ISSN 1995-4239, Numerical Analysis and Applications, 2017, Vol. 10, No. 1, pp. 74–89. © Pleiades Publishing, Ltd., 2017.
Original Russian Text © N. Choubey, J. P. Jaiswal, 2017, published in Sibirskii Zhurnal Vychislitel'noi Matematiki, 2017, Vol. 20, No. 1, pp. 91–106.

# Two- and Three-Point with Memory Methods for Solving Nonlinear Equations

**N. Choubey[1*] and J. P. Jaiswal[2**]**

[1]*Department of Mathematics, Oriental Institute of Science and Technology, Bhopal, M.P., 462021 India*

[2]*Department of Mathematics, Maulana Azad National Institute of Technology, Bhopal, M.P., 462051 India*

Received April 21, 2016; in final form, May 26, 2016

**Abstract**—The main objective and inspiration in the construction of two- and three-point with memory method is to attain the utmost computational efficiency, without any additional function evaluations. At this juncture, we have modified the existing fourth and eighth order without memory method with optimal order of convergence by means of different approximations of self-accelerating parameters. The parameters have been calculated by Hermite interpolating polynomial, which accelerates the order of convergence of the without memory methods. In particular, the $R$-order convergence of the proposed two- and three-step with memory methods is increased from four to five and eight to ten. One more advantage of these methods is that the condition $f'(x) \neq 0$, in the neighborhood of the required root, imposed on Newton's method, can be removed. Numerical comparison is also stated to confirm the theoretical results.

**DOI:** 10.1134/S1995423917010086

Keywords: *iterative method, without memory scheme, with memory scheme, computational efficiency, numerical result.*

## 1. INTRODUCTION

Nowadays, solving nonlinear equation $f(x) = 0$ is a very important problem in real world phenomena. To find the solution of nonlinear equations, many iterative methods have been proposed (see [1–3, 6, 8, 11, 12]), where these iterative methods have an important area of research in numerical analysis because they have applications in many branches of pure and applied sciences. Out of them the most famous one-point iterative method without memory is Newton–Rapson method, which is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \tag{1.1}$$

and converges quadratically. One disadvantage of this method is the condition $f'(x_n) \neq 0$, which restricts its applications in practice. To resolve this problem, Kumar et al. [3] developed a new one-point iterative method, given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n) - \lambda_1 f(x_n)}. \tag{1.2}$$

If we take $\lambda_1 = 0$, then we obtain Newton method. The error expression for the above method is:

$$e_{n+1} = (\lambda_1 - c_2)e_n^2 + O(e_n^3), \tag{1.3}$$

*E-mail: nehachby2@gmail.com
**E-mail: asstprofjpmanit@gmail.com

where $e_n = x_n - \alpha$, $c_k = \frac{1}{k!}\frac{f^{(k)}(\alpha)}{f'(\alpha)}$, $k = 2, 3, \ldots$, and $\alpha$ is the root of $f(x) = 0$. Next, we discuss the classification of possible types of iteration function (IF). These IFs have been categorized on the basis of the information they require [4, 5].

**(i)** *One-point iterative method without memory.* In this type of methods $x_{k+1}$ can be determined by only new data at $x_k$. No old data are reused. Thus, $x_{k+1} = \phi(x_k)$, then $\phi$ will be called a one-point IF. The best known example of this type of method is Newton's IF.

**(ii)** *One-point iteration function with memory.* In this category $x_{k+1}$ can be determined by new data at $x_k$ and reused data at $x_{k-1}, \ldots, x_{k-n}$. Thus $x_{k+1} = \phi(x_k; x_{k-1}, \ldots, x_{k-n})$, then $\phi$ will be called a one-point IF with memory, because here $x_k$ is the new information, while $x_{k-1}, \ldots, x_{k-n}$ are reused information. The best known example of the with memory one-point method is a secant method. In the above mapping, the semicolon separates the point at which new data are used from the points at which old data are reused.

**(iii)** *Multi-point iteration function without memory.* In this type of methods $x_{k+1}$ can be determined by only new information at $x_k, w_1(x_k), \ldots, w_n(x_k)$ $(n \geq 1)$. No old data are reused. Thus, $x_{k+1} = \phi(x_k, w_1(x_k), \ldots, w_n(x_k))$, then $\phi$ is called a multipoint iteration function without memory. The multipoint IFs are useful because they avoid certain characteristic limitations over one-point IFs with and without memory.

**(iv)** *Multi-point iteration function with memory.* Finally, in this category, let us define another iteration function $\phi$ having arguments $z_j$, where each such argument represents $k+1$ quantities $x_j, w_1(x_j), \ldots, w_n(x_j)$ $(n \geq 1)$. Let the iteration mapping be defined by $x_{k+1} = \phi(z_k; z_{k-1}, \ldots, z_{k-n})$. Then $\phi$ is called a multipoint IF with memory. In the above-mentioned mapping, semicolon separates the points at which new information is used from the point at which old information is reused, i.e., at each iterative step we must preserve information of the last $n$ approximations $x_j$ and for each approximation, we must calculate $n$ expressions $w_1(x_j), \ldots, w_n(x_j)$.

Multipoint schemes are of immense practical importance, since they overcome theoretical limits of any one-point method in terms of convergence order and computational efficiency. Also, multipoint methods create approximations of higher accuracy; the high-speed development of digital computers, highly developed computer arithmetic and symbolic calculation allow for an even more efficient execution of multipoint methods. Multipoint methods with memory make use of information from the recent and preceding iterations. While the initial scheme for the construction of this class of methods date back to 1964 and Traub's book, the role of this area very rarely appears in the literature. To fill this gap we present two-step and three-step with memory schemes. The order of the convergence of the new with memory multipoint method is higher than that of the corresponding optimal without memory multipoint method. Improved convergence order is derived by several self-accelerating parameters. The accelerated convergence rate has been obtained without additional evaluation of function, which results in greater computational efficiency.

The main objective of this paper is to work on the multipoint iteration function with memory, because it can improve the order of convergence of the without memory methods, without using any additional calculations and it has very high computational efficiency. In this paper, we have presented two new multipoint iterative methods with memory, to solve the nonlinear equations followed by their convergence analysis. Sections 2 and 3 can be summarized as follows: In Section 2, we construct two-point and three-point iterative methods with memory. These methods have been obtained by employing a self-accelerating parameter. This parameter is calculated by the Hermite interpolating polynomial, where the $R$-order convergence of the two-point method is increased from 4 to 4.5616, 4.7913, 5, and that of the three-point method increased from 8 to 9, 9.5846, 9.7958, 10. At the end, the theoretical results are confirmed by considering different numerical examples.

## 2. CONVERGENCE ANALYSIS FOR WITH MEMORY METHODS

In the ensuing section, we will improve the convergence rate of the method by Behl and Kanwar [6] by replacing the parameter $T$ with $T_n$. First we consider the optimal fourth-order without memory scheme presented in [6]:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n) - Tf(x_n)},$$

$$x_{n+1} = y_n - \left(\frac{f(y_n)}{b_1 f'(x_n) - Tf(x_n)}\right)\left(\frac{b_3 f(x_n) + b_4 f(y_n)}{f(x_n) + b_2 f(y_n)}\right), \tag{2.1}$$

where $b_1 = \frac{1}{2}$, $b_3 = \frac{1}{2}$, $b_4 = \frac{b_2+2}{2}$, $b_2 = \gamma - 2$, and $T, \gamma \in \mathbb{R}$. The error equation for each step of (2.1) is

$$e_{n,y} = y_n - \alpha = (c_2 - T)e_n^2 + (2c_2^2 + T^2 - 2c_2 T - 2c_3)e_n^3$$

$$+(T^3 + 5Tc_2^2 - 4c_2^3 - 4Tc_3 + c_2(7c_3 - 3T^2) - 3c_4)e_n^4 + O(e_n^5) \tag{2.2}$$

and

$$e_{n+1} = (c_2 - T)[2(b_2+1)T^2 - (4b_2+7)Tc_2 + (2b_2+5)c_2^2 - c_3]e_n^4 + O(e_n^5), \tag{2.3}$$

where $e_{n,y} = y_n - \alpha$, $e_n = x_n - \alpha$ and $c_j = \frac{f^{(j)}(\alpha)}{j!f'(\alpha)}$, for $j = 2, 3, \ldots$. Substituting $T_n$ in place of $T$ in (2.1), we obtain the following iterative method with memory:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n) - T_n f(x_n)},$$

$$x_{n+1} = y_n - \left(\frac{f(y_n)}{b_1 f'(x_n) - T_n f(x_n)}\right)\left(\frac{b_3 f(x_n) + b_4 f(y_n)}{f(x_n) + b_2 f(y_n)}\right), \tag{2.4}$$

which is denoted by OM4. It is easy to recognize from (2.3) that the order of convergence of (2.1) is four when $T \neq c_2$. By taking the value of $T = c_2 = f''(\alpha)/(2f'(\alpha))$, it can be established that the order of the method (2.1) would be 5. For this type of acceleration of convergence and in actual fact the exact values of $f'(\alpha)$ and $f''(\alpha)$ are not obtainable. Otherwise, we could replace the parameter $T$ by $T_n$. To locate the values of the parameter, we can utilize the information accessible from the current and previous iteration and it satisfies $\lim_{n\to\infty} T_n = c_2 = f''(\alpha)/(2f'(\alpha))$, such that the fourth-order asymptotic convergence constant to be zero in (2.3). We consider the following formula for $T_n$:

**Method 1:**

$$T_n = \frac{H_2''(x_n)}{2f'(x_n)}, \tag{2.5}$$

where $H_2(x) = f(x_n) + f[x_n, x_n](x - x_n) + f[x_n, x_n, y_{n-1}](x - x_n)^2$ and $H_2''(x) = 2f[x_n, x_n, y_{n-1}]$.

**Method 2:**

$$T_n = \frac{H_3''(x_n)}{2f'(x_n)}, \tag{2.6}$$

where $H_3(x) = H_2(x) + f[x_n, x_n, y_{n-1}, x_{n-1}](x - x_n)^2(x - y_{n-1})$ and $H_3''(x) = 2f[x_n, x_n, y_{n-1}] + 2f[x_n, x_n, y_{n-1}, x_{n-1}](x_n - y_{n-1})$.

**Method 3:**

$$T_n = \frac{H_4''(x_n)}{2f'(x_n)}, \tag{2.7}$$

where $H_4(x) = H_3(x) + f[x_n, x_n, y_{n-1}, x_{n-1}, x_{n-1}](x - x_n)^2(x - y_{n-1})(x - x_{n-1})$, $H_4''(x) = 4f[x_n, x_n, y_{n-1}] + (2f[x_n, x_n, y_{n-1}, x_{n-1}] - 2f[x_n, y_{n-1}x_{n-1}, x_{n-1}])(x_n - y_{n-1})$ and $f[x_n, x_n] = f'(x_n)$, $f[x_n, y_n] = \frac{f(x_n)-f(y_n)}{x_n-y_n}$ are two first-order divided differences. The higher order divided difference of $f[x_n, x_n, t_0, t_1, \ldots, t_{m-2}]$ of order $m$ is defined as

$$f[x_n, x_n, t_0, t_1, \ldots, t_{m-2}] = \frac{f[x_n, t_0, t_1, \ldots, t_{m-2}] - f[x_n, x_n, t_0, t_1, \ldots, t_{m-3}]}{t_{m-2} - x_n},$$

$m \geq 2$, and we will use these notations throughout the paper.

**Note.** The Hermite interpolation polynomial $H_m(x)$ $(m = 2, 3, 4)$ satisfied the condition $H_m'(x_n) = f'(x_n)$ $(m = 2, 3, 4)$. So, $T_n = \frac{H_m''(x_n)}{2f'(x_n)}$ can be expressed as $T_n = \frac{H_m''(x_n)}{2H_m'(x_n)}$ $(m = 2, 3, 4)$.

**Theorem 1.** *Let $H_m$ be the Hermite interpolating polynomial of degree m that interpolates a function f at interpolation nodes $x_n, x_n, t_0 \ldots t_{m-2}$ contained in an interval I, and let the derivative $f^{(m+1)}$ be continuous in I and the Hermite interpolating polynomial $H_m(x_n) = f(x_n)$, $H_m'(x_n) = f'(x_n)$, and $H_m(t_j) = f(t_j)$ $(j = 0, 1, \ldots, m-2)$. Define the errors $e_{t,j} = t_j - \alpha$ $(j = 0, 1, \ldots, m-2)$ and assume that*

*(1) all nodes $x_n, t_0, \ldots, t_{m-2}$ are sufficiently close to the zero $\alpha$;*

*(2) the condition $e_n = O(e_{t,0} \ldots e_{t,m-2})$ holds.*

*Then*

$$H_m''(x_n) = 2f'(\alpha)\Big(c_2 - (-1)^{m-1}c_{m+1}\prod_{j=0}^{m-2} e_{t,j} + 3c_3e_n\Big), \tag{2.8}$$

$$T_n = \frac{H''(x_n)}{2f'(x_n)} \sim \Big(c_2 - (-1)^{m-1}c_{m+1}\prod_{j=0}^{m-2} e_{t,j} + (3c_3 - 2c_2^2)e_n\Big), \tag{2.9}$$

$$T_n - c_2 \sim \Big(-(-1)^{m-1}c_{m+1}\prod_{j=0}^{m-2} e_{t,j} + (3c_3 - 2c_2^2)e_n\Big). \tag{2.10}$$

*Proof.* The error expression of the Hermite interpolation can be uttered in this way:

$$f(x) - H_m(x_n) = \frac{f^{(m+1)}(\xi)}{(m+1)!}(x - x_n)^2\prod_{j=0}^{m-2}(x_n - t_j), \quad \xi \in I. \tag{2.11}$$

After twice differentiating (2.11) at the point $x = x_n$, we get

$$H_m''(x_n) = f''(x) - 2\frac{f^{(m+1)}(\xi)}{(m+1)!}\prod_{j=0}^{m-2}(x_n - t_j), \quad \xi \in I. \tag{2.12}$$

Taylor's series of derivative of $f$ at the point $x_n \in I$ and $\xi \in I$ about the zero $\alpha$ of $f$ provides

$$f'(x_n) = f'(\alpha)\big(1 + 2c_2e_n + 3c_3e_n^2 + O(e_n^3)\big), \tag{2.13}$$

$$f''(x_n) = f'(\alpha)\big(2c_2 + 6c_3 e_n + O(e_n^2)\big), \tag{2.14}$$

and

$$f^{(m+1)}(\xi) = f'(\alpha)\big((m+1)!c_{m+1} + (m+2)!c_{m+2}e_\xi + O(e_\xi^2)\big), \tag{2.15}$$

where $e_\xi = \xi - \alpha$. Putting (2.14), (2.15) in (2.12), we obtain

$$H''_m(x_n) = 2f'(\alpha)\left(c_2 - (-1)^{m-1}c_{m+1}\prod_{j=0}^{m-2} e_{t,j} + 3c_3 e_n\right), \tag{2.16}$$

which implies

$$\frac{H''_m(x_n)}{2f'(x_n)} \sim \left(c_2 - (-1)^{m-1}c_{m+1}\prod_{j=0}^{m-2} e_{t,j} + (3c_3 - 2c_2^2)e_n\right). \tag{2.17}$$

And hence,

$$T_n \sim \left(c_2 - (-1)^{m-1}c_{m+1}\prod_{j=0}^{m-2} e_{t,j} + (3c_3 - 2c_2^2)e_n\right), \tag{2.18}$$

or

$$T_n - c_2 \sim \left(-(-1)^{m-1}c_{m+1}\prod_{j=0}^{m-2} e_{t,j} + (3c_3 - 2c_2^2)e_n\right). \tag{2.19}$$

The conception of $R$-order of convergence [7] and the subsequent declaration (see [8, p. 287]) will be applied to approximate the convergence order of the iterative method (2.4) . □

**Theorem 2.** *If the errors of approximations $e_j = x_j - \alpha$ obtained in an iterative root finding method IM satisfy*

$$e_{k+1} \sim \prod_{i=0}^{m-2} (e_{k-i})^{m_i}, \quad k \geq k(\{e_k\}),$$

*then the R-order of convergence of IM, denoted with $O_R(IM,\alpha)$, satisfies the inequality $O_R(IM,\alpha) \geq s^*$, where $s^*$ is the unique positive solution of the equation $s^{n+1} - \sum_{i=0}^{n} m_i s^{n-i} = 0$.*

At this moment, we can state the following convergence theorem for the iterative method with memory (2.4).

**Theorem 3.** *Let the varying parameter $T_n$ in the iterative method* (2.4) *be calculated by* (2.5)–(2.7). *If an initial approximation $x_0$ is sufficiently close to a simple root $\alpha$ of $f(x)$, then the $R$-order of convergence of iterative methods* (2.4), (2.5) *and* (2.4), (2.6) *and* (2.4), (2.7) *with memory is at least* $(5+\sqrt{17})/2 \approx 4.5616$, $(5+\sqrt{21})/2 \approx 4.7913$ *and* 5, *respectively.*

*Proof.* Let the sequence $\{x_n\}$ be generated by an iterative method (IM) and converge to the root $\alpha$ of $f(x)$, with $R$-order $O_R(IM,\alpha) \geq r$, we write

$$e_{n+1} \sim D_{n,r} e_n^r. \tag{2.20}$$

If we take $n \to \infty$, then $D_{n,r}$ tends to the asymptotic error constant $D_r$ of IM. So

$$e_{n+1} \sim D_{n,r}(D_{n-1,r} e_{n-1}^r)^r = D_{n,r} D_{n-1,r}^r e_{n-1}^{r^2}. \tag{2.21}$$

The following error expression of the with memory method (2.4), can be obtained by (2.2), (2.3) and the varying parameter $T_n$:

$$e_{n,y} = y_n - \alpha \sim (T_n - c_2) e_n^2, \tag{2.22}$$

and

$$e_{n+1} = x_{n+1} - \alpha \sim B_{n,4}(T_n - c_2) e_n^4, \tag{2.23}$$

where $B_{n,4}$ is a varying quantity because of the self-accelerating parameter $T_n$ and it comes from (2.3). Here, we excluded higher order terms in (2.22) and (2.23).

**Method 1.** $T_n$ is calculated by (2.5), it is similar to the derivation of (2.20). We assume that the iterative sequence $\{y_n\}$ has the $R$-order $p$, then

$$e_{n,y} \sim D_{n,p} e_n^p \sim D_{n,p}(D_{n-1,r} e_{n-1}^r)^p = D_{n,p} D_{n-1,r}^p e_{n-1}^{rp}. \tag{2.24}$$

Using Theorem 1 for $m = 2$ and $t_0 = y_{n-1}$, we attain

$$T_n - c_2 \sim c_3 e_{t,0} = c_3 e_{n-1,y}. \tag{2.25}$$

Now from (2.22), (2.23), and (2.25), we get

$$e_{n,y} \sim c_3 e_{n-1,y}(D_{n-1,r} e_{n-1}^r)^2 \sim c_3 D_{n-1,p} D_{n-1,r}^2 e_{n-1}^{2r+p}, \tag{2.26}$$

$$e_{n+1} \sim B_{n,4} c_3 e_{n-1,y} e_n^4 \sim B_{n,4} c_3 D_{n-1,p} e_{n-1}^p (D_{n-1,r} e_{n-1}^r)^4, \sim B_{n,4} c_3 D_{n-1,p} D_{n-1,r}^4 e_{n-1}^{4r+p}. \tag{2.27}$$

By comparing the components of $e_{n-1}$ featuring in two pairs of relations (2.24), (2.26) and (2.21), (2.27), we obtain the subsequent system of equations:

$$\begin{aligned} 2r + p &= rp, \\ 4r + p &= r^2. \end{aligned} \tag{2.28}$$

Positive solution of system (2.28) is given by $r = (5 + \sqrt{17})/2$ and $p = (1 + \sqrt{17})/2$. Therefore, the $R$-order of the methods with memory (2.4), (2.5) is at least $(5 + \sqrt{17})/2 \approx 4.5616$.

**Method 2.** $T_n$ is calculated by (2.6). Using Theorem 1 for $m = 3$, $t_0 = y_{n-1}$, and $t_1 = x_{n-1}$, we get

$$T_n - c_2 \sim -c_4 e_{t,0} e_{t,1} = -c_4 e_{n-1,y} e_{n-1}. \tag{2.29}$$

In accordance with (2.22), (2.23) and (2.29), we find

$$e_{n,y} \sim (T_n - c_2) e_n^2 \sim -c_4 e_{n-1} e_{n-1,y}(D_{n-1,r} e_{n-1}^r)^2 \sim -c_4 D_{n-1,p} D_{n-1,r}^2 e_{n-1}^{2r+p+1}, \tag{2.30}$$

$$e_{n+1} \sim -B_{n,4}c_4 e_{n-1} e_{n-1,y} e_n^4 \sim -B_{n,4}c_4 e_{n-1} D_{n-1,p} e_{n-1}^p (D_{n-1,r} e_{n-1}^r)^4$$
$$\sim -B_{n,4}c_4 D_{n-1,p} D_{n-1,r}^4 e_{n-1}^{4r+p+1}. \tag{2.31}$$

By comparing the components of $e_{n-1}$ featuring in two pairs of relations (2.24), (2.30) and (2.21), (2.31), we obtain the subsequent system of equations:

$$\begin{aligned} 2r+p+1 &= rp, \\ 4r+p+1 &= r^2. \end{aligned} \tag{2.32}$$

Positive solution of system (2.32) is given by $r = (5+\sqrt{21})/2$ and $p = (1+\sqrt{21})/2$. Therefore, $R$-order of the methods with memory (2.4), (2.6) is at least $(5+\sqrt{21})/2 \approx 4.7913$.

**Method 3.** $T_n$ is calculated by (2.7). Hermite interpolating polynomial $H_4(x)$ satisfied the condition $H_4(x_n) = f(x_n)$, $H_4'(x_n) = f'(x_n)$, $H_4(y_{n-1}) = f(y_{n-1})$, $H_4(x_{n-1}) = f(x_{n-1})$, and $H_4'(x_{n-1}) = f'(x_{n-1})$. The error of the Hermite interpolation can be expressed as follows:

$$f(x) - H_4(x) = \frac{f^{(5)}(\xi)}{5!}(x-x_n)^2(x-x_{n-1})^2(x-y_{n-1}), \quad \xi \in I. \tag{2.33}$$

After twice differentiating (2.33) at the point $x = x_n$, we attain

$$H_4''(x_n) = f''(x_n) - 2\frac{f^{(5)}(\xi)}{5!}(x_n-x_{n-1})^2(x_n-y_{n-1}), \quad \xi \in I. \tag{2.34}$$

Taylor's series of derivatives of $f$ at the points $x_n \in I$ and $\xi \in I$ about the zero $\alpha$ of $f$ provide

$$f'(x_n) = f'(\alpha)\big(1 + 2c_2 e_n + 3c_3 e_n^2 + O(e_n^3)\big), \tag{2.35}$$

$$f''(x_n) = f'(\alpha)\big(2c_2 + 6c_3 e_n + O(e_n^2)\big), \tag{2.36}$$

$$f^{(m+1)}(\xi) = f'(\alpha)\big((m+1)!c_{m+1} + (m+2)!c_{m+2}e_\xi + O(e_\xi^2)\big), \tag{2.37}$$

where $e_\xi = \xi - \alpha$. Substituting (2.37) and (2.36) into (2.34), we obtain

$$H_4''(x_n) = 2f'(\alpha)\big(c_2 - c_5 e_{n-1,y} e_{n-1}^2 + 3c_3 e_n\big). \tag{2.38}$$

Using (2.22) and (2.23), we have

$$e_{n-1,y} = y_{n-1} - \alpha \sim (T_{n-1} - c_2)e_{n-1}^2, \tag{2.39}$$

$$e_n = x_n - \alpha \sim B_{n-1,4}(T_{n-1} - c_2)e_{n-1}^4. \tag{2.40}$$

Then

$$e_{n-1,y} e_{n-1}^2 \sim (T_{n-1} - c_2)e_{n-1}^4 \sim \frac{1}{B_{n-1,4}} e_n. \tag{2.41}$$

Now, substituting the value of (2.41) into (2.38), we attain

$$H_4''(x_n) = 2f'(a)\left(c_2 + \left(3c_3 - \frac{c_5}{B_{n-1,4}}\right) e_n\right), \tag{2.42}$$

which implies

$$\frac{H_4''(x_n)}{2f'(x_n)} \sim c_2 + \left(3c_3 - 2c_2^2 - \frac{c_5}{B_{n-1,4}}\right)e_n. \tag{2.43}$$

And hence,

$$T_n = \frac{H_4''(x_n)}{2f'(x_n)} \sim c_2 + \left(3c_3 - 2c_2^2 - \frac{c_5}{B_{n-1,4}}\right)e_n, \tag{2.44}$$

or

$$T_n - c_2 \sim \left(3c_3 - 2c_2^2 - \frac{c_5}{B_{n-1,4}}\right)e_n. \tag{2.45}$$

Putting the value of (2.45) into (2.23), we obtain

$$e_{n+1} \sim B_{n,4}\left(3c_3 - 2c_2^2 - \frac{c_5}{B_{n-1,4}}\right)e_n^5. \tag{2.46}$$

As a result, the $R$-order of the methods with memory (2.4), (2.7) is at least 5. Thus, the proof is completed. □

Next, we are considering the optimal eighth-order without memory scheme of the same paper [6], given by

$$y_n = x_n - \frac{f(x_n)}{f'(x_n) - Tf(x_n)},$$

$$z_n = y_n - \left(\frac{f(y_n)(f(x_n) + \gamma f(y_n))}{(f'(x_n) - 2Tf(x_n))(f(x_n) + (\gamma - 2)f(y_n))}\right),$$

$$x_{n+1} = z_n - f(z_n)(f[z_n, y_n] + f[z_n, y_n, x_n](z_n - y_n) + f[z_n, y_n, x_n, x_n](z_n - y_n)(z_n - x_n))^{-1}, \tag{2.47}$$

where $T, \gamma \in \mathbb{R}$. The error expressions for each step of the method (2.47) are

$$e_{n,y} = y_n - \alpha = (T - c_2)e_n^2 + O(e_n^3), \tag{2.48}$$

$$e_{n,z} = z_n - \alpha = (c_2 - T)[2(b_2 + 1)T^2 - (4b_2 + 7)Tc_2 + (2b_2 + 5)c_2^2 - c_3]e_n^4 + O(e_n^5), \tag{2.49}$$

$$\begin{aligned} e_{n+1} = {} & [(T - c_2)^2(2(\gamma - 1)T^2 - (4\gamma - 17)Tc_2 + (2\gamma + 1)c_2^2 - c_3) \\ & \times(-T(4\gamma - 1)c_2^2 + (2\gamma + 1)c_2^3 + c_2(2T^2(\gamma - 1) - c_3) + c_4)]e_n^8 + O(e_n^9). \end{aligned} \tag{2.50}$$

It is easy to recognize from (2.50) that the order of convergence of (2.47) is eighth when $T \neq c_2$. By captivating the value of $T = c_2 = f''(\alpha)/(2f'(\alpha))$, it can be established that the order of the method (2.47) would be 10. For this type of acceleration of convergence and in actual fact the exact values of $f'(\alpha)$ and $f''(\alpha)$ are not obtainable. But we could replace the parameter $T$ by $T_n$. To locate the values of the parameter, we can utilize the information accessible from the current and previous iteration and it satisfies $\lim_{n\to\infty} T_n = c_2 = f''(\alpha)/(2f'(\alpha))$, such that the eighth-order asymptotic convergence constant to be zero in (2.50). We consider the following formula for $T_n$:

**Method 4:**

$$T_n = \frac{H_2''(x_n)}{2f'(x_n)}, \tag{2.51}$$

where $H_2(x) = f(x_n) + f[x_n, x_n](x - x_n) + f[x_n, x_n, z_{n-1}](x - x_n)^2$ and $H_2''(x_n) = 2f[x_n, x_n, z_{n-1}]$.

**Method 5:**

$$T_n = \frac{H_3''(x_n)}{2f'(x_n)}, \tag{2.52}$$

where $H_3(x) = H_2(x) + f[x_n, x_n, z_{n-1}, y_{n-1}](x - x_n)^2(x - z_{n-1})$ and $H_3''(x) = H_2''(x_n) + 2f[x_n, x_n, z_{n-1}, y_{n-1}](x_n - z_{n-1})$.

**Method 6:**

$$T_n = \frac{H_4''(x_n)}{2f'(x_n)}, \tag{2.53}$$

where $H_4(x) = H_2(x) + f[x_n, x_n, z_{n-1}, y_{n-1}](x - x_n)^2(x - z_{n-1}) + f[x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}](x - x_n)^2(x - z_{n-1})(x - y_{n-1})$ and $H_4''(x_n) = H_2''(x_n) + 2f[x_n, x_n, z_{n-1}, y_{n-1}](x_n - z_{n-1}) + 2f[x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}](x_n - z_{n-1})(x_n - y_{n-1})$.

**Method 7:**

$$T_n = \frac{H_5''(x_n)}{2f'(x_n)}, \tag{2.54}$$

where $H_5(x) = H_4(x) + f[x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}, x_{n-1}](x - x_n)^2(x - z_{n-1})(x - y_{n-1})(x - x_{n-1})$ and $H_5''(x_n) = H_4''(x_n) + 2f[x_n, x_n, z_{n-1}, y_{n-1}, x_{n-1}, x_{n-1}](x_n - z_{n-1})(x_n - y_{n-1})(x_n - x_{n-1})$. Then $T_n$ can be calculated by using the above three equations. Substituting $T_n$ in place of $T$ in (2.47), we obtain the following iterative method with memory:

$$y_n = x_n - \frac{f(x_n)}{f'(x_n) - T_n f(x_n)},$$

$$z_n = y_n - \left( \frac{f(y_n)(f(x_n) + \gamma f(y_n))}{(f'(x_n) - 2T_n f(x_n))(f(x_n) + (\gamma - 2)f(y_n))} \right),$$

$$x_{n+1} = z_n - f(z_n)(f[z_n, y_n] + f[z_n, y_n, x_n](z_n - y_n) + f[z_n, y_n, x_n, x_n](z_n - y_n)(z_n - x_n))^{-1}. \tag{2.55}$$

For $\gamma = 1$ the method is denoted by OM81 and for $\gamma = 0$ the method is denoted by OM82.

**Theorem 4.** *Let the varying parameter $T_n$ in the iterative method* (2.55) *be calculated by* (2.51)–(2.54). *If an initial approximation $x_0$ is sufficiently close to a simple root $\alpha$ of $f(x)$, then the R-order of convergence of iterative methods* (2.55) *with the corresponding expressions* (2.51)–(2.54) *of $T_n$ is at least* 9, $(5 + \sqrt{21}) \approx 9.5826$, $(5 + \sqrt{23}) \approx 9.7958$, *and* 10, *respectively.*

*Proof.* Let the sequence $\{x_n\}$ be generated by an iterative method (IM) and converge to the root $\alpha$ of $f(x)$ with $R$-order $O_R(IM, \alpha) \geq r$. We state

$$e_{n+1} \sim D_{n,r} e_n^r. \tag{2.56}$$

If we take $n \to \infty$, then $D_{n,r}$ tends to the asymptotic error constant $D_r$ of IM. So,

$$e_{n+1} \sim D_{n,r}(D_{n-1,r} e_{n-1}^r)^r = D_{n,r} D_{n-1,r}^r e_{n-1}^{r^2}. \tag{2.57}$$

The following error expression of the with memory method (2.55) can be obtained by (2.48)–(2.50) and the varying parameter $T_n$:

$$e_{n,y} = y_n - \alpha \sim (T_n - c_2)e_n^2, \tag{2.58}$$

$$e_{n,z} = z_n - \alpha \sim B_{n,4}(T_n - c_2)e_n^4, \tag{2.59}$$

$$e_{n+1} = x_{n+1} - \alpha \sim B_{n,8}(T_n - c_2)^2 e_n^8, \tag{2.60}$$

where $B_{n,4}$ and $B_{n,8}$ are two varying quantities because of the self-accelerating parameter $T_n$ and they come from (2.49) and (2.50). Here, we excluded higher order terms in (2.59) and (2.60).

**Method 4.** $T_n$ is calculated by (2.51). It is similar to the derivation of (2.56). Let us suppose that $p$ and $q$ are the $R$-order of the iteration sequences $\{y_n\}$ and $\{z_n\}$, respectively, then we have

$$e_{n,y} \sim D_{n,p}e_n^p \sim D_{n,p}(D_{n-1,r}e_{n-1}^r)^p = D_{n,p}D_{n-1,r}^p e_{n-1}^{rp}, \tag{2.61}$$

$$e_{n,z} \sim D_{n,q}e_n^q \sim D_{n,q}(D_{n-1,r}e_{n-1}^r)^q = D_{n,q}D_{n-1,r}^q e_{n-1}^{rq}. \tag{2.62}$$

By taking $m = 2$ and $t_0 = z_{n-1}$ for Theorem 1, we attain

$$T_n - c_2 \sim c_3 e_{t,0} = c_3 e_{n-1,z}. \tag{2.63}$$

Now from (2.58)–(2.60) and (2.63), we get

$$e_{n,y} \sim c_3 e_{n-1,z}(D_{n-1,r}e_{n-1}^r)^2 \sim c_3 D_{n-1,q}D_{n-1,r}^2 e_{n-1}^{2r+q}, \tag{2.64}$$

$$e_{n,z} \sim c_3 e_{n-1,z}B_{n,4}e_n^4 \sim c_3 D_{n-1,q}B_{n,4}D_{n-1,r}^4 e_{n-1}^{4r+q}, \tag{2.65}$$

$$e_{n+1} \sim c_3^2 B_{n,8}e_{n-1,z}^2 e_n^8 \sim c_3^2 B_{n,8}D_{n-1,q}^2 e_{n-1}^{2q} D_{n-1,r}^8 e_{n-1}^{8r+2q}. \tag{2.66}$$

By comparing the components of $e_{n-1}$ featuring in three pairs of relations (2.61), (2.64), (2.62), (2.65) and (2.57), (2.66), we obtain the subsequent system of equations:

$$\begin{aligned} 2r + q &= rp, \\ 4r + q &= rq, \\ 8r + 2q &= r^2. \end{aligned} \tag{2.67}$$

Positive solution of system (2.67) is given by $r = 9$, $q = 4.5$, and $p = 2.5$. Therefore, the $R$-order of the methods with memory (2.55), when $T_n$ is calculated by (2.51), is at least 9.

**Method 5.** $T_n$ is calculated by (2.52). By taking $m = 3$, $t_0 = z_{n-1}$ and $t_1 = y_{n-1}$ for Theorem 1, we attain

$$T_n - c_2 \sim -c_4 e_{t,0}e_{t,1} = -c_4 e_{n-1,z}e_{n-1,y}. \tag{2.68}$$

In accordance with (2.68) into (2.58)–(2.60), we find

$$\begin{aligned} e_{n,y} &\sim (T_n - c_2)e_n^2 \sim -c_4 2 e_{n-1,z}e_{n-1,y}(D_{n-1,r}e_{n-1}^r)^2 \\ &\sim -c_4 D_{n-1,q}D_{n-1,p}D_{n-1,r}^2 e_{n-1}^{2r+q+p}, \end{aligned} \tag{2.69}$$

$$e_{n,z} \sim (T_n - c_2)B_{n,4}e_n^4 \sim -c_4 B_{n,4} e_{n-1,z} e_{n-1,y} (D_{n-1,r} e_{n-1}^r)^4$$

$$\sim -c_4 B_{n,4} D_{n-1,q} D_{n-1,p} D_{n-1,r}^4 e_{n-1}^{4r+q+p}, \tag{2.70}$$

$$e_{n+1} \sim -B_{n,8} c_4^2 e_{n-1,z}^2 e_{n-1,y}^2 e_n^8$$

$$\sim -B_{n,8} c_4^2 D_{n-1,q}^2 D_{n-1,p}^2 D_{n-1,r}^8 e_{n-1}^{8r+2q+2p}. \tag{2.71}$$

By comparing the components of $e_{n-1}$ featuring in three pairs of relations (2.61), (2.69), (2.62), (2.70) and (2.57), (2.71), we obtain the subsequent system of equations:

$$\begin{aligned} 2r + p + q &= rp, \\ 4r + p + q &= rq, \\ 8r + 2p + 2q &= r^2. \end{aligned} \tag{2.72}$$

Positive solution of system (2.72) is given by $r = (5 + \sqrt{21})$, $q = (5 + \sqrt{21})/2$, and $p = (1 + \sqrt{21})/2$. Therefore, the $R$-order of the methods with memory (2.55), when $T_n$ is calculated by (2.52), is at least 9.5826.

**Method 6.** $T_n$ is calculated by (2.53). By taking $m = 4$, $t_0 = z_{n-1}$, $t_1 = y_{n-1}$, and $t_2 = x_{n-1}$ for Theorem 1, we attain

$$T_n - c_2 \sim c_5 e_{t,0} e_{t,1} e_{t,2} = c_5 e_{n-1,z} e_{n-1,y} e_{n-1}. \tag{2.73}$$

In accordance with (2.73) into (2.58)–(2.60), we find

$$e_{n,y} \sim (T_n - c_2) e_n^2 \sim c_5 e_{n-1} e_{n-1,z} e_{n-1,y} (D_{n-1,r} e_{n-1}^r)^2$$

$$\sim c_5 D_{n-1,q} D_{n-1,p} D_{n-1,r}^2 e_{n-1}^{2r+q+p+1}, \tag{2.74}$$

$$e_{n,z} \sim (T_n - c_2) B_{n,4} e_n^4 \sim c_5 e_{n-1} e_{n-1,z} e_{n-1,y} (D_{n-1,r} e_{n-1}^r)^4$$

$$\sim c_5 B_{n,4} D_{n-1,q} D_{n-1,p} D_{n-1,r}^4 e_{n-1}^{4r+q+p+1}, \tag{2.75}$$

$$e_{n+1} \sim B_{n,8} c_5^2 e_{n-1}^2 e_{n-1,z}^2 e_{n-1,y}^2 e_n^8$$

$$\sim B_{n,8} c_5^2 D_{n-1,q}^2 D_{n-1,p}^2 D_{n-1,r}^8 e_{n-1}^{8r+2q+2p+2}. \tag{2.76}$$

By comparing the components of $e_{n-1}$ featuring in three pairs of relations (2.61), (2.74), (2.62), (2.75) and (2.57), (2.76), we obtain the subsequent system of equations:

$$\begin{aligned} 2r + p + q + 1 &= rp, \\ 4r + p + q + 1 &= rq, \\ 8r + 2p + 2q + 2 &= r^2. \end{aligned} \tag{2.77}$$

Positive solution of system (2.77) is given by $r = (5 + \sqrt{23})$, $q = (5 + \sqrt{23})/2$, and $p = (1 + \sqrt{23})/2$. Therefore, the $R$-order of the methods with memory (2.55), when $T_n$ is calculated by (2.53), is at least $(5 + \sqrt{23}) \approx 9.7958$.

**Method 7.** $T_n$ is calculated by (2.54). Hermite interpolating polynomial $H_5(x)$ satisfied the conditions $H_5(x_n) = f(x_n)$, $H_5'(x_n) = f'(x_n)$, $H_5(z_{n-1}) = f(z_{n-1})$, $H_5(y_{n-1}) = f(y_{n-1})$, $H_5(x_{n-1}) = f(x_{n-1})$, and $H_5'(x_{n-1}) = f'(x_{n-1})$. The error expression of the Hermite interpolation can be uttered in this way:

$$f(x) - H_5(x) = \frac{f^{(6)}(\xi)}{6!}(x - x_n)^2(x - x_{n-1})^2(x - z_{n-1})(x - y_{n-1}), \quad \xi \in I. \tag{2.78}$$

After twice differentiating (2.78) at the point $x = x_n$, we get

$$H_5''(x_n) = f''(x_n) - 2\frac{f^{(6)}(\xi)}{6!}(x_n - x_{n-1})^2(x_n - z_{n-1})(x_n - y_{n-1}), \quad \xi \in I. \tag{2.79}$$

Taylor's series of derivatives of $f$ at the points $x_n \in I$ and $\xi \in I$ about the zero $\alpha$ of $f$ provide

$$f'(x_n) = f'(\alpha)\big(1 + 2c_2e_n + 3c_3e_n^2 + O(e_n^3)\big), \tag{2.80}$$

$$f''(x_n) = f'(\alpha)\big(2c_2e + 6c_3e_n + O(e_n^2)\big), \tag{2.81}$$

$$f^{(m+1)}(\xi) = f'(\alpha)\big((m+1)!c_{m+1} + (m+2)!c_{m+2}e_\xi + O(e_\xi^2)\big), \tag{2.82}$$

where $e_\xi = \xi - \alpha$. Substituting (2.81) and (2.82) into (2.79), we obtain

$$H_5''(x_n) = 2f'(\alpha)\big(c_2 - c_6e_{n-1,y}e_{n-1,z}e_{n-1}^2 + 3c_3e_n\big). \tag{2.83}$$

Using (2.58)–(2.60), we have

$$e_{n-1,y} = y_{n-1} - \alpha \sim (T_{n-1} - c_2)e_{n-1}^2, \tag{2.84}$$

$$e_{n-1,z} = z_{n-1} - \alpha \sim B_{n-1,4}(T_{n-1} - c_2)e_{n-1}^4, \tag{2.85}$$

$$e_n = x_n - \alpha \sim B_{n-1,8}(T_{n-1} - c_2)^2e_{n-1}^8. \tag{2.86}$$

Then

$$e_{n-1,y}e_{n-1,z}e_{n-1}^2 \sim B_{n-1,4}(T_{n-1} - c_2)^2e_{n-1}^8 \sim \frac{B_{n-1,4}}{B_{n-1,8}}e_n. \tag{2.87}$$

Now, substituting the value (2.87) into (2.83):

$$H_5''(x_n) = 2f'(\alpha)\left(c_2 + \left(3c_3 - c_6\frac{B_{n-1,4}}{B_{n-1,8}}\right)e_n\right), \tag{2.88}$$

which implies

$$\frac{H_5''(x_n)}{2f'(x_n)} \sim c_2 + (3c_3 - 2c_2^2 - c_6\frac{B_{n-1,4}}{B_{n-1,8}})e_n. \tag{2.89}$$

And hence,

$$T_n = \frac{H_5''(x_n)}{2f'(x_n)} \sim c_2 + \left(3c_3 - 2c_2^2 - c_6\frac{B_{n-1,4}}{B_{n-1,8}}\right)e_n, \tag{2.90}$$

or

$$T_n - c_2 \sim \left(3c_3 - 2c_2^2 - c_6 \frac{B_{n-1,4}}{B_{n-1,8}}\right) e_n. \tag{2.91}$$

Substituting (2.91) into (2.60), we get

$$e_{n+1} \sim B_{n,8} \left(3c_3 - 2c_2^2 - c_6 \frac{B_{n-1,4}}{B_{n-1,8}}\right)^2 e_n^{10}. \tag{2.92}$$

Therefore, the $R$-order of the methods with memory (2.55), when $T_n$ is calculated by (2.54), is at least 10. □

## 3. NUMERICAL COMPARISON

In this section, after the review of the with memory methods, we have compared the presented schemes with the two step methods given in [9] and the three-point methods studied in [10]. Table 1 is furnished with the considered nonlinear test functions with their roots ($\alpha$). In the same table, there are infinite number of digits after decimal, but we have mentioned only four (the nonlinear functions are taken from [10, 11]. In Tables 2 and 3, the absolute errors $|x_k - \alpha|$ are given for the presented methods OM4 and OM81, respectively. The computational order of convergence (COC) is approximated by using the formula (see [12])

$$\text{COC} \approx \frac{\ln |f(x_{n+1})/f(x_n)|}{\ln |f(x_n)/f(x_{n-1})|},$$

to check the computational efficiency, which verified theoretical rate of convergence.

All the numerical results revealed in Tables 2 and 3 of two- and three-step with memory methods are in concordance with the theory built up in this paper. For this we have considered up to 1000 significant digits by using *"Set Accuracy"* command in "Mathematica 8." Our proposed schemes OM4 with (2.5)–(2.7) and OM81 with (2.51)–(2.54) have been used to solve the nonlinear functions and the calculated results are compared with the two-step methods XW41(16–18), XW42(16–19), XW43(16–20), XW44(17–18), XW45(17–19), and XW46(17–20) of [9] and the three-point methods XW81(37–34), XW82(37–35), XW83(37–36), XW84(38–34), XW85(38–35), and XW86(38–36) of [10]. In Table 3 "NC" means not convergent. From Tables 2 and 3, it is very easy to identify that the results obtained by the proposed methods are quite superior to the other two- and three-step methods. An additional effective approach to compare the efficiency of methods is CPU time used in the implementation of program. At this point, the CPU time has been computed by means of the command "*TimeUsed* [ ]" in "Mathematica 8." The CPU time depends on the specification of computer. The computer characteristics are Microsoft Windows 7 Intel Core i3-2330M CPU@ 2.20 GHz with 2 GB of RAM, 64-bit operating system throughout the paper. The mean CPU time is calculated by considering the mean of 30 performances of the program and is given in Tables 2 and 3.

**Table 1.** Test functions

| Nonlinear function | Root |
|---|---|
| $f_1 = xe^{x^2} - (\sin x)^2 + 3\cos x + 5$ | $-1.2076\ldots$ |
| $f_2 = x^5 + x^4 + 4x^2 - 15$ | $1.3474\ldots$ |
| $f_3 = x^3 - x^2 - 1$ | $1.4655\ldots$ |

**Table 2.** Numerical comparison of two-point with memory method

| Method | $\lvert x_1-\alpha\rvert$ | $\lvert x_2-\alpha\rvert$ | $\lvert x_3-\alpha\rvert$ | COC | CUP time |
|---|---|---|---|---|---|
| Example $f_1$, guess: $-1.6$ | | | | | |
| XW41(16–18), $T_0=-0.01,\ a=8$ | 3.5037e−2 | 2.8246e−6 | 1.2080e−25 | 4.7042 | 2.5721 |
| XW42(16–19), $T_0=-0.01,\ a=8$ | 3.5037e−2 | 4.7605e−7 | 1.7515e−27 | 4.1782 | 2.6509 |
| XW43(16–20), $T_0=-0.01,\ a=8$ | 3.5037e−2 | 3.4949e−7 | 4.1255e−28 | 4.1649 | 2.8477 |
| XW44(17–18), $T_0=-0.01,\ b=-2$ | 1.8398e−2 | 2.1773e−7 | 1.3052e−30 | 4.7018 | 2.4728 |
| XW45(17–19), $T_0=-0.01,\ b=-2$ | 1.8398e−2 | 3.0276e−8 | 1.7117e−32 | 4.1837 | 2.6747 |
| XW46(17–20), $T_0=-0.01,\ b=-2$ | 1.8398e−2 | 3.5032e−8 | 2.7935e−32 | 4.2039 | 2.9566 |
| O2M4 (2.4)–(2.5), $T_0=-0.01,\ \gamma=0$ | 1.8880e−2 | 2.3820e−7 | 1.9513e−30 | 4.7005 | 2.7019 |
| OM4 (2.4)–(2.6), $T_0=-0.01,\ \gamma=0$ | 1.8880e−2 | 3.3604e−8 | 2.6359e−32 | 4.1835 | 2.7236 |
| OM4 (2.4)–(2.7), $T_0=-0.01,\ \gamma=0$ | 1.8880e−2 | 3.8273e−8 | 4.0253e−32 | 4.2025 | 2.8162 |
| Example $f_2$, guess: 1.4 | | | | | |
| XW41(16–18), $T_0=-0.01,\ a=8$ | 1.8371e−5 | 1.3038e−22 | 7.0315e−101 | 4.5640 | 1.1329 |
| XW42(16–19), $T_0=-0.01,\ a=8$ | 1.8371e−5 | 1.5797e−22 | 5.6795e−107 | 4.3243 | 1.1474 |
| XW43(16–20), $T_0=-0.01,\ a=8$ | 1.8371e−5 | 6.7085e−25 | 1.5685e−108 | 4.3026 | 1.0778 |
| XW44(17–18), $T_0=-0.01,\ b=-2$ | 3.8040e−6 | 2.3630e−25 | 3.2074e−113 | 4.5748 | 1.2823 |
| XW45(17–19), $T_0=-0.01,\ b=-2$ | 3.8040e−6 | 4.3246e−27 | 9.8591e−118 | 4.3278 | 1.0925 |
| XW46(17–20), $T_0=-0.01,\ b=-2$ | 3.8040e−6 | 2.2214e−28 | 7.0328e−124 | 4.2953 | 1.1369 |
| OM4 (2.4)–(2.5), $T_0=-0.01,\ \gamma=0$ | 3.7144e−6 | 2.1871e−25 | 2.2845e−113 | 4.5752 | 1.0383 |
| OM4 (2.4)–(2.6), $T_0=-0.01,\ \gamma=0$ | 3.7144e−6 | 3.9924e−27 | 7.0907e−118 | 4.3279 | 1.1577 |
| OM4 (2.4)–(2.7), $T_0=-0.01,\ \gamma=0$ | 3.7144e−6 | 1.9614e−28 | 4.0581e−124 | 4.2951 | 1.0974 |
| Example $f_3$, guess: 1.3 | | | | | |
| XW41(16–18), $T_0=-0.01,\ a=8$ | 1.5319e−2 | 4.8667e−10 | 2.6217e−44 | 4.5743 | 1.2550 |
| XW42(16–19), $T_0=-0.01,\ a=8$ | 1.5319e−2 | 1.7723e−10 | 1.6516e−45 | 4.4174 | 1.2136 |
| XW43(16–20), $T_0=-0.01,\ a=8$ | 1.5319e−2 | 1.7723e−10 | 3.3034e−45 | 4.3794 | 1.1443 |
| XW44(17–18), $T_0=-0.01,\ b=-2$ | 7.1877e−4 | 7.3695e−16 | 1.0978e−70 | 4.5729 | 1.2125 |
| XW45(17–19), $T_0=-0.01,\ b=-2$ | 7.1877e−4 | 3.5313e−17 | 5.5819e−75 | 4.3430 | 1.0972 |
| XW46(17–20), $T_0=-0.01,\ b=-2$ | 7.1877e−4 | 3.5313e−17 | 1.1164e−74 | 4.3204 | 1.1482 |
| OM4 (2.4)–(2.5), $T_0=-0.01,\ \gamma=0$ | 7.1305e−4 | 7.3404e−16 | 1.0912e−70 | 4.5737 | 1.3175 |
| OM4 (2.4)–(2.6), $T_0=-0.01,\ \gamma=0$ | 7.1305e−4 | 3.3934e−17 | 4.6559e−75 | 4.3431 | 1.2563 |
| OM4 (2.4)–(2.7), $T_0=-0.01,\ \gamma=0$ | 7.1305e−4 | 3.3934e−17 | 9.3119e−75 | 4.3205 | 1.1916 |

**Table 3.** Numerical comparison of three-point with memory method

| Method | $\|x_1-\alpha\|$ | $\|x_2-\alpha\|$ | $\|x_3-\alpha\|$ | COC | CUP time |
|---|---|---|---|---|---|
| Example $f_1$, guess: $-1.6$ | | | | | |
| XW81(37–34), $T_0=1.5,\ L=0$ | 1.1801e−1 | 1.3693e−7 | 1.8971e−61 | 8.9496 | 2.6608 |
| XW82(37–35), $T_0=1.5,\ L=0$ | 1.1801e−1 | 1.4405e−9 | 3.5417e−87 | 9.7061 | 2.9674 |
| XW83(37–36), $T_0=1.5,\ L=0$ | 1.1801e−1 | 1.7309e−9 | 2.3974e−90 | 10.214 | 2.8367 |
| XW84(38–34), $T_0=2,\ K=6$ | 1.2272e−1 | 6.5242e−7 | 1.2627e−45 | 6.8131 | 2.5001 |
| XW85(38–35), $T_0=2,\ K=6$ | 3.5557e−2 | 3.6766e−15 | 3.5446e−140 | 9.6098 | 3.0186 |
| XW86(38–36), $T_0=2,\ K=6$ | 3.5557e−2 | 2.0055e−15 | 4.8248e−148 | 9.9921 | 3.0071 |
| OM81 (2.55)−(2.51), $T_0=1.5,\ \gamma=1$ | 1.9593e−2 | 4.0580e−15 | 2.5739e−129 | 8.9943 | 2.6720 |
| OM81 (2.55)−(2.53), $T_0=1.5,\ \gamma=1$ | 1.9593e−2 | 1.9159e−17 | 1.3449e−163 | 9.7289 | 2.8602 |
| OM81 (2.55)−(2.54), $T_0=1.5,\ \gamma=1$ | 1.9593e−2 | 7.4905e−18 | 4.5477e−171 | 9.9295 | 3.0666 |
| OM81 (2.55)−(2.52), $T_0=1.5,\ \gamma=1$ | 1.9593e−2 | 5.4549e−17 | 8.0689e−155 | 9.4610 | 2.8874 |
| Example $f_2$, guess: 2.3 | | | | | |
| XW81(37–34), $T_0=-1,\ L=1$ | HC | – | – | – | 1.3194 |
| XW82(37–35), $T_0=-1,\ L=1$ | HC | – | – | – | 1.1279 |
| XW83(37–36), $T_0=-1,\ L=1$ | HC | – | – | – | 1.1305 |
| XW84(38–34), $T_0=-1,\ K=0$ | 6.2961e−1 | 4.4250e−4 | 1.9236e−29 | 7.3534 | 1.1861 |
| XW85(38–35), $T_0=-1,\ K=0$ | 1.2396e−0 | 4.7897e−1 | 4.5123e−4 | 2.3441 | 1.1108 |
| XW86(38–36), $T_0=-1,\ K=0$ | 1.2396e−0 | 4.8136e−1 | 3.6709e−4 | 2.4270 | 1.1446 |
| OM81 (2.55)−(2.51), $T_0=-1.0,\ \gamma=1$ | 8.4611e−2 | 2.7477e−11 | 1.2500e−96 | 8.9573 | 1.4999 |
| OM81 (2.55)−(2.53), $T_0=-1.0,\ \gamma=1$ | 8.4611e−2 | 7.5983e−13 | 3.5136e−122 | 9.8625 | 1.1663 |
| OM81 (2.55)−(2.54), $T_0=-1.0,\ \gamma=1$ | 8.4611e−2 | 8.2840e−13 | 1.0967e−122 | 9.9451 | 1.1832 |
| OM81 (2.55)−(2.52), $T_0=-1.0,\ \gamma=1$ | 8.4611e−2 | 1.3930e−12 | 9.3226e−166 | 9.5331 | 1.0877 |
| Example $f_3$, guess: 1.3 | | | | | |
| XW81(37–34), $T_0=1.5,\ L=0$ | 2.4086e−2 | 2.0173e−17 | 3.8015e−153 | 9.0082 | 1.1468 |
| XW82(37–35), $T_0=1.5,\ L=0$ | 2.4086e−2 | 5.7528e−18 | 3.2909e−174 | 10.008 | 1.1619 |
| XW83(37–36), $T_0=1.5,\ L=0$ | 2.4086e−2 | 5.7528e−18 | 3.2909e−174 | 10.008 | 1.2208 |
| XW84(38–34), $T_0=2,\ K=6$ | HC | – | – | – | 1.1632 |
| XW85(38–35), $T_0=2,\ K=6$ | 3.3046e−1 | 2.4693e−3 | 7.1894e−27 | 11.901 | 1.2608 |
| XW86(38–36), $T_0=2,\ K=6$ | 3.3046e−1 | 2.4693e−3 | 7.1894e−27 | 11.901 | 1.1305 |
| OM81 (2.55)−(2.51), $T_0=1.5,\ \gamma=1$ | 2.3293e−7 | 1.3267e−62 | 8.3669e−560 | 9.0000 | 1.2703 |
| OM81 (2.55)−(2.53), $T_0=1.5,\ \gamma=1$ | 8.4611e−7 | 1.5593e−68 | 2.8183e−680 | 10.000 | 1.2083 |
| OM81 (2.55)−(2.54), $T_0=1.5,\ \gamma=1$ | 8.4611e−7 | 1.5593e−68 | 2.8183e−680 | 10.000 | 1.4507 |
| OM81 (2.55)−(2.52), $T_0=1.5,\ \gamma=1$ | 2.3293e−7 | 1.5593e−68 | 2.8183e−680 | 10.000 | 1.1088 |

## 4. CONCLUSIONS

In the present article, we have presented a new family of two-step and three-step iterative methods with memory for solving nonlinear equations. Since our aim is to construct the method of higher order convergence without additional calculation. So we have used three different approximations of self-correcting parameters, designed by Hermite interpolating polynomials in the fourth-order and eighth-order methods to achieve higher order convergence without any additional calculation. The $R$-order of convergence of the new with memory iterative methods is increased from 4 to 4.5616, 4.7913, and 5 and 8 to 9, 9.5846, 9.7958, and 10. The numerical results have been given to confirm the validity of theoretical results. We have also calculated the CPU time of the proposed and other existing methods.

## ACKNOWLEDGMENTS

The authors would like to pay their sincere gratitude to the reviewers and editor for careful reading and their valuable comments, which play a significant role in the improvement of the article.

## REFERENCES

1. Wang, X. and Zhang, T., Higher-Order Newton-Type Iterative Methods with and without Memory for Solving Nonlinear Equations, *Math. Comm.*, 2014, vol. 19, pp. 91–109.
2. Soleymani, F., Some Optimal Iterative Methods and Their with Memory Variants, *J. Egypt Math. Soc.*, 2013, vol. 21, pp. 133–141.
3. Kumar, S., Kanwar, V., Tomar, S.K., and Singh, S., Geometrically Constructed Families of Newton's Method for Unconstrained Optimization and Nonlinear Equations, *Int. J. Math. Math. Sci.*, 2011, vol. 2011; art. ID 972537.
4. Traub, J.F., *Iterative Methods for the Solution of Equations*, Englewood Cliffs, New Jersey: Prentice Hall, 1964.
5. Petkovic, M.S., Neta, B., Petcovic, L.D., and Dzunic, J., *Multipoint Methods for Solving Nonlinear Equations*, New York: Academic Press, 2013.
6. Behl, R. and Kanwar, V., New Highly Efficient Families of Higher-Order Methods for Simple Roots, Permitting $f'(x_n) = 0$, *Int. J. Math. Math. Sci.*, 2014, vol. 2014; art. ID 264529.
7. Ortega, J. and Rheinboldt, W., *Iterative Solution of Nonlinear Equations in Several Variables*, New York: Academic Press, 2000.
8. Alefeld, G. and Herzberger, J., *Introduction to Interval Computation*, New York: Academic Press, 1983.
9. Wang, X. and Zhang, T., A New Family of Newton-Type Iterative Methods with and without Memory for Solving Nonlinear Equations, *Calcolo*, 2014, vol. 51, pp. 1–15.
10. Wang, X. and Zhang, T., Some Newton-Type Iterative Methods with and without Memory for Solving Nonlinear Equations, *Int. J. Comp. Meth.*, 2013, vol. 11, pp. 1–20.
11. Chun, C. and Lee, M.Y., A New Optimal Eighth-Order Family of Iterative Methods for the Solution of Nonlinear Equations, *Appl. Math. Comput.*, 2013, vol. 223, pp. 506–519.
12. Weerakoon, S. and Fernando, T.G.I., A Variant of Newton's Method with Accelerated Third-Order Convergence, *Appl. Math. Lett.*, 2000, vol. 13, pp. 87–93.