_2+e_n(c_3+e_nc_4))))^{-1} \\ & (e_nf'(\alpha)(1+e_n(c_2+e_n(c_3+e_nc_4))) - f'(\alpha) \\ & (e_n + e_n\beta f'(\alpha)(1+e_n(c_2+e_n(c_3+e_nc_4))) \\ & + c_2(e_n+e_nf'(\alpha)(1+e_n(c_2+e_n(c_3+e_nc_4))))^2\beta \\ & + c_3(e_n+e_n\beta f'(\alpha)(1+e_n(c_2+e_n(c_3+e_nc_4))))^3 \\ & + c_4(e_n+e_n\beta f'(\alpha)(1+e_n(c_2+e_n(c_3+e_nc_4))))^4)). \end{aligned} \tag{19}$$

From (16) and (19), we now have

$$\begin{aligned} y_n =& \alpha + (1+\beta f'(\alpha))e_n^2 + (-(2+\beta f'(\alpha)(2+\beta f'(\alpha))c_2^2) \\ & + (1+\beta f'(\alpha))(2+\beta f'(\alpha))c_3e_n^3 + ((4+\beta f'(\alpha)(5+ \\ & \beta f'(\alpha)(3+\beta f'(\alpha))))c_2^3 - (7+\beta f'(\alpha)(10+\beta f'(\alpha) \\ & (7+2\beta f'(\alpha))))c_2c_3 + (1+\beta f'(\alpha)(3+\beta f'(\alpha)(3+ \\ & \beta f'(\alpha)))c_4)e_n^4 + O(e_n^5). \end{aligned} \tag{20}$$

The expansion of $f(y_n)$ about α is given as

$$\begin{aligned} f(y_n) =& f'(\alpha)(1+\beta f'(\alpha))c_2e_n^2 + f'(\alpha)(-(2+\beta f'(\alpha)(2+ \\ & \beta f'(\alpha))c_2^2 + (1+\beta f'(\alpha))(2+\beta f'(\alpha))c_3e_n^3 + f'(\alpha) \\ & ((5+\beta f'(\alpha)(7+\beta f'(\alpha)(4+\beta f'(\alpha))))c_2^3 - (7+ \\ & \beta f'(\alpha)(10+\beta f'(\alpha)(7+2\beta f'(\alpha))))c_2c_3 + (1+ \\ & \beta f'(\alpha)(3+\beta f'(\alpha)(3+\beta f'(\alpha)))c_4)e_n^4 + O(e_n^5). \end{aligned} \tag{21}$$

Using (18) and (21), we obtain

$$\begin{aligned} \frac{f(y_n)}{f[y_n,x_n]} =& (1+\beta f'(\alpha))c_2e_n^2 + (-(3+\beta f'(\alpha)(3+\beta f'(\alpha)))c_2^2 + \\ & (1+\beta f'(\alpha))(2+\beta f'(\alpha))c_3e_n^3 + ((7+\beta f'(\alpha)(8+\beta \\ & f'(\alpha)(4+\beta f'(\alpha))))c_2^3 - 2(5+\beta f'(\alpha)(7+\beta f'(\alpha) \\ & (4+\beta f'(\alpha))))c_2c_3 + (1+\beta f'(\alpha))(3+\beta f'(\alpha) \\ & (3+\beta f'(\alpha)))c_4)e_n^4 + O(e_n^5). \end{aligned} \tag{22}$$

Using the Taylor expansion $H(t_k)$ we have

$$H(t_n) = H(\frac{f(y_n)}{f(w_n)}) = H(0) + H'(0)(\frac{f(y_n)}{f(w_n)}) + H''(0)\frac{(\frac{f(y_n)}{f(w_n)})^2}{2}. \quad (23)$$

Now by using relations $h0 = H(0)$, $h1 = H'(0)$, $h2 = H''(0)$ and from (23), we get

$$\begin{aligned} H(t_n) =& h0 + h1c_2e_n + (\frac{1}{2}(h2 - 2h1(3 + 2\beta f'(\alpha))c_2^2 + h1(2+ \\ &\beta f'(\alpha))c_3e_n^2 + ((-h2(3 + 2\beta f'(\alpha)) + h1(8 + \beta f'(\alpha) \\ &(8 + 3\beta f'(\alpha))))c_2^3 + (h2(2 + \beta f'(\alpha)) - h1(10 + \beta f'(\alpha) \\ &(11 + 4\beta f'(\alpha))))c_2c_3 + h1(3 + \beta f'(\alpha)(3 + \beta f'(\alpha))) \\ &c_4)e_n^3 + (\frac{1}{2}(h2(25 + 2\beta f'(\alpha)(14 + 5\beta f'(\alpha))) - 2h1(20+ \\ &\beta f'(\alpha)(26 + \beta f'(\alpha)(15 + 4\beta f'(\alpha)(3 + \beta f'(\alpha)))))c_2^4+ \\ &(-2h2(8 + 3\beta f'(\alpha)(3 + \beta f'(\alpha)) + h1(37 + \beta f'(\alpha) \\ &(52 + 3\beta f'(\alpha)(11 + 3\beta f'(\alpha)))))c_2^2c_3 + \frac{1}{2}(h2(2 + \beta \\ &f'(\alpha))^2 - 2h1(8 + \beta f'(\alpha)(13 + \beta f'(\alpha)(9 + 2\beta f'(\alpha)))) \\ &c_3^2 + (h2(3 + \beta f'(\alpha)(3 + \beta f'(\alpha))) - h1(14 + \beta f'(\alpha) \\ &(21 + 2\beta f'(\alpha)(7 + 2\beta f'(\alpha)))))c_2c_4 + h1(2 + \beta f'(\alpha)) \\ &(2 + \beta f'(\alpha)(2 + \beta f'(\alpha)))c_5)e_n^4 + O(e_n^5). \end{aligned} \quad (24)$$

Finally, replacing (20), (21), (22) and (24) in the last step of (15),we get

$$x_{n+1}-\alpha=y_n-\alpha-H(t_n)\frac{f(y_n)}{f[y_n,w_n]}=-(-1+h0)(1+\beta f'(\alpha))$$
$$c_2e_n^2+(-(-2+h1(1+\beta f'(\alpha))+\beta f'(\alpha)(2+\beta f'(\alpha))-$$
$$h0(3+\beta f'(\alpha)(3+\beta f'(\alpha)))^2-\beta f'(\alpha)(2+\beta f'(\alpha)))c_2^2-$$
$$(-1+h0)(1+\beta f'(\alpha))(2+\beta f'(\alpha))c_3)e_n^3+(\frac{-1}{2})-8+$$
$$h2-10\beta f'(\alpha)+\beta f'(\alpha)(h2-2\beta f'(\alpha)(3+\beta f'(\alpha)))-2$$
$$h1(6+\beta f'(\alpha)(8+3\beta f'(\alpha)))+2h0(7+\beta f'(\alpha)(8+\beta f'(\alpha)$$
$$(4+\beta f'(\alpha)))))c_2^3+(-7-2h1(1+\beta f'(\alpha))(2+\beta f'(\alpha))$$
$$-\beta f'(\alpha)(10+\beta f'(\alpha)(7+2\beta f'(\alpha)))+2h0(5+\beta f'(\alpha)$$
$$(7+\beta f'(\alpha)(4+\beta f'(\alpha)))))c_2c_3-(-1+h0)(1+\beta f'(\alpha))$$
$$(3+\beta f'(\alpha)(3+\beta f'(\alpha)))c_4)e_n^4+O(e_n^5). \tag{25}$$

By putting $h0=1$ and $h1=1$ the final error expression is given by

$$e_{n+1}=\frac{-1}{2}((1+\beta f'(\alpha))c_2)((-6+h2-4\beta f'(\alpha))c_2^2+2(1+\beta f'(\alpha))$$
$$c_3))e_n^4+O(e_n^5). \tag{26}$$

If we set $h0=1, h1=1$ and $h2=2$ the error equation is

$$e_{n+1}=(1+\beta f'(\alpha))^2c_2(2c_2^2-c_3)e_n^4+O(e_n^5). \tag{27}$$

Hence, the fourth-order convergence is established. $\square$

Some concrete weight functions that satisfy the conditions $H(0)=1$, and $H'(0)=1$ are

$$\begin{cases} H_1(t_n)=1+t_n,\ H_2(t_n)=\frac{1}{1-t_n},\ H_3(t_n)=e^{t_n}, \\ H_4(t_n)=\cos(t_n)+\sin(t_n),\ H_5(t_n)=1+\tan(t_n), \\ H_6(t_n)=1+\arcsin(t_n). \end{cases} \tag{28}$$

Some of the weight functions that apply to the given conditions $H(0)=1$, $H'(0)=1$ and $H''(0)=2$ are as follows

$$H_7(t_n)=1+t_n+t_n^2,\ H_8(t_n)=\frac{1}{2}(\arctan(t_n)+\arcsin(t_n))+t_n^2+1. \tag{29}$$

Remark 2.2. *From Theorem 2.1 can be resulted that the new fourth-order convergent iterative method (15) satisfies the conjecture of Kung and Traub that a multipoint without memory method with three evaluations of functions and can achieve an optimal fourth-order of convergence and an efficiency index of* $\sqrt[3]{4} \approx 1.58$.

In the next section, we extend these schemes into with memory methods for solving the nonlinear equations.

3 Acceleration of the Family of Two-point Methods

In this section, we propose the following iterative method with memory based on (15)

$$\begin{cases} \beta_n = \frac{-1}{f'(\alpha)},\ n = 1, 2, 3, \cdots, \\ w_n = x_n + \beta_n f(x_n),\ y_n = x_n - \frac{f(x_n)}{f[w_n, x_n]},\ n = 0, 1, 2, \cdots, \\ t_n = \frac{f(y_n)}{f(w_n)},\ H(0) = H'(0) = 1,\ |H''(0)| < \infty, \\ x_{n+1} = y_n - H(t_n)\frac{f(y_n)}{f[y_n, w_n]}. \end{cases} \tag{30}$$

We observe from (26) that the order of convergence of the presented methods (15) is 4 when $\beta \neq \frac{-1}{f'(\alpha)}$. The exact root of the equation is not available, so the value of $f'(\alpha)$ cannot be calculated accurately. So it can be approximated as follows: $f'(\alpha) \approx \bar{f}'(\alpha)$. We could approximate the parameter β by β_n. Therefore, one of the following methods can be used to approximate the self-accelerator parameter

Method 1.

$$\beta_n \approx \frac{-1}{\bar{f}'(\alpha)} = -\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}. \tag{31}$$

Method 2.

$$\beta_n \approx \frac{-1}{\bar{f}'(\alpha)} = -\frac{w_n - x_{n-1}}{f(w_n) - f(x_{n-1})}. \tag{32}$$

Method 3.

$$\beta_n \approx \frac{-1}{\bar{f}'(\alpha)} = -\frac{y_n - x_{n-1}}{f(y_n) - f(x_{n-1})}. \tag{33}$$

Method 4.

$$\beta_n \approx \frac{-1}{\bar{f}'(\alpha)} = -\frac{1}{N_3'(x_n)}, \tag{34}$$

where $N_3(x_n)$ are defined as follows $N_3(x_n) = N_3(t; x_n, x_{n-1}, w_{n-1}, y_{n-1})$,

Theorem 3.1. *Let α is a simple root of $f(x) = 0$ and $f(x)$ is sufficiently differentiable. If an initial approximation x_0 is sufficiently close to the zero α of $f(x) = 0$ and the parameter β_n in the iterative method (30) is recursively calculated by the forms given in (31)-(34). Then, the R-order of convergence of the two-point with-memory methods (30) with the corresponding expressions (31), (32), (33) and (34) of β_n is at least $2+\sqrt{5}$, $\frac{1}{2}(5+\sqrt{13})$, $\frac{1}{2}(5+\sqrt{17})$ and $3+\sqrt{5}$ respectively.*

Proof. First, we assume that the R-orders of convergence of the sequences w_n, y_n and x_n are at least P, Q and R, respectively. Hence

$$\begin{cases} e_{n+1} \sim e_n^R \sim e_n^{R^2}, \\ e_{n,y} \sim e_n^Q \sim e_n^{RQ}, \\ e_{n,w} \sim e_n^P \sim e_n^{RP}, \end{cases} \tag{35}$$

The Taylor's series expansion of $f(x)$ about α is given as

$$f(x) = f(\alpha) + (x-\alpha) f'(\alpha) + \frac{(x-\alpha)^2 f''(\alpha)}{2!} + \frac{(x-\alpha)^3 f''(\alpha)}{3!} + \dots. \tag{36}$$

Method 1. Now using the relations $x_n - \alpha = e_n$ and $x_{n-1} - \alpha = e_{n-1}$ also the relation (36) we have

$$f(x_n) = f(\alpha) + (x_n - \alpha) f'(\alpha) + \frac{(x_n-\alpha)^2 f''(\alpha)}{2!} + \frac{(x_n-\alpha)^3 f'''(\alpha)}{3!} + \dots. \tag{37}$$

and

$$f(x_{n-1}) = f(\alpha) + (x_{n-1} - \alpha) f'(\alpha) + \frac{(x_{n-1}-\alpha)^2 f''(\alpha)}{2!} + \frac{(x_{n-1}-\alpha)^3 f'''(\alpha)}{3!} + \dots. \tag{38}$$

Using (37), (38) and $x_n - x_{n-1} = e_n - e_{n-1}$ we get

$$\begin{aligned}
&\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \qquad (39)\\
&= \frac{(e_n - e_{n-1})f'(\alpha) + \frac{(e_n-e_{n-1})^2 f''(\alpha)}{2!} + \frac{(e_n-e_{n-1})^3 f'''(\alpha)}{3!} + \dots}{e_n - e_{n-1}}\\
&= f'(x^*) + \frac{(e_n + e_{n-1})f''(\alpha)}{2!} + \frac{(e_n^2 - e_n e_{n-1} + e_{n-1}^2)f'''(\alpha)}{3!} + \dots
\end{aligned}$$

Using the relation (31) we have

$$\beta_n = -\frac{1}{f'(\alpha) + \frac{(e_n+e_{n-1})f''(\alpha)}{2!} + \frac{(e_n^2-e_n e_{n-1}+e_{n-1}^2)f'''(\alpha)}{3!} + \dots} \qquad (40)$$

Now to calculate $(1 + \beta_n f'(\alpha))$ using equation (40) we obtain

$$\begin{aligned}
&1 + \beta_n f'(\alpha) \qquad (41)\\
&= 1 - \frac{f'(\alpha)}{f'(\alpha) + \frac{(e_n+e_{n-1})f''(\alpha)}{2!} + \frac{(e_n^2-e_n e_{n-1}+e_{n-1}^2)f'''(\alpha)}{3!} + \dots}\\
&= \frac{(e_n + e_{n-1})c_2 + (e_n^2 - e_n e_{n-1} + e_{n-1}^2)c_3 + \dots}{1 + (e_n + e_{n-1})c_2 + (e_n^2 - e_n e_{n-1} + e_{k-1}^2)c_3 + \dots}\\
&\sim c_2 e_{n-1}
\end{aligned}$$

Following (17), (20), (27) and (41), we find

$$\begin{cases}
e_{n+1} \sim (1 + \beta_n f'(\alpha))e_n^4 \sim e_{n-1}e_n^4 \sim e_{n-1}^{1+4R},\\
e_{n,y} \sim (1 + \beta_n f'(\alpha))e_n^2 \sim e_{n-1}e_n^2 \sim e_{n-1}^{1+2R},\\
e_{n,w} \sim (1 + \beta_n f'(\alpha))e_n \sim e_{n-1}e_n \sim e_{n-1}^{1+R}.
\end{cases} \qquad (42)$$

Now, comparing the error exponents of e_{n-1} on the right hand sides of pairs (35) and (42), respectively, we obtain the following system

$$\begin{cases}
RP - 1 - R = 0,\\
RQ - 1 - 2R = 0,\\
R^2 - 1 - 4R = 0.
\end{cases}$$

This system has the solution $P = -1+\sqrt{5}$, $Q = \sqrt{5}$ and $R = 2+\sqrt{5}$. Thus, we can conclude that the lower bound of the convegence order of the with-memory methods (30) and (31) is $R = 2+\sqrt{5}$. We show this method with TM4.2.

Method 2. Also, by using the relations $w_{n-1}-\alpha = e_{n-1,w}$, $x_n - w_{n-1} = e_n - e_{n-1,w}$ and the relation (36) we have

$$f(w_{n-1}) = \tag{43}$$
$$f(\alpha)+(w_{n-1}-\alpha)f'(\alpha)+\frac{(w_{n-1}-\alpha)^2 f''(\alpha)}{2!}+\frac{(w_{n-1}-\alpha)^3 f'''(\alpha)}{3!}+\dots.$$

and

$$\frac{f(x_n)-f(w_{k-1})}{x_k-w_{k-1}} \tag{44}$$
$$= \frac{(e_n-e_{n-1,w})f'(\alpha)+\frac{(e_n-e_{n-1,w})^2 f''(\alpha)}{2!}+\frac{(e_n-e_{n-1,w})^3 f'''(\alpha)}{3!}+\dots}{e_n-e_{n-1,w}}$$
$$= f'(\alpha)+\frac{(e_n+e_{n-1,w})f''(\alpha)}{2!}+\frac{(e_n^2-e_ne_{n-1,w}+e_{n-1,w}^2)f'''(\alpha)}{3!}+\dots$$

Now to calculate $(1+\beta_n f'(\alpha))$ using eq. (44) we get

$$1+\beta_n f'(\alpha) \tag{45}$$
$$= 1-\frac{f'(\alpha)}{f'(\alpha)+\frac{(e_n+e_{n-1,w})f''(\alpha)}{2!}+\frac{(e_n^2-e_ne_{n-1,w}+e_{n-1,w}^2)f'''(x^*)}{3!}+\dots}$$
$$= \frac{(e_n+e_{n-1,w})c_2+(e_n^2-e_ne_{n-1,w}+e_{n-1,w}^2)c_3+\dots}{1+(e_n+e_{n-1,w})c_2+(e_n^2-e_ne_{n-1,w}+e_{n-1,w}^2)c_3+\dots}$$
$$\sim c_2e_{n-1,w}$$

Following (17), (20), (27) and (45), we find

$$\begin{cases} e_{n+1} \sim (1+\beta_n f'(\alpha))e_n^4 \sim e_{n-1,w}e_n^4 \sim e_{n-1}^{P+4R}, \\ e_{n,y} \sim (1+\beta_n f'(\alpha))e_n^2 \sim e_{n-1,w}e_k^2 \sim e_{k-1}^{P+2R}, \\ e_{n,w} \sim (1+\beta_n f'(\alpha))e_n \sim e_{n-1,w}e_n \sim e_{n-1}^{P+R}, \end{cases} \tag{46}$$

Now, comparing the error exponents of e_{n-1} on the right hand sides of pairs (35) and (46), respectively, we achieve as follows system:

$$\begin{cases} RP - P - R = 0, \\ RQ - P - 2R = 0, \\ R^2 - P - 4R = 0. \end{cases}$$

This system has the solution $P = \frac{1}{2}(-1 + \sqrt{13})$, $Q = \frac{1}{2}(1 + \sqrt{13})$ and $R = \frac{1}{2}(5 + \sqrt{13})$. Thus, we can conclude that the lower bound of the convegence order of the with-memory methods (30) and (32) is $r = \frac{1}{2}(5 + \sqrt{13})$. We show this method with TM4.3.

Method 3. Using the relations $x_n - y_{n-1} = e_n - e_{n-1,y}$, $y_{n-1} - \alpha = e_{n-1,y}$ and (35) we have

$$f(y_{n-1}) = f(\alpha) + (y_{n-1} - \alpha) f'(\alpha) + \frac{(y_{n-1} - \alpha)^2 f''(\alpha)}{2!} + \frac{(y_{n-1} - \alpha)^3 f'''(\alpha)}{3!} + \ldots. \tag{47}$$

Similarly, we have

$$1 + \beta_n f'(\alpha) \sim c_2 e_{n-1,y} \tag{48}$$

Following (17), (20), (27) and (48), we find

$$\begin{cases} e_{n+1} \sim (1 + \beta_n f'(\alpha)) e_n^4 \sim e_{n-1,y} e_n^4 \sim e_{n-1}^{Q+4R}, \\ e_{n,y} \sim (1 + \beta_n f'(\alpha)) e_n^2 \sim e_{n-1,y} e_n^2 \sim e_{n-1}^{Q+2R}, \\ e_{n,w} \sim (1 + \beta_n f'(\alpha)) e_n \sim e_{n-1,y} e_k \sim e_{n-1}^{Q+R}, \end{cases} \tag{49}$$

Now, comparing the error exponents of e_{k-1} on the right hand sides of pairs (35) and (49), respectively, we obtain the following system:

$$\begin{cases} RP - (Q + R) = 0, \\ RQ - (Q + 2R) = 0, \\ R^2 - (Q + 4R) = 0. \end{cases}$$

This system has the solution $P = \frac{1}{4}(9 - \sqrt{17})$, $Q = \frac{1}{2}(1 + \sqrt{17})$ and $R = \frac{1}{2}(5 + \sqrt{17})$. Thus, we can conclude that the lower bound of the convegence order of the with-memory methods (30) and (33) is $R = \frac{1}{2}(5+\sqrt{17})$. We show this method with TM4.5.

Method 4. To prove the last-part of Theorem 3.1, we use the following code written in Mathematica software:

```
ClearAll["Global'*"]
A[t_]:=InterpolatingPolynomial[{{e,fx},{ew,fw},{ey,fy}, {e1,fx1}},t]
Approximation=-1/A'[e1]//Simplify;
fx=fla*(e+c2*e^2+c3*e^3+c4*e^4);
fw=fla*(ew+c2*ew^2+c3*ew^3+c4*ew^4);
fy=fla*(ey+c2*ey^2+c3*ey^3+c4*ey^4);
fx1=fla*(e1+c2*e1^2+c3*e1^3+c4*e1^4)
\beta =Series[Approximation,{ e,0,2}, { ew,0,2}, { ey,0,2}, { e1,0,0}]//Simplify;
Collect[Series[1+\beta*fla,{e,0,1},{ew,0,1},{ey,0,1},{e1,0,0}],{e,ew,ey,e1},Simplify]
```

which results in

$$c_4 eewey \tag{50}$$

Therefore, one may obtain

$$1+\beta_n f'(\alpha) \sim c_4 e_{n-1} e_{n-1,w} e_{n-1,y} \tag{51}$$

Using Eq. (51) and the error equation of the two-step with-memory method in Equation (30), we have:

$$\begin{cases} e_{n+1} \sim (1+\beta_n f'(\alpha))e_n^4 \sim e_{n-1}e_{n-1,w}e_{n-1,y}e_n^4 \sim e_{n-1}^{1+P+Q+4R}, \\ e_{n,y} \sim (1+\beta_n f'(\alpha))e_n^2 \sim e_{n-1}e_{n-1,w}e_{n-1,y}e_n^2 \sim e_{n-1}^{1+P+Q+2R}, \\ e_{n,w} \sim (1+\beta_n f'(\alpha))e_n \sim e_{n-1}e_{n-1,w}e_{n-1,y}e_n \sim e_{n-1}^{1+P+Q+R}. \end{cases} \tag{52}$$

By comparing the error exponents of e_{k-1} on the right hand sides of pairs (42) and (52), respectively, we achieve as follows system:

$$\begin{cases} RP-(1+P+Q+R)=0, \\ RQ-(1+P+Q+2R)=0, \\ R^2-(1+P+Q+4R)=0. \end{cases}$$

This system has the solution $P=\sqrt{5}$, $Q=(1+\sqrt{5})$ and $R=(3+\sqrt{5})$. Thus, we can conclude that the lower bound of the convegence order of the with-memory methods (30) and (34) is $R = 3 + \sqrt{5}$. We show this method with TM5.2. This completes the proof. $\square$

Remark 3.2. *Convergence order of the two-point with-memory methods (30) with the corresponding expressions (31), (32), (33) and (34) of* β_k *is at least* $2+\sqrt{5}$, $\frac{1}{2}(5+\sqrt{13})$, $\frac{1}{2}(5+\sqrt{17})$, *and* $3+\sqrt{5}$. *Therefore, theirs efficiency index is* $(2+\sqrt{5})^{\frac{1}{3}} = 1.61$, $(\frac{1}{2}(5+\sqrt{13}))^{\frac{1}{3}} = 1.62$, $(\frac{1}{2}(5+\sqrt{17}))^{\frac{1}{3}} = 1.65$ *and* $(3+\sqrt{5})^{\frac{1}{3}} = 1.73$.

In the following, we convert the without-memory method (15) to a with-memory method with weight function conditions $H(0) = H'(0) = 1$, $H''(0) = 2$ as follows.

$$\begin{cases} \beta_k = \frac{-1}{N_3'(x_k)}, \ k = 1, 2, 3, \cdots, \\ w_k = x_k + \beta_k f(x_k), \ y_k = x_k - \frac{f(x_k)}{f[w_k, x_k]}, \ k = 0, 1, 2, \cdots, \\ t_k = \frac{f(y_k)}{f(w_k)}, \ H(0) = H'(0) = 1, \ H''(0) = 2, \ x_{k+1} = y_k - H(t_k)\frac{f(y_k)}{f[y_k, w_k]}. \end{cases} \tag{53}$$

Then, by approximating the self-accelerator parameter β_k by Newton's interpolation polynomial, we propose the method whose convergence order is equal to 6. The next Theorem shows a proof of the order of convergence of method (53).

Theorem 3.3. *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, *then, the R-order of convergence of the two-step method (53) with memory is at least 6.*

Proof. We apply Herzberger's matrix method [14] to define the convergence order. Observe that the order lower bound for a single-step s-point method (53) $x_n = \varphi(x_{n-1}, x_{n-2}, \cdots, x_{n-s})$ is the spectral radius of a matrix $N^{(s)} = (n_{ij})$, related to the method with elements:

$$\begin{cases} n_{1,j} = amount\ of\ information\ required\ at\ point\ x_{n-j}, \ j = 1, 2, 3, \cdots, s, \\ n_{i,i-1} = 1, \ i = 2, 3, \cdots, s, \\ n_{i,j} = 0 \ otherwise \end{cases} \tag{54}$$

So, the order lower bound of an s-step method $\psi = \psi_1 \circ \psi_2 \circ \cdots \circ \varphi_s$ is the spectral radius of the product of matrices $N = N_1.N_2.\cdots.N_s$. We can state each approximation $x_{n+1}, y_n,$ and w_n as a function of accessible information $f(y_n), f(w_n)$ and $f(x_n)$ from the n-th iteration and $f(y_{n-1}), f(w_{n-1})$ and $f(x_{n-1})$ from the previous iteration, depending on the accelerating technique. Now, we construct the corresponding matrices as follows

$$x_{n+1} = \psi_1(y_n, w_n, x_n, y_{n-1}); \Rightarrow N_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$y_n = \psi_2(w_n, x_n, y_{n-1}, w_{n-1}); \Rightarrow N_2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$w_n = \psi_3(x_n, y_{n-1}, w_{n-1}, x_{n-1}); \Rightarrow N_3 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Hence, we obtain

$$N = N_1 N_2 N_3 = \begin{pmatrix} 4 & 4 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix},$$

an its eigenvalues are $(6, 0, 0, 0)$. Since the spectral radius of the matrix N is 6, we conclude that the R-order of the methods with memory (53) is at least 6. □

At the end of Section 3, we note that recently several multi-point with-memory methods for solving nonlinear equations have been studied by Lalehchini et al. [21] and Zafar et al. [43]. Also some new families have been proposed of with-memory methods by Torkashvand et al. [36, 37, 38, 40, 42]. They approximated the self-accelerator parameter by using Newton's interpolation polynomials. Also, Argyros [2, 19] and Moccari et al. [26] have studied the local and semilocal convergence iterative methods. In the sequal, we study the efficiency of the proposed methods by five numerical examples.

4 Numerical Results

In this section, we check the effectiveness of the new optimal fourth-order family of methods (15), taking $\beta_0 = 0.1$, which is denoted by TM4; compared with the with-memory methods (TM4.2), (TM4.3), (TM4.5), (TM5.2), (TM6) and Compos et al.'s method (CCTVM)[6], Chicharro et al.'s method (CCGTM)[7], Choubey et al.'s method (CPGM)[8], Petković et al.'s method (PDPM)[30], Mohamadi et al.'s method (MLAM)[27] and Traub's method (TM)[41]. The five nonlinear functions, and the exact root also the initial approximation of the roots can be seen below to check the degree of convergence as well as the efficiency index of the proposed methods with other methods:

$$\begin{cases} f_1(x) = x^5 + x^4 + 4x^2 - 15,\ \alpha \approx 1.34,\ x_0 = 1.1, \\ f_2(x) = x^3 + 4x^2 - 10,\ \alpha \approx 1.36,\ x_0 = 1, \\ f_3(x) = 10xe^{-x^2} - 1,\ \alpha \approx 1.67,\ x_0 = 1, \\ f_4(x) = x\log(1 + x\sin(x)) + e^{-1+x^2+x\cos(x)}\sin(\pi x),\ \alpha = 0,\ x_0 = 0.6, \\ f_5(x) = \sqrt{x^4+8}\sin(\frac{\pi}{x^2+2} + \frac{x^3}{x^4+1} - \sqrt{6} + \frac{8}{17},\ \alpha = -2,\ x_0 = -2.3. \end{cases}$$

The convergence criterion selected is a toleration of 10^{-6} with a maximum of 150 iterations. All computations have been done on Intel $Corei5 - 4210UCPU@1.70GHz2.40GHZ$ with 4 GB RAM, using Microsoft Windows 10, 32 bit based on X32-based processor. Mathematica 10 has been used to generate all graphs and computations. The symbols used in these tables are as follows:

1. The number of iterations to approximate the zero (Iter).
2. The value of the computational order of convergence r_c approximated by (see ([15])

$$r_c = \frac{\log|f(x_k)/f(x_{k-1})|}{\log|f(x_{k-1})/f(x_{k-2})|}. \tag{55}$$

3. The errors $|x_{k+1} - x_k|$ of approximations to the corresponding zeros of the functions $f_1(x) - f_5(x)$.
4. In 1960, Ostrowski [28] defined the efficiency index of an iterative method as follows:

$$EI = r^{\frac{1}{\theta_f}}. \tag{56}$$

The R-order of convergence r and the number of function evaluations θ_f per iteration. The efficiency index is still an important indicator by comparing iterative methods for solving nonlinear equations.

The results of the numerical calculations shown in Tables 1 to 6 confirm that the degree of convergence of the methods proposed in Equations (TM4), (TM4.2), (TM4.3), (TM4.5), (TM5.2) and (TM6) is the same as theoretically proved in Theorems (2.1), (3.1) and (3.3).

Table 1: Numerical results of the method TM4.

functions		TM4, $H_1(t)$	TM4, $H_2(t)$	TM4, $H_3(t)$	TM4, $H_4(t)$	TM4, $H_5(t)$
$f_1, x_0 = 1.1$	$\lvert x_{n+1} - x_n\rvert$	0.00e-0	4.21e-7	2.54e-6	0.00e-0	0.00e-0
	$\lvert f(x_{n+1})\rvert$	2.42e-17	3.78e-24	6.26e-21	5.22e-15	3.22e-16
	Iter	3	3	3	3	3
	r_c	3.95	3.98	3.97	3.92	3.94
$f_2, x_0 = 1$	$\lvert x_{n+1} - x_n\rvert$	6.64e-8	2.94e-9	1.34e-8	2.10e-7	8.82e-8
	$\lvert f(x_{n+1})\rvert$	1.33e-28	3.45e-34	2.11e-31	1.58e-26	4.18e-28
	Iter	3	3	3	3	3
	r_c	3.99	3.99	3.99	3.99	3.99
$f_3, x_0 = 1$	$\lvert x_{n+1} - x_n\rvert$	2.63e-8	1.68e-8	2.15e-8	3.11e-8	1.75e-8
	$\lvert f(x_{n+1})\rvert$	3.23e-30	3.50e-31	1.19e-30	7.46e-30	3.97e-31
	Iter	3	3	3	3	3
	r_c	4.00	4.00	4.00	4.00	3.99
$f_4, x_0 = 0.6$	$\lvert x_{n+1} - x_n\rvert$	0.00e-0	0.00e-0	0.00e-0	0.00e-0	0.00e-0
	$\lvert f(x_{n+1})\rvert$	4.27e-9	6.20e-18	6.00e-17	1.75e-15	4.36e-16
	Iter	3	3	3	3	57
	r_c	3.98	4.02	4.03	4.06	4.05
$f_5, x_0 = -2.3$	$\lvert x_{n+1} - x_n\rvert$	1.10e-14	8.50e-15	9.76e-15	1.25e-14	0.00e-0
	$\lvert f(x_{n+1})\rvert$	2.17e-57	6.27e-58	1.20e-57	3.75e-57	4.36e-16
	Iter	3	3	3	3	10
	r_c	3.99	4.00	3.99	4.00	4.05

Table 2: Numerical results of the method TM4.2.

functions		TM4.2, $H_1(t)$	TM4.2, $H_2(t)$	TM4.2, $H_3(t)$	TM4.2, $H_4(t)$	TM4.2, $H_5(t)$
$f_1, x_0 = 1.1$	$\|x_{n+1} - x_n\|$	9.20e-7	1.94e-8	2.32e-8	5.60e-6	1.75e-6
	$\|f(x_{n+1})\|$	1.64e-24	3.72e-33	2.10e-31	4.52e-21	2.61e-23
	Iter	3	3	3	3	3
	r_c	4.09	4.37	4.03	4.08	4.07
$f_2, x_0 = 1$	$\|x_{n+1} - x_n\|$	2.55e-9	9.24e-11	1.41e-10	1.24e-8	3.38e-9
	$\|f(x_{n+1})\|$	8.28e-37	6.13e-45	2.76e-42	9.12e-34	2.75e-36
	Iter	3	3	3	3	3
	r_c	4.15	4.40	4.11	4.16	4.15
$f_3, x_0 = 1$	$\|x_{n+1} - x_n\|$	4.44e-9	1.60e-9	3.00e-9	5.88e-9	4.45e-9
	$\|f(x_{n+1})\|$	8.96e-36	2.88e-38	9.23e-37	4.18e-35	9.04e-36
	Iter	3	3	3	3	3
	r_c	4.27	4.44	4.28	4.26	4.27
$f_4, x_0 = 0.6$	$\|x_{n+1} - x_n\|$	2.29e-5	5.10e-6	1.28e-5	3.46e-5	2.37e-5
	$\|f(x_{n+1})\|$	2.39e-20	6.55e-24	1.02e-21	1.96e-19	2.73e-20
	Iter	3	3	3	3	3
	r_c	4.27	4.31	4.29	4.27	4.27
$f_5, x_0 = -2.3$	$\|x_{n+1} - x_n\|$	1.65e-16	4.25e-17	1.01e-16	2.36e-16	1.65e-16
	$\|f(x_{n+1})\|$	2.65e-70	9.87e-75	1.82e-70	1.66e-68	2.64e-69
	Iter	3	3	3	3	4
	r_c	4.22	4.40	4.23	4.22	4.22

Table 3: Numerical results of the method TM4.3.

functions		TM4.3, $H_1(t)$	TM4.3, $H_2(t)$	TM4.3, $H_3(t)$	TM4.3, $H_4(t)$	TM4.3(32), $H_5(t)$
$f_1, x_0 = 1.1$	$\|x_{n+1} - x_n\|$	9.76e-7	3.41e-8	9.29e-9	6.70e-6	1.86e-6
	$\|f(x_{n+1})\|$	7.37e-25	5.41e-33	1.87e-33	3.33e-21	1.15e-23
	Iter	3	3	3	3	3
	r_c	4.19	4.57	4.03	4.21	4.17
$f_2, x_0 = 1$	$\|x_{n+1} - x_n\|$	2.70e-9	1.24e-10	9.91e-11	1.37e-8	3.58e-9
	$\|f(x_{n+1})\|$	2.44e-37	1.15e-44	1.55e-43	3.19e-34	8.11e-37
	Iter	3	3	3	3	3
	r_c	4.25	4.58	4.16	4.27	4.25
$f_3, x_0 = 1$	$\|x_{n+1} - x_n\|$	4.46e-9	1.61e-9	3.02e-9	5.91e-9	4.47e-9
	$\|f(x_{n+1})\|$	2.79e-36	2.70e-40	2.92e-37	1.29e-34	2.82e-36
	Iter	3	3	3	3	3
	r_c	4.35	4.59	4.36	4.35	4.35
$f_4, x_0 = 0.6$	$\|x_{n+1} - x_n\|$	7.63e-10	9.87e-11	3.63e-10	1.35e-9	7.44e-10
	$\|f(x_{n+1})\|$	6.77e-40	3.03e-46	1.62e-41	1.06e-38	6.07e-40
	Iter	3	3	3	3	3
	r_c	4.32	4.57	4.32	4.32	4.32
$f_5, x_0 = -2.3$	$\|x_{n+1} - x_n\|$	1.65e-16	4.23e-17	1.01e-16	2.36e-16	1.65e-16
	$\|f(x_{n+1})\|$	1.85e-70	4.82e-77	1.27e-71	1.16e-69	1.84e-70
	Iter	3	3	3	3	3
	r_c	4.31	4.58	4.32	4.31	4.31

Table 4: Numerical results of the method TM4.5.

functions		TM4.5, $H_1(t)$	TM4.5, $H_2(t)$	TM4.5, $H_3(t)$	TM4.5, $H_4(t)$	TM4.5, $H_5(t)$
$f_1, x_0 = 1.1$	$\lvert x_{n+1} - x_n\rvert$	6.43e-7	2.79e-9	5.71e-8	2.99e-6	1.22e-6
	$\lvert f(x_{n+1})\rvert$	1.96e-27	8.13e-42	2.39e-32	2.52e-24	3.65e-26
	Iter	3	3	3	3	3
	r_c	4.52	4.98	4.54	4.51	4.51
$f_2, x_0 = 1$	$\lvert x_{n+1} - x_n\rvert$	1.01e-9	5.24e-12	1.36e-10	4.08e-9	1.35e-9
	$\lvert f(x_{n+1})\rvert$	9.97e-42	3.81e-57	7.99e-46	6.41e-39	3.58e-41
	Iter	3	3	3	3	3
	r_c	4.55	4.99	4.54	4.55	4.54
$f_3, x_0 = 1$	$\lvert x_{n+1} - x_n\rvert$	2.60e-10	1.57e-11	1.21e-10	4.01e-10	2.61e-10
	$\lvert f(x_{n+1})\rvert$	3.14e-44	2.25e-54	7.28e-45	2.66e-43	3.19e-43
	Iter	3	3	3	3	3
	r_c	4.53	4.91	4.50	4.54	4.53
$f_4, x_0 = 0.6$	$\lvert x_{n+1} - x_n\rvert$	1.35e-10	4.71e-12	5.53e-11	2.55e-10	1.32e-10
	$\lvert f(x_{n+1})\rvert$	1.02e-45	2.66e-57	1.24e-47	2.20e-44	9.14e-46
	Iter	3	3	3	3	3
	r_c	4.56	4.97	4.54	4.57	4.56
$f_5, x_0 = -2.3$	$\lvert x_{n+1} - x_n\rvert$	6.99e-18	1.40e-19	3.40e-18	1.09e-17	6.98e-18
	$\lvert f(x_{n+1})\rvert$	5.48e-81	4.83e-97	1.49e-82	5.04e-80	5.45e-81
	Iter	3	3	3	3	3
	r_c	4.55	4.98	4.54	4.55	4.55

Table 5: Numerical results of the method TM5.2.

functions		TM5.2, $H_1(t)$	TM5.2, $H_2(t)$	TM5.2, $H_3(t)$	TM5.2, $H_4(t)$	TM5.2, $H_5(t)$
$f_1, x_0 = 1.1$	$\lvert x_{n+1} - x_n\rvert$	2.64e-26	1.06e-10	1.34e-8	1.70e-6	5.99e-7
	$\lvert f(x_{n+1})\rvert$	6.14e-32	1.31e-58	1.04e-38	1.04e-26	3.68e-30
	Iter	3	3	3	3	3
	r_c	4.97	5.98	4.99	4.95	4.96
$f_2, x_0 = 1$	$\lvert x_{n+1} - x_n\rvert$	1.91e-10	5.36e-14	1.68e-11	1.01e-9	2.73e-10
	$\lvert f(x_{n+1})\rvert$	2.44e-49	1.94e-80	6.51e-55	1.51e-45	1.45e-48
	Iter	3	3	3	3	3
	r_c	4.99	5.99	4.99	4.99	4.99
$f_3, x_0 = 1$	$\lvert x_{n+1} - x_n\rvert$	8.38e-12	1.36e-14	4.07e-12	1.28e-11	8.42e-12
	$\lvert f(x_{n+1})\rvert$	9.60e-56	1.19e-83	1.30e-57	1.22e-54	9.77e-56
	Iter	3	3	3	3	3
	r_c	4.89	5.85	4.89	4.88	4.89
$f_4, x_0 = 0.6$	$\lvert x_{n+1} - x_n\rvert$	4.36e-12	6.31e-8	7.84e-7	2.36e-6	1.63e-6
	$\lvert f(x_{n+1})\rvert$	1.83e-57	7.85e-45	1.62e-30	1.26e-28	1.30e-29
	Iter	3	3	3	3	3
	r_c	4.93	6.09	4.96	4.94	4.95
$f_5, x_0 = -2.3$	$\lvert x_{n+1} - x_n\rvert$	6.71e-19	1.32e-21	3.18e-19	1.06e-18	6.70e-19
	$\lvert f(x_{n+1})\rvert$	1.19e-93	3.74e-127	1.43e-95	1.77e-92	1.18e-93
	Iter	3	3	3	3	3
	r_c	5.02	6.01	5.02	5.02	5.02

Table 6: Numerical results.

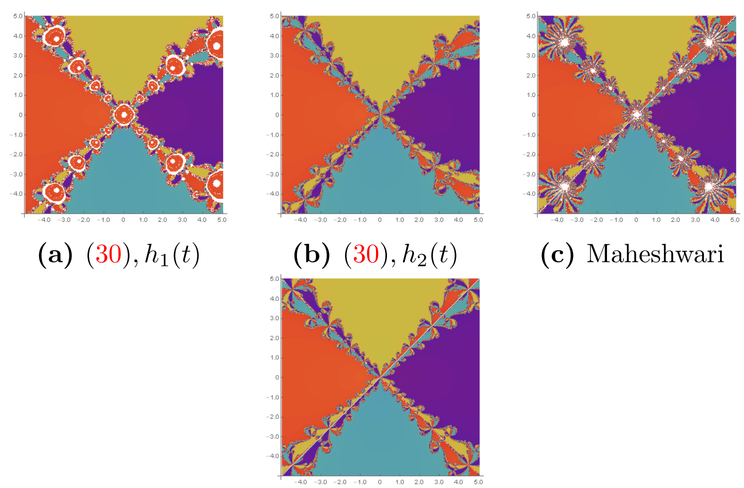
functions		TM [41]	CCTVM [6]	CPGM[8]	PDPM [30]	MLAM [27]
$f_1, x_0 = 1.1$	$\lvert x_{n+1} - x_n \rvert$	3.61e-40	0.00e-0	1.06e-122	8.95e-95	4.98e-208
	$\lvert f(x_{n+1}) \rvert$	2.68e-94	1.55e-15	8.05e-434	1.86e-397	7.38e-881
	Iter	6	3	3	3	5
	r_c	2.41	4.12	3.56	4.23	4.24
$f_2, x_0 = 1$	$\lvert x_{n+1} - x_n \rvert$	0.00e-0	0.00e-0	8.27e-174	3.58e-357	5.43e-247
	$\lvert f(x_{n+1}) \rvert$	4.51e-10	1.03e-10	1.67e-617	1.08e-1514	6.64e-1049
	Iter	3	3	3	3	5
	r_c	2.42	4.33	3.56	4.24	4.24
$f_3, x_0 = 1$	$\lvert x_{n+1} - x_n \rvert$	2.59e-9	0.00e-0	1.80e-116	3.36e-684	9.99e-159
	$\lvert f(x_{n+1}) \rvert$	4.28e-29	8.14e-16	2.46e-414	7.99e-2900	1.85e-671
	Iter	4	3	3	3	5
	r_c	2.44	4.07	3.57	4.24	4.24
$f_4, x_0 = 0.6$	$\lvert x_{n+1} - x_n \rvert$	1.02e-12	1.11e-6	3.03e-54	6.51e-178	8.61e-108
	$\lvert f(x_{n+1}) \rvert$	1.60e-29	2.20e-25	1.62e-191	1.81e-755	2.02e-454
	Iter	4	3	3	3	5
	r_c	2.37	4.08	3.57	4.24	4.24
$f_5, x_0 = -2.3$	$\lvert x_{n+1} - x_n \rvert$	3.02e-14	1.59e-12	3.28e-156	9.90e-1394	7.25e-269
	$\lvert f(x_{n+1}) \rvert$	2.02e-34	3.04e-48	2.28e-555	2.42e-5903	1.93e-1137
	Iter	3	3	3	3	5
	r_c	2.42	4.03	3.56	4.24	4.24

Table 7: Numerical results.

functions		TM6, $H_2(t)$	TM6, $H_7(t)$	TM6, $H_8(t)$	TM4.5, $H_6(t)$	TM4.3, $H_6(t)$
$f_1, x_0 = 1.1$	$\lvert x_{n+1} - x_n \rvert$	1.06e-10	4.22e-14	1.02e-14	9.10e-7	1.38e-6
	$\lvert f(x_{n+1}) \rvert$	1.31e-58	7.81e-79	1.65e-82	9.41e-27	3.23e-24
	Iter	3	3	3	3	3
	r_c	5.98	6.01	6.01	4.51	4.18
$f_2, x_0 = 1$	$\lvert x_{n+1} - x_n \rvert$	5.36e-14	8.21e-15	6.45e-15	1.17e-9	3.12e-9
	$\lvert f(x_{n+1}) \rvert$	1.94e-80	3.94e-85	9.78e-86	1.92e-41	4.52e-37
	Iter	3	3	3	3	3
	r_c	5.99	5.99	5.99	4.54	4.25
$f_3, x_0 = 1$	$\lvert x_{n+1} - x_n \rvert$	1.36e-14	9.17e-14	9.83e-14	2.61e-10	4.46e-9
	$\lvert f(x_{n+1}) \rvert$	1.19e-83	3.54e-78	5.55e-78	3.16e-44	2.80e-36
	Iter	3	3	3	3	3
	r_c	5.85	5.87	5.87	4.53	4.35
$f_4, x_0 = 0.6$	$\lvert x_{n+1} - x_n \rvert$	6.31e-8	5.74e-8	6.80e-8	1.33e-10	7.53e-10
	$\lvert f(x_{n+1}) \rvert$	7.85e-45	2.46e-43	6.21e-43	9.69e-46	6.41e-40
	Iter	3	3	3	3	3
	r_c	6.09	5.80	5.82	4.56	4.32
$f_5, x_0 = -2.3$	$\lvert x_{n+1} - x_n \rvert$	1.32e-21	1.42e-21	1.43e-21	6.99e-18	1.65e-16
	$\lvert f(x_{n+1}) \rvert$	3.74e-127	5.96e-127	6.19e-127	5.47e-81	1.84e-70
	Iter	3	3	3	3	3
	r_c	6.01	6.01	6.01	4.55	4.31

5 Basins of Attraction

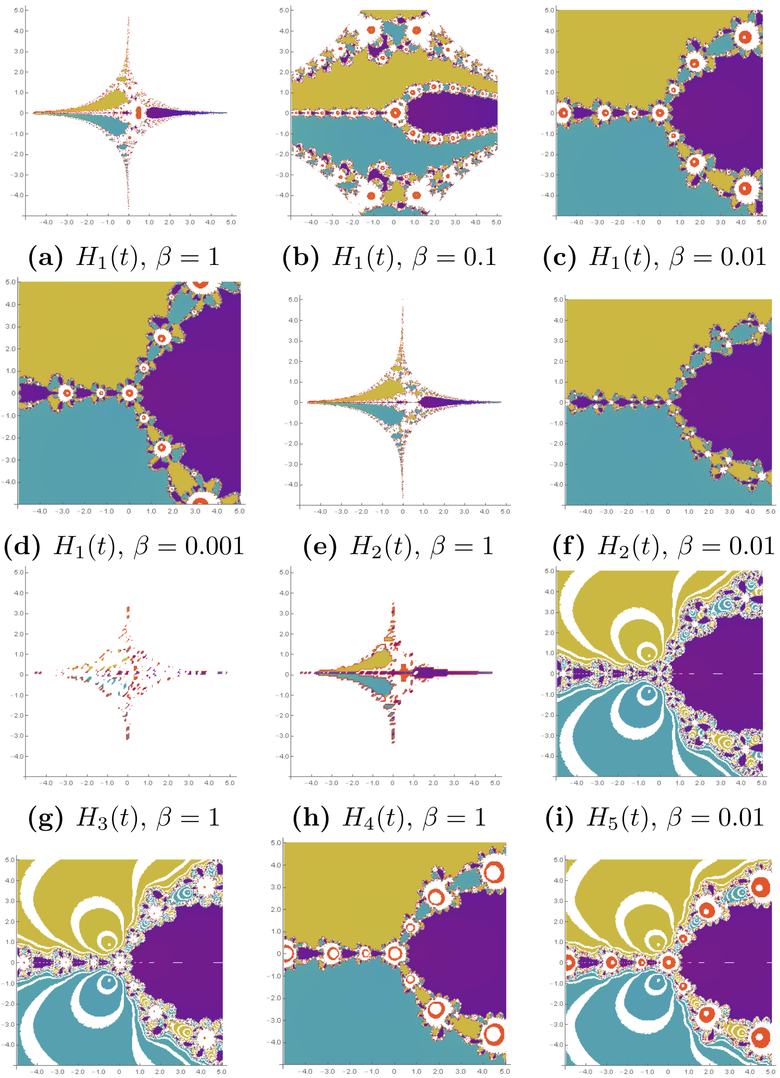
We now consider the basins of attraction of iterative root-finding methods to solve nonlinear equations. We have used the weight functions relations (28) and (29) for five different polynomials. The polynomials have complex roots and the work have a combination of real and complex ones. We have used the polynomial which they have simple zeros. We have taken the following polynomials. $f_1(z) = z^3 - 1,\ f_2(z) = z^2 - 1,\ f_3(z) = z^2 + 1, f_4(z) = z^3 - z,\ f_5(z) = z^4 - 1$. To create the attraction basins for the zeros of the polynomial, and an iterative method, we get a grid of 500×500 points n in a rectangle $D = [-5, 5] \times [-5, 5] \subset \mathbb{C}$. The criterion for stopping Mathematica programs is $|z_{old} - z_{new}| < 10^{-6}$. The comparison of the attraction basins Kung-Traub's method [20], Maheshwari's method [25] and methods (30) are shown in Figures (1) to (8). As can be seen, the adsorption region of the proposed methods competes with Kung-Traub and Maheshwari. In some cases, they also have a larger stability area and does not require the calculation of a function derivative.



(a) (30), $h_1(t)$ **(b)** (30), $h_2(t)$ **(c)** Maheshwari

(d) Kung-Traub

Figure 1: Comparison of the attraction basins of the proposed methods with other methods for finding the roots of the equation $f_5(z) = z^4 - 1$



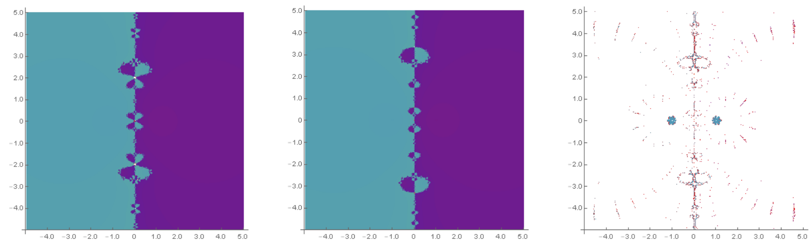
(a) $H_1(t)$, $\beta = 1$ **(b)** $H_1(t)$, $\beta = 0.1$ **(c)** $H_1(t)$, $\beta = 0.01$

(d) $H_1(t)$, $\beta = 0.001$ **(e)** $H_2(t)$, $\beta = 1$ **(f)** $H_2(t)$, $\beta = 0.01$

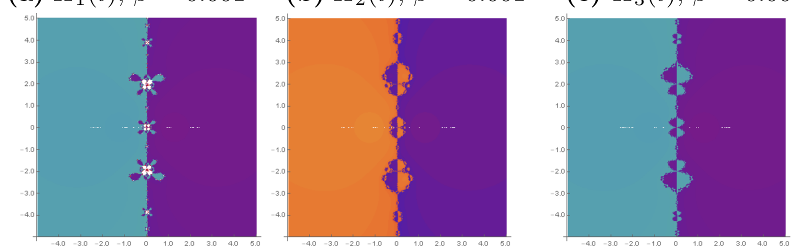
(g) $H_3(t)$, $\beta = 1$ **(h)** $H_4(t)$, $\beta = 1$ **(i)** $H_5(t)$, $\beta = 0.01$

(j) $H_6(t)$, $\beta = 0.01$ **(k)** $H_7(t_k)$, $\beta = 0.01$ **(l)** $H_8(t)$, $\beta = 0.01$

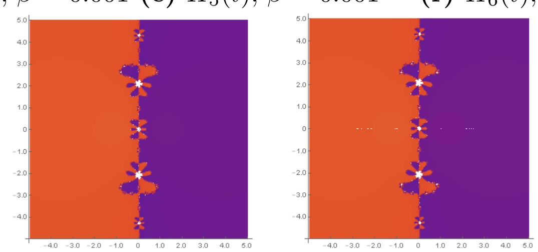
Figure 2: Method TM4 (15) for detecting the roots of the polynomial $f(z) = z^3 - 1$



(a) $H_1(t)$, $\beta = 0.001$ **(b)** $H_2(t)$, $\beta = 0.001$ **(c)** $H_3(t)$, $\beta = 0.001$

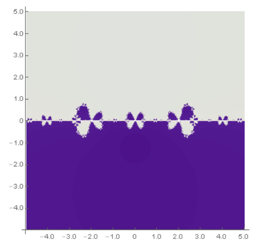


(d) $H_4(t)$, $\beta = 0.001$ **(e)** $H_5(t)$, $\beta = 0.001$ **(f)** $H_6(t)$, $\beta = 0.001$

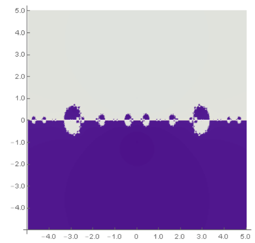


(g) H_7, $\beta = 0.001$ **(h)** $H_8(t)$, $\beta = 0.001$

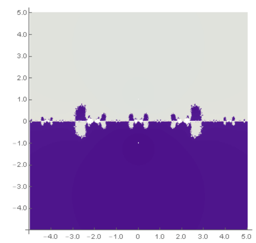
Figure 3: Method TM4 (15) for detecting the roots of the polynomial $f(z) = z^2 - 1$



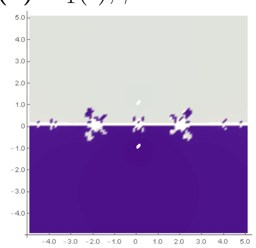
(a) $H_1(t)$, $\beta = 0.001$



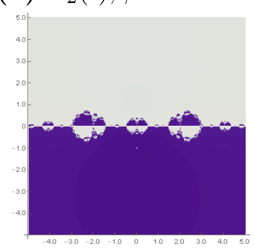
(b) $H_2(t)$, $\beta = 0.001$



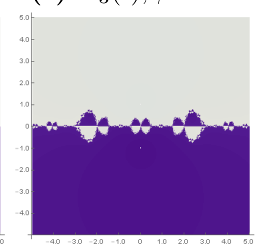
(c) $H_3(t)$, $\beta = 0.001$



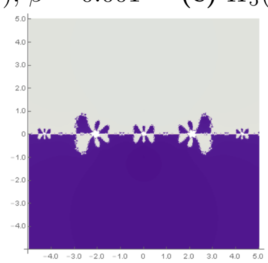
(d) $H_4(t)$, $\beta = 0.001$



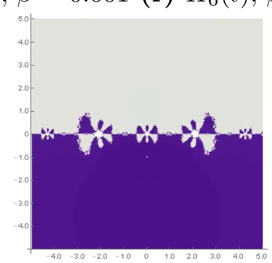
(e) $H_5(t)$, $\beta = 0.001$



(f) $H_6(t)$, $\beta = 0.001$

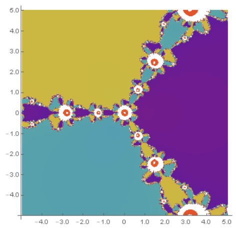


(g) $H_7(t_k)$, $\beta = 0.001$

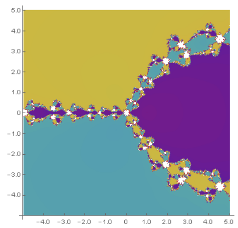


(h) $H_8(t)$, $\beta = 0.001$

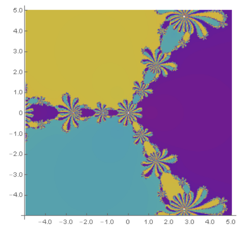
Figure 4: Method TM4 (15) for detecting the roots of the polynomial $f(z) = z^2 + 1$



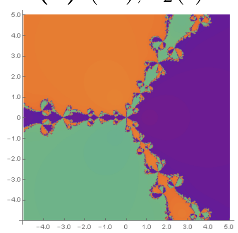
(a) (30), $h_1(t)$



(b) (30), $h_2(t)$

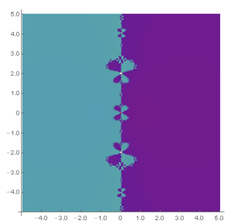


(c) Maheshwari

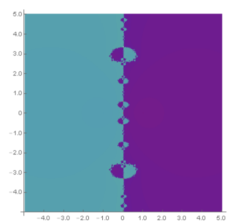


(d) Kung-Traub

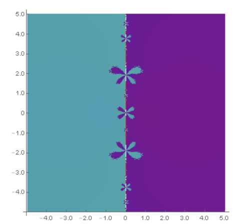
Figure 5: Comparison of the attraction basins of the proposed methods with other methods for detecting the roots of the equation $f_1(z) = z^3 - 1$



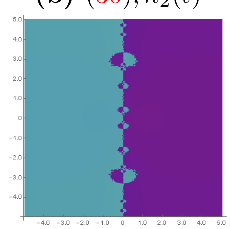
(a) (30), $h_1(t)$



(b) (30), $h_2(t)$

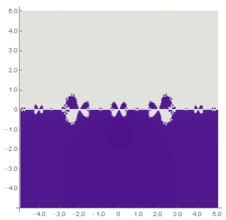


(c) Maheshwari

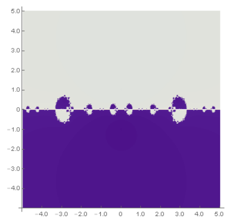


(d) Kung-Traub

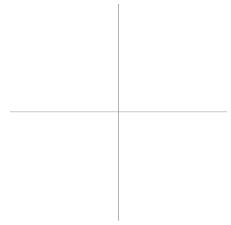
Figure 6: Comparison of the attraction basins of the proposed methods with other methods for detecting the roots of the equation $f_2(z) = z^2 - 1$



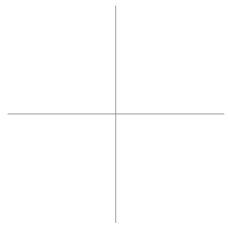
(a) (30), $h_1(t)$



(b) (30), $h_2(t)$

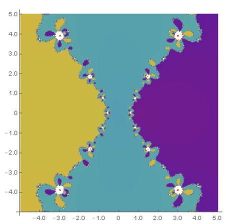


(c) Maheshwari

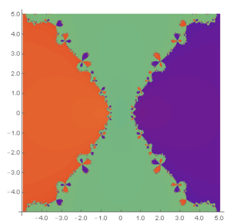


(d) Kung-Traub

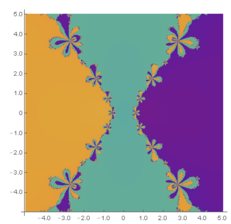
Figure 7: Comparison of the attraction basins of the proposed methods with other methods for finding the roots of the equation $f_3(z) = z^2 + 1$



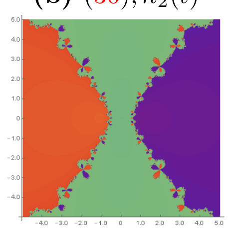
(a) (30), $h_1(t)$



(b) (30), $h_2(t)$



(c) Maheshwari



(d) Kung-Traub

Figure 8: Comparison of the attraction basins of the proposed methods with other methods for finding the roots of the equation $f_4(z) = z^3 - z$

6 Conclusion

In this work, we have constructed the with-memory without-derivative methods formulae having efficiency indices 1.58, 1.61, 1.62, 1.65, 1.73 and 1.81, respectively, which is higher than the efficiency index of the methods mentioned in the references of this article, especially references [5, 13]. From the analysis done in the with-memory methods section and the computational results shown in the above tables, we observe that the proposed methods have the efficiency index more than the other methods. Also, improving the convergence order of the new-family with-memory methods is higher than other previously proposed methods. We have shown that the best member of the weight function here is the $H_2(t) = \frac{1}{1-t}$. We found that the worst weight function here with maximum instability is the $H_3(t) = e^t$. Besides, the best value of the data parameter is $\beta = 0.001$.

References

[1] F. Ahmad, F. Soleymani, F. Khaksar Haghani, S. Serra-Capizzano, Higher order derivative-free iterative methods with and without memory for systems of nonlinear equations, *Applied Mathematics and Computation*, 314 (2017), 199-211.

[2] I. K. Argyros, On the Secant method for solving nonsmooth equations, *Journal of Mathematical Analysis and Applications*, 322 (2006), 146-157.

[3] D. K. R. Babajee, A. Cordero, F. Soleymani, J. R. Torregrosa, On improved three-step schemes with high efficiency index and their dynamics, *Numerical Algorithms*, 65 (1) (2014), 153-169.

[4] R. Behl, A. Cordero, S. S. Motsa, J. R. Torregrosa, V. Kanwar, An optimal fourth-order family of methods for multiple roots and its dynamics, *Numerical Algorithms*, 71 (4) (2016), 775-796.

[5] C. W. Chang, S. Qureshi, I. K. Argyros, F. I. Chicharro, A. Soomro, A modified two-step optimal iterative method for solving nonlinear

models in one and higher dimensions, *Mathematics and Computers in Simulation*, 229 (2025), 448-467.

[6] B. Campos, A. Cordero, J. R. Torregrosa, P. Vindel Caas, Stability of King's family of iterative methods with memory, *Journal of Computational and Applied Mathematics*, 318 (2017), 504-514.

[7] F. I. Chicharro, A. Cordero, N. Garrido, J. R. Torregrosa, On the choice of the best members of Kim's family and the improvement of its convergence, *Mathematical Methods in the Applied Sciences*, 43 (14)(2020), 8051-8066.

[8] N. Choubey, B.Panday, J. P. Jaiswal, Several two-point with memory iterative methods for solving nonlinear equations, *Afrika Matematika*, 29 (3-4) (2018), 435-449.

[9] C. Chun, Some fourth-order iterative methods for solving nonlinear equations, *Applied Mathematics and Computation*, 195 (2008), 454-459.

[10] C. Chun, B. Neta, An analysis of a new family of eighth-order optimal methods, *Applied Mathematics and Computation*, 245 (2014), 86-107.

[11] A. Cordero, M. Fardi, M. Ghasemi, J. R. Torregrosa, Accelerated iterative methods for finding solutions of nonlinear equations and their dynamical behavior, *Calcolo*, 51 (1) (2014), 17-30.

[12] A. Cordero, C. Jordán, E. Sanabria-Codesal, J. R. Torregrosa, Design, convergence and stability of a fourth-order class of iterative methods for solving nonlinear vectorial problems, *Fractal and Fractional*, 5 (3) (125) (2021), 1-16.

[13] A. Cordero, R.V. Rojas-Hiciano, J. R. Torregrosa, M. P. Vassileva, A highly efficient class of optimal fourth-order methods for solving nonlinear systems, *Numerical Algorithms*, 95 (4) (2024), 1879-1904.

[14] J. Herzberger, Über Matrixdarstellungen für Iterationverfahren bei nichtlinearen Gleichungen, *Computing*, 12 (1974) 215-222.

[15] I. O. Jay, A note on Q-order of convergence. *BIT Numerical Mathematics*, 41(2001), 422-429.

[16] M. D. Junjua, F. Zafar, N. Yasmin, Optimal derivative-free root finding methods based on inverse interpolation, *Mathematics*, 7 (2)(2019), 1-10.

[17] M. Kansal, V. Kanwar, S. Bhatia, Optimized mean based second derivative-free families of Chebyshev-Halley type methods, *Numerical Analysis and Applications*, 9 (2) (2016), 129-140.

[18] J. Kou, Y. Li, X. Wang, A family of fourth-order methods for solving non-linear equations, *Applied Mathematics and Computation*, 188(2007), 1031-1036.

[19] S. Kumar, J. R. Sharma, I. K. Argyros, Multi-step methods for equations, *Annali Dell Universita Di Ferrara*, 70 (2024), 1193-1215.

[20] H. T. Kung, J. F. Traub, Optimal order of one-point and multipoint iteration, *J. Assoc. Comput. Mach.*, 21(1974), 643-651.

[21] M. Lalehchini, T. Lotfi, K. Mahdiani, Adaptive Steffensen-like Methods with Memory for Solving Nonlinear Equations with the Highest Possible Efficiency Indices, *International Journal of Industrial Mathematics*, 11 (4) (2019), 337-345.

[22] M. Y. Lee, Y. I. Kim, A family of fast derivative-free fourth order multipoint optimal methods nonlinear equations, *International Journal of Computer Mathematics*, 89 (15)(2012), 2081-2093.

[23] T. Lotfi, A new optimal method of fourth-order convergence for solving nonlinear equations, *International Journal Industrial Mathematics*, 6 (2)(2014), 121-124.

[24] T. Lotfi, S. Sharifi, M. Salimi, S. Siegmund, A new class of three-point methods with optimal convergence order eight and its dynamics, *Numerical Algorithms*, 68 (2) (2015), 261-288.

[25] A. K. Maheshwari, A fourth order iterative method for solving nonlinear equations, *Applied Mathematics and Computation*, 211 (2009), 383-391.

[26] M. Moccari, T. Lotfi, V. Torkashvand, On the stability of a two-step method for a fourth-degree family by computer designs along with applications, *International Journal of Nonlinear Analysis and Applications*, 14 (4) (2023), 261-282.

[27] M. Mohamadi Zadeh, T. Lotfi, M. Amirfakhrian, Developing two efficient adaptive Newton-type methods with memory, *Mathematical Methods in the Applied Sciences*, 42 (17) (2019), 5687-5695.

[28] A. M. Ostrowski, *Solutions of Equations and System of Equations*, Academic Press, New York, (1960).

[29] M. S. Petković, L . D. Petković, Families of optimal multipoint methods for solving nonlinear equations: a survey, *Applicable Analysis and Discrete Mathematics*, 4 (2010), 1-22.

[30] M. S. Petković, J. Džunić, L . D. Petković, A family of two-point with memory for solving nonlinear equations, *Applicable Analysis and Discrete Mathematics*, 5 (2011), 298-317.

[31] M. Scott, B. Neta, C. Chun, Basin attractors for various methods, *Applied Mathematics and Computation*, 218 (2011), 2584-2599.

[32] F. Soleymani, Novel Computational Iterative Methods with Optimal Order for Nonlinear Equations, *Advances in Numerical Analysis*, 2011 (2011), 1-11.

[33] F. Soleymani, S. K. Khattri, S. Karimi Vanani, Two new classes of optimal Jarratt-type fourth-order methods, *Applied Mathematics Letters*, 25 (2012), 847-853.

[34] F. Soleymani, D. K. R. Babajee, T. Lotfi, On a numerical technique for finding multiple zeros and its dynamic, *Journal of the Egyptian Mathematical Society*, 21 (3) (2013), 346-353.

[35] R. Thukral, Further Development of Jarratt Method for Solving Nonlinear Equations, *Advances in Numerical Analysis*, 2012 (2012), 1-9.

[36] V. Torkashvand, T. Lotfi, M. A. Fariborzi Araghi, A new family of adaptive methods with memory for solving nonlinear equations, *Mathematical Sciences*, 13 (2019), 1-20.

[37] V. Torkashvand, M. Kazemi, On an Efficient Family with Memory with High Order of Convergence for Solving Nonlinear Equations, *International Journal of Industrial Mathematics*, 12 (2) (2020), 209-224.

[38] V. Torkashvand, M. A. Fariborzi Araghi, Construction of iterative adaptive methods with memory with 100% improvement of convergence order, *Journal of Mathematical Extension*, 15 (3) (16) (2021), 1-32.

[39] V. Torkashvand, M. Kazemi, M. J. Lalehchini, An efficient family of two-step with-memory methods with convergence order 6 and their dynamics, *Boletín de la Sociedad Matemática Mexicana*, 29 (3) (90) (2023), 1-19.

[40] V. Torkashvand, M. Momenzadeh, T.Lotfi, Creating a new two-step recursive memory method with eight-order based on Kung and Traub's method, *Proyecciones Journal of Mathematics*, 39 (5) (2020), 1167-1189.

[41] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice Hall, New York, USA, (1964).

[42] M. Z. Ullah, V. Torkashvand, S. Shateyi, M. Asma, Using matrix eigenvalues to construct an iterative method with the highest possible efficiency index two, *Mathematics*, 10 (1370) (2022), 1-15.

[43] F. Zafar, A. Cordero, J.R. Torregrosa, A. Rafi, A class of four parametric with-and without-memory root finding methods, *Computational and Mathematical Methods*, 1 (3) (2019), 1-13.

Vali Torkashvand
Instructor
Department of Mathematics
Farhangian University, Tehran, Iran.
ShQ.C., Islamic Azad University, Tehran, Iran

E-mail: torkashvand1978@gmail.com

Mohammad Ali Fariborzi Araghi
Full Professor of Applied Mathematics
Department of Mathematics
CT.C., Islamic Azad University
Tehran, Iran
E-mail: ma.fariborzi@iau.ac.ir