MATHEMATICAL COMMUNICATIONS
Math. Commun. **19**(2014), 91–109

High-order Newton-type iterative methods with memory for solving nonlinear equations*

XIAOFENG WANG[1,†] AND TIE ZHANG[2]

[1] *School of Mathematics and Physics, Bohai University, Jinzhou 121 013, Liaoning, China*
[2] *College of Sciences, Northeastern University, Shenyang 110 819 Liaoning, China*

Received May 30, 2013; accepted December 17, 2013

Abstract. In this paper, we present a new family of two-step Newton-type iterative methods with memory for solving nonlinear equations. In order to obtain a Newton-type method with memory, we first present an optimal two-parameter fourth-order Newton-type method without memory. Then, based on the two-parameter method without memory, we present a new two-parameter Newton-type method with memory. Using two self-correcting parameters calculated by Hermite interpolatory polynomials, the R-order of convergence of a new Newton-type method with memory is increased from 4 to 5.7016 without any additional calculations. Numerical comparisons are made with some known methods by using the basins of attraction and through numerical computations to demonstrate the efficiency and the performance of the presented methods.
AMS subject classifications: 65H05, 65B99
Key words: Newton-type iterative method with memory, nonlinear equations, $R-$order convergence, root-finding

1. Introduction

Finding the root of a nonlinear equation $f(x) = 0$ is a classical problem in scientific computation. Recently, many iterative methods have been proposed for solving nonlinear equations, see [1-14] and the references therein. In these methods, the iterative methods with memory are worth studying. The iterative method with memory can improve the order of convergence of the method without memory without any additional calculations and it has a very high computational efficiency. There are two kinds of iterative methods with memory, i.e. Steffensen-type method and Newton-type method. In this paper, we only consider the Newton-type method with memory. For example, in [9] Petković et al. proposed a two-step Newton-type iterative method with memory of the R-order 4.562 by using inverse interpolation, which is written as

$$\begin{cases} N(x_n) = x_n - \dfrac{f(x_n)}{f'(x_n)}, \ \varphi(t) = \dfrac{1}{f(t)-f(x_n)}\left[\dfrac{t-x_n}{f(t)-f(x_n)} - \dfrac{1}{f'(x_n)}\right], \\ y_n = N(x_n) + f(x_n)^2\varphi(y_{n-1}), \\ x_{n+1} = N(x_n) + f(x_n)^2\varphi(y_n), \end{cases} \tag{1}$$

*For interpretation of color in all figures, the reader is referred to the web version of this article available at www.mathos.hr/mc.
†Corresponding author. *Email addresses:* `w200888w@163.com` (X. Wang), `ztmath@163.com` (T. Zhang)

where $y_{-1} = N(x_0)$. The basic idea of (1) comes from Neta who in [5] derived the following three-step Newton-type method with memory of the R-order 10.815:

$$\begin{cases} N(x_n) = x_n - \dfrac{f(x_n)}{f'(x_n)},\ \psi(t) = \dfrac{1}{f(t)-f(x_n)}\left[\dfrac{t-x_n}{f(t)-f(x_n)} - \dfrac{1}{f'(x_n)}\right], \\ y_n = N(x_n) + [f(y_{n-1})\psi(z_{n-1}) - f(z_{n-1})\psi(y_{n-1})]\dfrac{f(x_n)^2}{f(y_{n-1})-f(z_{n-1})}, \\ z_n = N(x_n) + [f(y_n)\psi(z_{n-1}) - f(z_{n-1})\psi(y_n)]\dfrac{f(x_n)^2}{f(y_n)-f(z_{n-1})}, \\ x_{n+1} = N(x_n) + [f(y_n)\psi(z_n) - f(z_n)\psi(y_n)]\dfrac{f(x_n)^2}{f(y_n)-f(z_n)}, \end{cases} \tag{2}$$

where $y_{-1} = N(x_0)$ and $z_{-1} = y_{-1} + |f(x_0)|/10$. The efficiency indices of methods (1) and (2) are low. The reason is that iterative methods (1) and (2) require four and six functional evaluations in their first iteration, respectively. In order to accelerate the convergence or improve the computational efficiency of the Newton-type method with memory, in [14] Wang and Zhang developed a two-step Newton-type method with memory of R-order 5, which is given by

$$\begin{cases} \lambda_n = -\dfrac{H_4''(x_n)}{2f'(x_n)}, \\ y_n = x_n - \dfrac{f(x_n)}{\lambda_n f(x_n) + f'(x_n)}, \\ x_{n+1} = y_n - \dfrac{f(y_n)}{2\lambda_n f(x_n) + f'(x_n)}\left(\dfrac{f(x_n) + (2+\beta)f(y_n)}{f(x_n) + \beta f(y_n)}\right), \end{cases} \tag{3}$$

where $\beta \in R$, $H_4''(x_n) = 2f[x_n, x_n, y_{n-1}] + (4f[x_n, x_n, y_{n-1}, x_{n-1}] - 2f[x_n, y_{n-1}, x_{n-1}, x_{n-1}])(x_n - y_{n-1})$ and $H_4(x) = H_4(x; x_n, x_n, y_{n-1}, x_{n-1}, x_{n-1})$ is a Hermite interpolatory polynomial of fourth degree. The efficiency index of method (3) is $5^{1/3} \approx 1.7100$, which is higher than the efficiency indices of methods (1) and (2), see [14].

The purpose of this paper is to improve further the R-order of convergence and the efficiency index of the two-step Newton-type iterative method and give the convergence analysis. This paper is organized as follows. In Section 2, we derive a family of optimal fourth-order iterative methods without memory for solving nonlinear equations. Further accelerations of convergence speed are attained in Section 3. We obtain a new family of two-point Newton-type iterative methods with memory by varying two free parameters in per full iteration. The two self-accelerating parameters are calculated using information available from the current and previous iterations. The corresponding R-order of convergence is increased from 4 to 5.7016. The maximal efficiency index of the new method with memory is $(5.7016)^{1/3} \approx 1.7865$, which is higher than the efficiency indices of the existing Newton-type methods. Numerical examples are given in Section 4 to illustrate convergence behavior of our methods for simple roots. In Section 5, some dynamical aspects associated to the presented methods are studied. Section 6 gives a short conclusion.

2. The two-step Newton-type method without memory

In order to get a two-step Newton-type iterative method with memory based on the well known Ostrowski's method [7, 10], we consider the following two-step iterative method without memory by using the weight function method.

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)} G(s_n), \\ x_{n+1} = y_n - \dfrac{f(y_n)}{2f[x_n, y_n] - f'(x_n)} H(s_n), \end{cases} \tag{4}$$

where $G(s_n)$ and $H(s_n)$ are two weight functions with $s_n = f(x_n)/f'(x_n)$. The functions $G(s_n)$ and $H(s_n)$ should be determined so that the iterative method (4) is of order four. For the iterative method defined by (4) we have the following result.

Theorem 1. *Let $a \in I$ be a simple zero of a sufficiently differentiable function $f : I \subset R \to R$ for an open interval I. Then the iterative method defined by (4) is of fourth-order convergence when*

$$G(0) = 1,\ G'(0) = \gamma,\ H(0) = 1,\ H'(0) = 0,\ H''(0) = 2\beta \text{ and } \gamma, \beta \in R, \tag{5}$$

and it satisfies the error equation below

$$e_{n+1} = (c_2 - \gamma)(c_2^2 - c_3 - \beta - c_2\gamma)e_n^4 + O(e_n^5). \tag{6}$$

Proof. Let $e_n = x_n - a$, $c_n = (1/n!)f^{(n)}(a)/f'(a)$, $n = 2, 3, \ldots$. Using the Taylor expansion and taking into account $f(a) = 0$ we have

$$f(x_n) = f'(a)[e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + O(e_n^6)], \tag{7}$$

$$f'(x_n) = f'(a)[1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + O(e_n^5)]. \tag{8}$$

Dividing (7) by (8) we obtain

$$s_n = \frac{f(x_n)}{f'(x_n)} = e_n - c_2 e_n^2 + (2c_2^2 - 2c_3)e_n^3 + (-4c_2^3 + 7c_2c_3 - 3c_4)e_n^4 + O(e_n^5). \tag{9}$$

Using (5), (9) and Taylor expansion $G(s_n) = G(0) + G'(0)s_n + O(s_n^2)$ we have

$$\begin{aligned} G(s_n) =& 1 + \gamma s_n \\ =& 1 + \gamma(e_n - c_2 e_n^2 + (2c_2^2 - 2c_3)e_n^3 + (-4c_2^3 + 7c_2c_3 - 3c_4)e_n^4 + O(e_n^5)). \end{aligned} \tag{10}$$

From (9) and (10) we get

$$\begin{aligned} e_{n,y} =& y_n - a = x_n - a - G(s_n)\frac{f(x_n)}{f'(x_n)} \\ =& (c_2 - \gamma)e_n^2 + 2(-c_2^2 + c_3 + c_2\gamma)e_n^3 \\ &+ (4c_2^3 - 7c_2c_3 + 3c_4 - 5c_2^2\gamma + 4c_3\gamma)e_n^4 + O(e_n^5). \end{aligned} \tag{11}$$

By an argument similar to that of (7), we have

$$\begin{aligned} f(y_n) =& f'(a)[(c_2-\gamma)e_n^2 - 2(c_2^2 - c_3 - c_2\gamma)e_n^3 \\ &+ (5c_2^3 - 7c_2^2\gamma + 4c_4\gamma + c_2(-7c_3 - \gamma^2))e_n^4 + O(e_n^5)]. \end{aligned} \tag{12}$$

From (7), (11) and (12) we obtain

$$\begin{aligned} f[x_n, y_n] =& f'(a)(1 + c_2e_n + (c_2^2 + c_3 - c_2\gamma)e_n^2 \\ &+ (-2c_2^3 + 3c_2c_3 + c_4 + 2c_2^2\gamma - c_3\gamma)e_n^3 + O(e_n^4)), \end{aligned} \tag{13}$$

$$\begin{aligned} f(y_n)(2f[x_n, y_n] - f'(x_n))^{-1} =& (c_2-\gamma)e_n^2 + 2(-c_2^2 + c_3 + c_2\gamma)e_n^3 \\ &+ (3c_2^3 - 3c_2^2\gamma + 3(c_4 + c_3\gamma) \\ &- c_2(6c_3 + \gamma^2))e_n^4 + O(e_n^5). \end{aligned} \tag{14}$$

Using (5), (9) and Taylor expansion $H(s_n) = H(0) + H'(0)s_n + H''(0)s_n^2/2 + O(s_n^3)$ we get

$$H(s_n) = 1 + \beta e_n^2 - 2\beta c_2 e_n^3 + \beta(5c_2^2 - 4c_3)e_n^4 + O(e_n^5). \tag{15}$$

Hence, together with (11), (14) and (15), we obtain the error equation

$$\begin{aligned} e_{n+1} =& x_{n+1} - a = y_n - a - f(y_n)H(s_n)(2f[x_n, y_n] - f'(x_n))^{-1} \\ =& (c_2-\gamma)(c_2^2 - c_3 - \beta - \gamma c_2)e_n^4 + O(e_n^5). \end{aligned} \tag{16}$$

The proof is completed. □

It is obvious that the iterative method (4) requires two functions and one derivative per iteration, thus it is an optimal scheme. Let $p^{1/n}$ be the efficiency index (see [7]), where p is the order of the method and n is the number of functional evaluations per iteration required by the method. We see that the new method (4) without memory has the efficiency index of $\sqrt[3]{4} \approx 1.587$.

Remark 1. *Based on Theorem 1, we obtain a new iterative method without memory as follows:*

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)} G(s_n), \\ x_{n+1} = y_n - \dfrac{f(y_n)}{2f[x_n, y_n] - f'(x_n)} H(s_n), \end{cases} \tag{17}$$

where $s_n = f(x_n)/f'(x_n)$. $G(s_n)$ and $H(s_n)$ in (17) are two weight functions satisfying condition (5). Functions $G(s_n)$ and $H(s_n)$ in (17) can take many forms. For example, taking the functions $G(s_n) = (1-\gamma s_n)^{-1}$ and $H(s_n) = (1-\beta s_n^2)^{-1}$, we can give the following fourth-order iterative method without memory:

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n) - \gamma f(x_n)}, \\ x_{n+1} = y_n - \dfrac{f(y_n)}{2f[x_n, y_n] - f'(x_n)} \dfrac{f'(x_n)^2}{f'(x_n)^2 - \beta f(x_n)^2}, \end{cases} \quad (\gamma,\ \beta \in R). \tag{18}$$

Taking the functions $G(s_n) = (1 - \gamma s_n)^{-1}$ *and* $H(s_n) = 1 + \beta s_n^2$*, we obtain the following fourth-order iterative method without memory:*

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n) - \gamma f(x_n)}, \\ x_{n+1} = y_n - \dfrac{f(y_n)}{2f[x_n, y_n] - f'(x_n)} \left(1 + \beta \left(\dfrac{f(x_n)}{f'(x_n)}\right)^2\right), \end{cases} \quad (\gamma,\ \beta \in R). \tag{19}$$

Remark 2. *It is noteworthy that the error equation (6) can be written as the following scheme*

$$\begin{aligned} e_{n+1} =& (c_2 - \gamma)(c_2^2 - c_3 - \beta - c_2\gamma)e_n^4 + O(e_n^5) \\ =& (c_2 - \gamma)[c_2(c_2 - \gamma) - (c_3 + \beta)]e_n^4 + O(e_n^5). \end{aligned} \tag{20}$$

We can improve the order of convergence of method (17) by taking parameters $\gamma = c_2$ *and* $\beta = -c_3$ *in (20). Therefore, the new method with memory can be obtained by method (17) without memory.*

3. New Newton-type iterative methods with memory

In this section we will improve the convergence rate of method (17) by varying the parameters γ and β in per full iteration. Taking $\gamma = c_2$ and $\beta = -c_3$ in (20), we can improve the order of convergence of method (17). However, the exact values of $f'(a)$, $f''(a)$ and $f'''(a)$ are not available in practice and such acceleration of convergence cannot be realized. But we could approximate the parameter γ by γ_n and approximate the parameter β by β_n. Parameters γ_n and β_n can be computed by using information available from the current and previous iterations and satisfy

$$\lim_{n\to\infty} \gamma_n = c_2 = \frac{f''(a)}{2f'(a)} \quad \text{and} \quad \lim_{n\to\infty} \beta_n = -c_3 = -\frac{f'''(a)}{6f'(a)}$$

such that the fourth-order asymptotic convergence constant is zero in (20). In this paper, the self-accelerating parameter γ_n is equal to the parameter $-\lambda_n$ of Newton-type method (3). We consider the following four methods for β_n:

Method 1:

$$\gamma_n = \frac{H_4''(x_n)}{2f'(x_n)} \quad \text{and} \quad \beta_n = -\frac{H_4'''(x_n)}{6f'(x_n)}, \tag{21}$$

where

$$H_4'''(x_n) = 6f[x_n, x_n, y_{n-1}, x_{n-1}] + 6f[x_n, x_n, y_{n-1}, x_{n-1}, x_{n-1}](2x_n - x_{n-1} - y_{n-1}).$$

Method 2:

$$\gamma_n = \frac{H_4''(x_n)}{2f'(x_n)} \quad \text{and} \quad \beta_n = -\frac{\bar{H}_3'''(y_n)}{6f'(x_n)}, \tag{22}$$

where $\bar{H}_3'''(y_n) = 6f[y_n, x_n, x_n, y_{n-1}]$. $\bar{H}_3(x) = \bar{H}_3(x\ ; y_n,\ x_n, x_n, y_{n-1})$ is a Hermite interpolatory polynomial of third degree.

Method 3:

$$\gamma_n = \frac{H_4''(x_n)}{2f'(x_n)} \quad \text{and} \quad \beta_n = -\frac{\bar{H}_4'''(y_n)}{6f'(x_n)}, \tag{23}$$

where

$$\bar{H}_4'''(y_n) = \bar{H}_3'''(y_n) + 6f[y_n, x_n, x_n, y_{n-1}, x_{n-1}](3y_n - y_{n-1} - 2x_n).$$

$\bar{H}_4(x) = \bar{H}_4(x; y_n, x_n, x_n, y_{n-1}, x_{n-1})$ is a Hermite interpolatory polynomial of fourth degree.

Method 4:

$$\gamma_n = \frac{H_4''(x_n)}{2f'(x_n)} \quad \text{and} \quad \beta_n = -\frac{\bar{H}_5'''(y_n)}{6f'(x_n)}, \tag{24}$$

where

$$\begin{aligned}\bar{H}_5'''(y_n) =& \bar{H}_4'''(y_n) + 6f[y_n, x_n, x_n, y_{n-1}, x_{n-1}, x_{n-1}] \\ & [(y_n - y_{n-1})(y_n - x_{n-1}) + (y_n - x_n)^2 + 2(y_n - x_n)(2y_n - y_{n-1} - x_{n-1})].\end{aligned}$$

$\bar{H}_5(x) = \bar{H}_5(x; y_n, x_n, x_n, y_{n-1}, x_{n-1}, x_{n-1})$ is a Hermite interpolatory polynomial of fifth degree.

Here $f[x_n, x_n] = f'(x_n)$ and $f[x_n, t] = (f(t) - f(x_n))/(t - x_n)$ are two first-order divided differences. The higher order divided differences are defined recursively. The divided difference $f[y_n, x_n, x_n, t_0, t_1, \ldots, t_{m-3}]$ of order m ($m \geq 3$) is defined as

$$f[y_n, x_n, x_n, t_0, t_1, \ldots, t_{m-3}] = \frac{f[x_n, x_n, t_0, t_1, \ldots, t_{m-3}] - f[y_n, x_n, x_n, t_0, t_1, \ldots t_{m-4}]}{t_{m-3} - y_n}.$$

The divided difference $f[t_0, t_1, \ldots, t_{m-3}]$ of order $m(m \geq 3)$ is defined as

$$f[t_0, t_1, \ldots, t_{m-3}] = \frac{f[t_1, t_2, \ldots, t_{m-3}] - f[t_0, t_1, \ldots t_{m-4}]}{t_{m-3} - t_0}.$$

Remark 3. *Now, we can obtain the following two-parameter Newton-type iterative method with memory*

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)} G(s_n), \\ x_{n+1} = y_n - \dfrac{f(y_n)}{2f[x_n, y_n] - f'(x_n)} H(s_n), \end{cases} \tag{25}$$

where $s_n = f(x_n)/f'(x_n), G(s_n)$ *satisfies the condition* $G(0) = 1,\ G'(0) = \gamma_n$ *and* $H(s_n)$ *satisfies the condition* $H(0) = 1,\ H'(0) = 0,\ H''(0) = 2\beta_n$*. The parameters* γ_n *and* β_n *are calculated by using one of the formulas (21)-(24) and they depend on the data available from the current and the previous iterations. In this paper, the self-accelerating parameter* γ_n *is equal to the parameter* $-\lambda_n$ *of Newton-type method (3).*

Remark 4. *Parameter (21) can be computed easily, because Hermite interpolation polynomial $H_4(x)$ has been used in the parameter γ_n. Computation of self-correcting parameters γ_n and β_n is complicated in this paper. The main purpose of this study is to choose available nodes as good as possible to obtain the maximal convergence order of two-step Newton type methods with memory. Hence, some simple approximations of self-accelerating parameters γ_n and β_n are not considered in this paper.*

The concept of the R-order of convergence [6] and the following assertion (see [1]) will be applied to estimate the convergence order of the iterative method with memory (25).

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

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

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

Lemma 1. *Let $\bar{H}_m$ be the Hermite interpolating polynomial of the degree m that interpolates a function f at interpolation nodes $y_n, x_n, t_0, \ldots, t_{m-3}$ contained in an interval I and the derivative $f^{(m+1)}$ is continuous in I and the Hermite interpolating polynomial $\bar{H}_m(x)$ satisfied the condition $\bar{H}_m(y_n) = f(y_n), \bar{H}_m(x_n) = f(x_n), \bar{H}'_m(x_n) = f'(x_n), \bar{H}_m(t_j) = f(t_j) (j = 0, \ldots, m-3)$. Define the errors $e_{t,j} = t_j - a$ $(j = 0, \ldots, m-3)$ and assume that*

1. *all nodes $y_n, x_n, t_0, \ldots, t_{m-3}$ are sufficiently close to the zero a;*
2. *the conditions $e_n = x_n - a = O(e_{t,0} \ldots e_{t,m-3})$ and $e_{n,y} = y_n - a = O(e_n^2 e_{t,0} \ldots e_{t,m-3})$ hold.*

Then

$$\beta_n + c_3 \sim (-1)^{m-2} c_{m+1} \prod_{j=0}^{m-3} e_{t,j}. \tag{26}$$

Proof. The error of the Hermite interpolation can be expressed as follows

$$f(x) - \bar{H}_m(x) = \frac{f^{(m+1)}(\xi)}{(m+1)!} (x - y_n)(x - x_n)^2 \prod_{j=0}^{m-3} (x - t_j) \ (\xi \in I). \tag{27}$$

Differentiating (27) at the point $x = y_n$ we obtain

$$f'''(y_n) - \bar{H}'''_m(y_n) = 6 \frac{f^{(m+1)}(\xi)}{(m+1)!} \left\{ \prod_{j=0}^{m-3} (y_n - t_j) + 2(y_n - x_n) \sum_{k=0}^{m-3} \left(\prod_{\substack{j=0 \\ j \neq k}}^{m-3} (y_n - t_j) \right) \right.$$

$$+\frac{(y_n-x_n)^2}{2}\sum_{k=0}^{m-3}\sum_{\substack{i=0\\ i\neq k}}^{m-3}\left(\prod_{\substack{j=0\\ j\neq k\\ j\neq i}}^{m-3}(y_n-t_j)\right)\Bigg\}\ (\xi\in I), \tag{28}$$

$$\bar{H}_m'''(y_n)=f'''(y_n)-6\frac{f^{(m+1)}(\xi)}{(m+1)!}\left\{\prod_{j=0}^{m-3}(y_n-t_j)+2(y_n-x_n)\sum_{k=0}^{m-3}\left(\prod_{\substack{j=0\\ j\neq k}}^{m-3}(y_n-t_j)\right)\right.$$

$$\left.+\frac{(y_n-x_n)^2}{2}\sum_{k=0}^{m-3}\sum_{\substack{i=0\\ i\neq k}}^{m-3}\left(\prod_{\substack{j=0\\ j\neq k\\ j\neq i}}^{m-3}(y_n-t_j)\right)\right\}\quad(\xi\in I), \tag{29}$$

Taylor's series of derivatives of f at the point $x_n, y_n\in I$ and $\xi\in I$ about the zero a of f give

$$f'(x_n)=f'(a)(1+2c_2e_n+3c_3e_n^2+4c_4e_n^3+O(e_n^4)), \tag{30}$$
$$f'(y_n)=f'(a)(1+2c_2e_{n,y}+3c_3e_{n,y}^2+4c_4e_{n,y}^3+O(e_{n,y}^4)), \tag{31}$$
$$f''(y_n)=f'(a)(2c_2+6c_3e_{n,y}+12c_4e_{n,y}^2+O(e_{n,y}^3)), \tag{32}$$
$$f'''(y_n)=f'(a)(6c_3+24c_4e_{n,y}+O(e_{n,y}^2)), \tag{33}$$
$$f^{(m+1)}(\xi)=f'(a)((m+1)!c_{m+1}+(m+2)!c_{m+2}e_\xi+O(e_\xi^2)), \tag{34}$$

where $e_\xi=\xi-a$. Substituting (33) and (34) into (29) we have

$$\bar{H}_m'''(y_n)\sim 6f'(a)\left\{c_3+4c_4e_{n,y}-c_{m+1}\left\{\prod_{j=0}^{m-3}(y_n-t_j)+2(y_n-x_n)\right.\right.$$

$$\left.\left.\sum_{k=0}^{m-3}\left(\prod_{\substack{j=0\\ j\neq k}}^{m-3}(y_n-t_j)\right)+\frac{(y_n-x_n)^2}{2}\sum_{k=0}^{m-3}\sum_{\substack{i=0\\ i\neq k}}^{m-3}\left(\prod_{\substack{j=0\\ j\neq k\\ j\neq i}}^{m-3}(y_n-t_j)\right)\right\}\right\}. \tag{35}$$

Dividing (35) by (30) we obtain

$$\beta_n = -\frac{\bar{H}_m'''(y_n)}{6f'(x_n)} \sim -\left(c_3 - c_{m+1}\left\{\prod_{j=0}^{m-3}(y_n - t_j) + 2(y_n - x_n)\right.\right.$$

$$\left.\left.\sum_{k=0}^{m-3}\left(\prod_{\substack{j=0\\ j\neq k}}^{m-3}(y_n - t_j)\right) + \frac{(y_n - x_n)^2}{2}\sum_{k=0}^{m-3}\sum_{\substack{i=0\\ i\neq k}}^{m-3}\left(\prod_{\substack{j=0\\ j\neq k\\ j\neq i}}^{m-3}(y_n - t_j)\right)\right\}\right), \quad (36)$$

and

$$\beta_n + c_3 \sim c_{m+1}\left\{\prod_{j=0}^{m-3}(y_n - t_j) + 2(y_n - x_n)\sum_{k=0}^{m-3}\left(\prod_{\substack{j=0\\ j\neq k}}^{m-3}(y_n - t_j)\right)\right.$$

$$\left. + \frac{(y_n - x_n)^2}{2}\sum_{k=0}^{m-3}\sum_{\substack{i=0\\ i\neq k}}^{m-3}\left(\prod_{\substack{j=0\\ j\neq k\\ j\neq i}}^{m-3}(y_n - t_j)\right)\right\} \sim (-1)^{m-2}c_{m+1}\prod_{j=0}^{m-3}e_{t,j}. \quad (37)$$

The proof is completed. □

Theorem 3. *Let the varying parameters γ_n and β_n in the iterative method (25) be calculated by (21). If an initial approximation x_0 is sufficiently close to a simple root a of $f(x)$, then the R-order of convergence of the iterative method (25) with memory is at least 5.3059.*

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

$$e_{n+1} \sim D_{n,r}e_n^r, \; e_n = x_n - a, \quad (38)$$

where $D_{n,r}$ tends to the asymptotic error constant D_r of (IM) when $n \to \infty$. So,

$$e_{n+1} \sim D_{n,r}(D_{n-1,r}e_{n-1}^r)^r = D_{n,r}D_{n-1,r}^r e_{n-1}^{r^2}. \quad (39)$$

It is similar to the derivation of (39). We assume that the iterative sequence $\{y_n\}$ has the R-order p; then

$$e_{n,y} \sim D_{n,p}e_n^p \sim D_{n,p}(D_{n-1,r}e_{n-1}^r)^p = D_{n,p}D_{n-1,r}^p e_{n-1}^{rp}. \quad (40)$$

Using (11), (20), γ_n and β_n, we obtain the corresponding error relations for the methods with memory (25)

$$e_{n,y} = y_n - a \sim (c_2 - \gamma_n)e_n^2, \tag{41}$$

$$e_{n+1} = x_{n+1} - a \sim (c_2 - \gamma_n)[c_2(c_2 - \gamma_n) - (c_3 + \beta_n)]e_n^4. \tag{42}$$

Here, we omitted higher order terms in (41)-(42).

Hermite interpolating polynomial $H_4(x)$ satisfied the conditions $H_4(x_n) = f(x_n)$, $H_4'(x_n) = f'(x_n)$, $H_4(y_{n-1}) = f(y_{n-1})$, $H_4(x_{n-1}) = f(x_{n-1})$ and $H_4'(x_{n-1}) = f'(x_{n-1})$. The error of the Hermite interpolation can be expressed as follows:

$$f(x) - H_4(x) = \frac{f^{(5)}(\xi)}{5!}(x - x_n)^2(x - x_{n-1})^2(x - y_{n-1}) \quad (\xi \in I). \tag{43}$$

Differentiating (43) at the point $x = x_n$ we obtain

$$f''(x_n) - H_4''(x_n) = 2\frac{f^{(5)}(\xi)}{5!}(x_n - x_{n-1})^2(x_n - y_{n-1}) \quad (\xi \in I), \tag{44}$$

$$H_4''(x_n) = f''(x_n) - 2\frac{f^{(5)}(\xi)}{5!}(x_n - x_{n-1})^2(x_n - y_{n-1}) \quad (\xi \in I), \tag{45}$$

$$f'''(x_n) - H_4'''(x_n) = 6\frac{f^{(5)}(\xi)}{5!}[2(x_n - x_{n-1})(x_n - y_{n-1}) + (x_n - x_{n-1})^2] \quad (\xi \in I), \tag{46}$$

$$\begin{aligned} H_4'''(x_n) =& f'''(x_n) - 6\frac{f^{(5)}(\xi)}{5!}[2(x_n - x_{n-1})(x_n - y_{n-1}) \\ &+ (x_n - x_{n-1})^2] \quad (\xi \in I). \end{aligned} \tag{47}$$

Taylor's series of derivatives of f at the point $x_n, \xi \in I$ about the zero a of f give

$$f''(x_n) = f'(a)(2c_2 + 6c_3e_n + 12c_4e_n^2 + O(e_n^3)), \tag{48}$$

$$f'''(x_n) = f'(a)(6c_3 + 24c_4e_n + O(e_n^2)), \tag{49}$$

$$f^{(5)}(\xi) = f'(a)(5!c_5 + 6!c_6e_\xi + O(e_\xi^2)), \tag{50}$$

where $e_\xi = \xi - a$.

Substituting (48) and (50) into (45) we have

$$H_4''(x_n) \sim 2f'(a)(c_2 + c_5e_{n-1,y}e_{n-1}^2 + 3c_3e_n). \tag{51}$$

Substituting (49) and (50) into (47) we have

$$H_4'''(x_n) \sim 6f'(a)(c_3 - c_5(2e_{n-1,y}e_{n-1} + e_{n-1}^2) + 4c_4e_n). \tag{52}$$

Dividing (51) by (30) we obtain

$$\gamma_n = \frac{H_4''(x_n)}{2f'(x_n)} \sim \left(c_2 + c_5e_{n-1,y}e_{n-1}^2 + (3c_3 - 2c_2^2)e_n\right), \tag{53}$$

$$c_2 - \gamma_n \sim -c_5e_{n-1,y}e_{n-1}^2. \tag{54}$$

Dividing (52) by (30) we obtain

$$\beta_n = -\frac{H_4'''(x_n)}{6f'(x_n)} \sim -\big(c_3 - c_5(2e_{n-1,y}e_{n-1} + e_{n-1}^2) + (4c_4 - c_2c_3)e_n\big)\,, \tag{55}$$

$$c_3 + \beta_n \sim \big(c_5(2e_{n-1,y}e_{n-1} + e_{n-1}^2) - (4c_4 - c_2c_3)e_n\big) \sim c_5 e_{n-1}^2. \tag{56}$$

According to (41), (42), (54) and (56), we get

$$\begin{aligned} e_{n,y} \sim&(c_2 - \gamma_n)e_n^2 \sim -c_5 e_{n-1,y}e_{n-1}^2(D_{n-1,r}e_{n-1}^r)^2 \\ \sim& - c_5 D_{n-1,p}D_{n-1,r}^2 e_{n-1}^{2r+p+2}, \end{aligned} \tag{57}$$

$$\begin{aligned} e_{n+1} \sim& c_5 e_{n-1,y}e_{n-1}^2[c_2c_5 e_{n-1,y}e_{n-1}^2 + c_5 e_{n-1}^2]e_n^4 \\ \sim& c_5^2 e_{n-1}^4 D_{n-1,p}e_{n-1}^p(D_{n-1,r}e_{n-1}^r)^4 \sim c_5^2 D_{n-1,p}D_{n-1,r}^4 e_{n-1}^{4r+p+4}. \end{aligned} \tag{58}$$

By comparing exponents of e_{n-1} appearing in two pairs of relations ((40),(57)) and ((39),(58)), we get the following system of equations

$$\begin{cases} 2r + p + 2 = rp, \\ 4r + p + 4 = r^2. \end{cases} \tag{59}$$

Positive solution of system (59) is given by $r \approx 5.3059$ and $p \approx 2.9209$. Therefore, the R-order of the methods with memory (25), when γ_n is calculated by (21), is at least 5.3059. □

Using the results of Lemma 1 and Theorem 3, we get Theorem 4 as follows:

Theorem 4. *Let the varying parameters γ_n and β_n in the iterative method (25) be calculated by (22), (23) and (24), respectively. If an initial approximation x_0 is sufficiently close to a simple root a of $f(x)$, then the R-order of convergence of iterative methods (25) with memory is at least* 5.4356, 5.5708 *and* 5.7016, *respectively.*

Proof. Method 2, γ_n and β_n are calculated by (22):
Using Lemma 1 for $m = 3$ and $t_0 = y_{n-1}$, we obtain

$$c_3 + \beta_n \sim -c_4 e_{n-1,y}. \tag{60}$$

According to (42), (54) and (60), we get

$$\begin{aligned} e_{n+1} \sim& c_5 e_{n-1}^2 e_{n-1,y}[c_2c_5 e_{n-1}^2 e_{n-1,y} - c_4 e_{n-1,y}]e_n^4 \sim -c_4c_5 e_{n-1}^2 e_{n-1,y}^2 e_n^4 \\ \sim& - c_4c_5 e_{n-1}^2 D_{n-1,p}^2 e_{n-1}^{2p}(D_{n-1,r}e_{n-1}^r)^4 \sim -c_4c_5 D_{n-1,p}^2 D_{n-1,r}^4 e_{n-1}^{4r+2p+2}. \end{aligned} \tag{61}$$

By comparing exponents of e_{n-1} appearing in two pairs of relations ((40),(57)) and ((39),(61)), we get the following system of equations

$$\begin{cases} 2r + p + 2 = rp, \\ 4r + 2p + 2 = r^2. \end{cases} \tag{62}$$

Positive solution of system (62) is given by $r \approx 5.4356$ and $p \approx 2.9018$. Therefore, the R-order of the methods with memory (25), when γ_n is calculated by (22), is at least 5.4356.

Method 3, γ_n and β_n are calculated by (23):
Using Lemma 1 for $m=4,\ t_0=y_{n-1}$ and $t_1=x_{n-1}$, we obtain

$$c_3+\beta_n\sim c_5e_{n-1,y}e_{n-1}, \tag{63}$$

According to (42), (54) and (63), we get

$$\begin{aligned}e_{n+1}\sim&c_5e_{n-1}^2e_{n-1,y}[c_2c_5e_{n-1}^2e_{n-1,y}+c_5e_{n-1,y}e_{n-1}]e_n^4\sim c_5^2e_{n-1}^3e_{n-1,y}^2e_n^4\\ \sim&c_5^2e_{n-1}^3D_{n-1,p}^2e_{n-1}^{2p}(D_{n-1,r}e_{n-1}^r)^4\sim c_5^2D_{n-1,p}^2D_{n-1,r}^4e_{n-1}^{4r+2p+3}.\end{aligned} \tag{64}$$

By comparing exponents of e_{n-1} appearing in two pairs of relations ((40), (57)) and ((39), (64)), we get the following system of equations

$$\begin{cases}2r+p+2=rp,\\ 4r+2p+3=r^2.\end{cases} \tag{65}$$

Positive solution of system (65) is given by $r\approx 5.5708$ and $p\approx 2.8751$. Therefore, the R-order of the methods with memory (25), when γ_n is calculated by (23), is at least 5.5708.

Method 4, γ_n and β_n are calculated by (24):
Hermite interpolating polynomial $\bar{H}_5(x)$ satisfied the condition $\bar{H}_5(y_n)=f(y_n)$, $\bar{H}_5(x_n)=f(x_n),\ \bar{H}'_5(x_n)=f'(x_n),\ \bar{H}_5(y_{n-1})=f(y_{n-1}),\ \bar{H}_5(x_{n-1})=f(x_{n-1})$ and $\bar{H}'_5(x_{n-1})=f'(x_{n-1})$. The error of the Hermite interpolation can be expressed as follows:

$$f(x)-\bar{H}_5(x)=\frac{f^{(5)}(\xi)}{6!}(x-y_n)(x-x_n)^2(x-x_{n-1})^2(x-y_{n-1})\qquad(\xi\in I). \tag{66}$$

Differentiating (66) at the point $x=y_n$ we obtain

$$\begin{aligned}f'''(y_n)-\bar{H}_5'''(y_n)=&6\frac{f^{(5)}(\xi)}{6!}[(y_n-y_{n-1})(y_n-x_{n-1})^2+2(y_n-x_n)(y_n-x_{n-1})^2\\ &+4(y_n-x_n)(y_n-x_{n-1})(y_n-y_{n-1})+2(y_n-x_n)^2(y_n-x_{n-1})\\ &+(y_n-x_n)^2(y_n-y_{n-1})]\quad(\xi\in I),\end{aligned} \tag{67}$$

$$\begin{aligned}\bar{H}_5'''(y_n)=&f'''(y_n)-6\frac{f^{(5)}(\xi)}{6!}[(y_n-y_{n-1})(y_n-x_{n-1})^2\\ &+2(y_n-x_n)(y_n-x_{n-1})^2+4(y_n-x_n)(y_n-x_{n-1})(y_n-y_{n-1})\\ &+2(y_n-x_n)^2(y_n-x_{n-1})\\ &+(y_n-x_n)^2(y_n-y_{n-1})]\quad(\xi\in I).\end{aligned} \tag{68}$$

Taylor's series of derivatives of f at the point $\xi\in I$ about the zero a of f give

$$f^{(5)}(\xi)=f'(a)(6!c_6+7!c_7e_\xi+O(e_\xi^2)), \tag{69}$$

where $e_\xi=\xi-a$.
Substituting (33) and (69) into (68) we have

$$\bar{H}_5'''(y_n)\sim 6f'(a)(c_3+c_6e_{n-1,y}e_{n-1}^2). \tag{70}$$

Dividing (70) by (30) we obtain

$$\beta_n = -\frac{\bar{H}_5'''(y_n)}{6f'(x_n)} \sim -\left(c_3 + c_6 e_{n-1,y} e_{n-1}^2\right), \tag{71}$$

$$c_3 + \beta_n \sim -\, c_6 e_{n-1,y} e_{n-1}^2. \tag{72}$$

According to (42), (54) and (72), we get

$$\begin{aligned} e_{n+1} \sim & c_5 e_{n-1,y} e_{n-1}^2 [c_2 c_5 e_{n-1,y} e_{n-1}^2 - c_6 e_{n-1,y} e_{n-1}^2] e_n^4 \sim c_5(c_2 c_5 - c_6) e_{n-1}^4 e_{n-1,y}^2 e_n^4 \\ \sim & c_5(c_2 c_5 - c_6) D_{n-1,p}^2 D_{n-1,r}^4 e_{n-1}^{4r+2p+4}. \end{aligned} \tag{73}$$

By comparing exponents of e_{n-1} appearing in two pairs of relations ((40),(57)) and ((39),(73)), we get the following system of equations

$$\begin{cases} 2r + p + 2 = rp, \\ 4r + 2p + 4 = r^2. \end{cases} \tag{74}$$

Positive solution of system (74) is given by $r \approx 5.7016$ and $p \approx 2.8508$. Therefore, the R-order of the methods with memory (25), when γ_n is calculated by (24), is at least 5.7016.

The proof is completed. □

Remark 5. *The efficiency index of iterative method (25) with the corresponding expressions (21)-(24) of* γ_n *and* β_n *is* $(5.3059)^{1/3} \approx 1.7442$, $(5.4356)^{1/3} \approx 1.7583$, $(5.5708)^{1/3} \approx 1.7727$ *and* $(5.7016)^{1/3} \approx 1.7865$, *respectively. The efficiency indices of our methods with memory are higher than the efficiency index* $(5^{1/3} \approx 1.7100)$ *of Newton-type method (3) with memory.*

4. Numerical results

The new methods (18) and (19) with and without memory are used to solve nonlinear functions $f_i(x) (i = 1, 2)$ and the computation results are compared with other Newton-type iterative methods (1) and (3) with memory, see Tables 1-2. The absolute errors$|x_k - a|$ in the first four iterations are given in Tables 1-2, where a is the exact root computed with 1200 significant digits. The computational order of convergence ρ is defined by [3]:

$$\rho \approx \frac{\ln(|x_{n+1} - x_n| \,/\, |x_n - x_{n-1}|)}{\ln(|x_n - x_{n-1}| \,/\, |x_{n-1} - x_{n-2}|)}. \tag{75}$$

The following test functions are used:

$$f_1(x) = xe^{x^2} - \sin^2(x) + 3\cos(x) + 5, \; a \approx -1.2076478271309189, x_0 = -1.3,$$
$$f_2(x) = \arcsin(x^2 - 1) - 0.5x + 1, \; a \approx 0.59481096839836918, x_0 = 0.7.$$

The numerical results shown in Tables 1-2 are in concordance with the theory developed in this paper. We can notice that the results obtained by our methods with memory are better than the other two-step Newton-type methods with memory. The highest order of convergence of our methods with memory is 5.7016, which is higher than methods (1) and (3).

Methods	$\lvert x_1-a\rvert$	$\lvert x_2-a\rvert$	$\lvert x_3-a\rvert$	$\lvert x_4-a\rvert$	ρ
$(18), \gamma=\beta=-1$	0.23e-4	0.37e-19	0.23e-78	0.40e-315	4.00
$(19), \gamma=\beta=-1$	0.23e-4	0.36e-19	0.20e-78	0.22e-315	4.00
(1)	0.53e-5	0.12e-23	0.25e-108	0.18e-494	4.56
$(3), \lambda_0=-0.01, \beta=-2$	0.34e-4	0.66e-21	0.32e-105	0.90e-527	5.00
$(18),(21), \gamma_0=\beta_0=-1$	0.23e-4	0.85e-24	0.47e-127	0.62e-675	5.31
$(18),(22), \gamma_0=\beta_0=-1$	0.23e-4	0.45e-24	0.45e-132	0.89e-719	5.43
$(18),(23), \gamma_0=\beta_0=-1$	0.23e-4	0.34e-25	0.32e-141	0.13e-787	5.57
$(18),(24), \gamma_0=\beta_0=-1$	0.23e-4	0.11e-25	0.53e-148	0.24e-844	5.69
$(19),(21), \gamma_0=\beta_0=-1$	0.23e-4	0.82e-24	0.38e-127	0.20e-675	5.31
$(19),(22), \gamma_0=\beta_0=-1$	0.23e-4	0.43e-24	0.36e-132	0.28e-719	5.43
$(19),(23), \gamma_0=\beta_0=-1$	0.23e-4	0.32e-25	0.26e-141	0.41e-788	5.57
$(19),(24), \gamma_0=\beta_0=-1$	0.23e-4	0.10e-25	0.40e-148	0.47e-845	5.69

Table 1: Numerical results for $f_1(x)$ by the methods with and without memory

Methods	$\lvert x_1-a\rvert$	$\lvert x_2-a\rvert$	$\lvert x_3-a\rvert$	$\lvert x_4-a\rvert$	ρ
$(18), \gamma=\beta=-0.6$	0.56e-4	0.48e-17	0.27e-69	0.28e-278	4.00
$(19), \gamma=\beta=-0.6$	0.56e-4	0.49e-17	0.30e-69	0.42e-278	4.00
(1)	0.37e-8	0.70e-41	0.13e-189	0.84e-868	4.56
$(3), \lambda_0=-0.1, \beta=-1.75$	0.40e-5	0.25e-27	0.96e-139	0.88e-696	5.00
$(18),(21), \gamma_0=\beta_0=-0.6$	0.56e-4	0.65e-24	0.33e-128	0.77e-683	5.32
$(18),(22), \gamma_0=\beta_0=-0.6$	0.56e-4	0.41e-24	0.16e-133	0.13e-727	5.43
$(18),(23), \gamma_0=\beta_0=-0.6$	0.56e-4	0.55e-25	0.28e-141	0.37e-789	5.57
$(18),(24), \gamma_0=\beta_0=-0.6$	0.56e-4	0.62e-26	0.15e-149	0.26e-855	5.71
$(19),(21), \gamma_0=\beta_0=-0.6$	0.56e-4	0.66e-24	0.38e-128	0.17e-682	5.32
$(19),(22), \gamma_0=\beta_0=-0.6$	0.56e-4	0.42e-24	0.19e-133	0.31e-727	5.43
$(19),(23), \gamma_0=\beta_0=-0.6$	0.56e-4	0.57e-25	0.33e-141	0.92e-789	5.57
$(19),(24), \gamma_0=\beta_0=-0.6$	0.56e-4	0.64e-26	0.19e-149	0.85e-855	5.71

Table 2: Numerical results for $f_2(x)$ by the methods with and without memory

5. Dynamical analysis

Dynamical properties of the rational function give us important information about numerical features of the iterative method as its stability and reliability. In what follows, we compare our methods with and without memory to methods with memory (1) and (3) by using the basins of attraction for three complex polynomials $f(z) = z^k - 1, k = 2, 3, 4$. We use a similar like technique as in [15] to generate the basins of attraction. To generate the basins of attraction for the zeros of a polynomial and an iterative method we take a grid of 400×400 points in a rectangle $D = [-3.0, 3.0] \times [-3.0, 3.0] \subset C$ and we use these points as z_0. If the sequence generated by the iterative method reaches a zero z^* of the polynomial with a tolerance $|z_k - z^*| < 10^{-15}$ and a maximum of 25 iterations, we decide that z_0 is in the basin of attraction of the zero and we paint this point blue for this root. In the same basin of attraction, the number of iterations needed to achieve the solution is shown in darker or

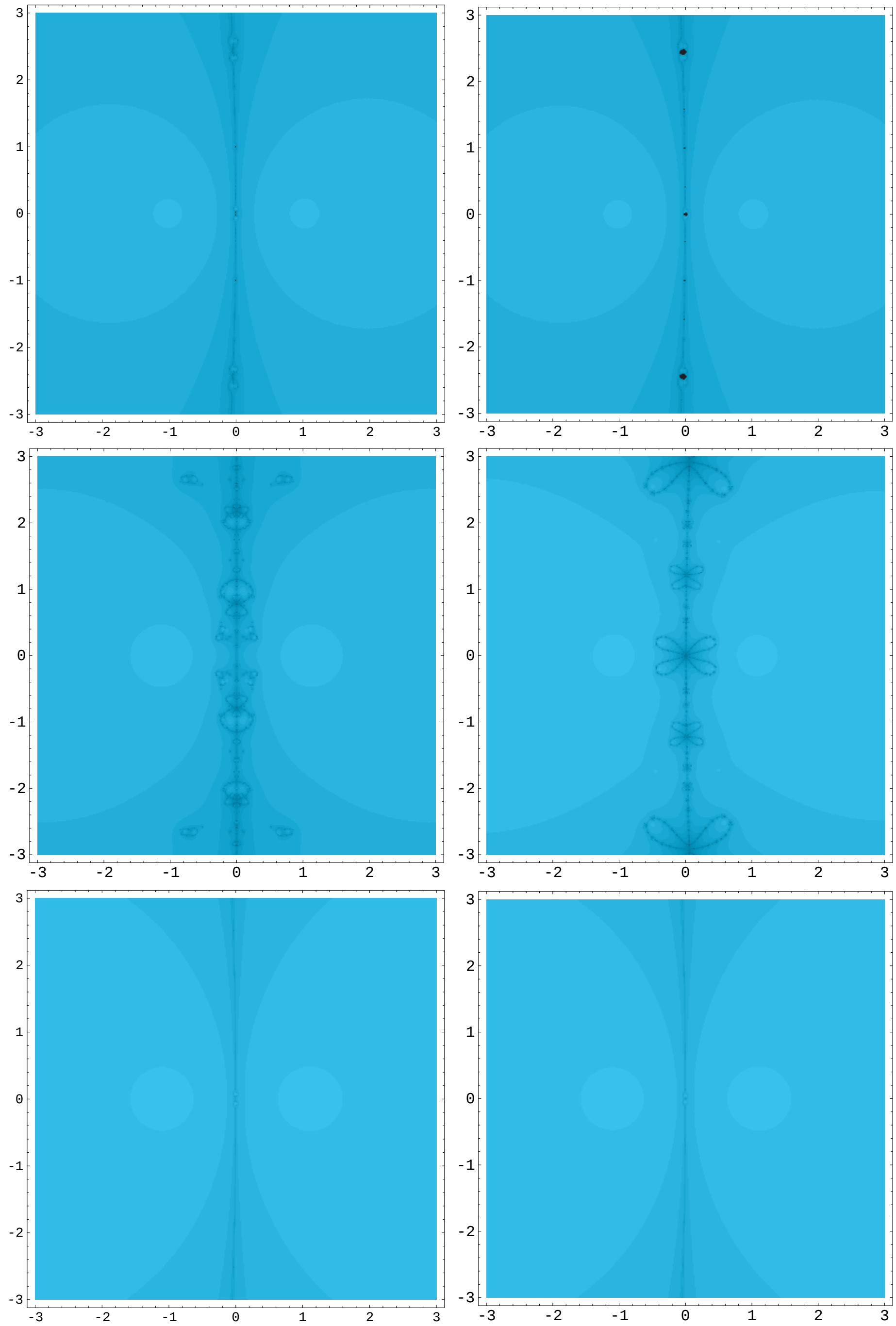

Figure 1: Top row: Methods without memory (18)(left) and (19)(right). Middle row: Methods (1)(left) and (3)(right). Bottom row: Methods (18) with (24)(left) and (19) with (24)(right). The results are for the polynomial $z^2 - 1$.

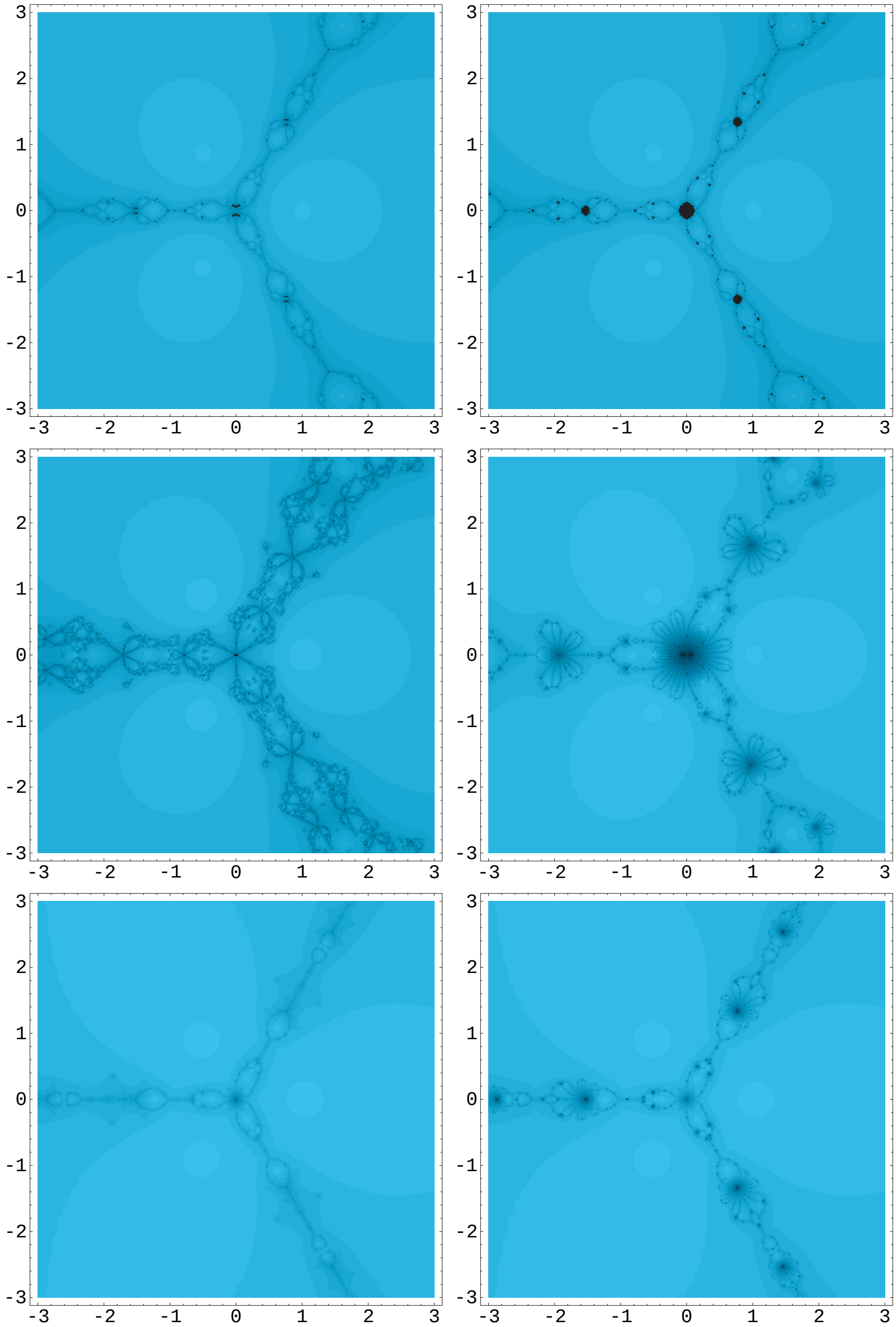

Figure 2: Top row: Methods without memory (18)(left) and (19)(right). Middle row: Methods (1)(left) and (3)(right). Bottom row: Methods (18) with (24)(left) and (19) with (24)(right).The results are for the polynomial $z^3 - 1$.

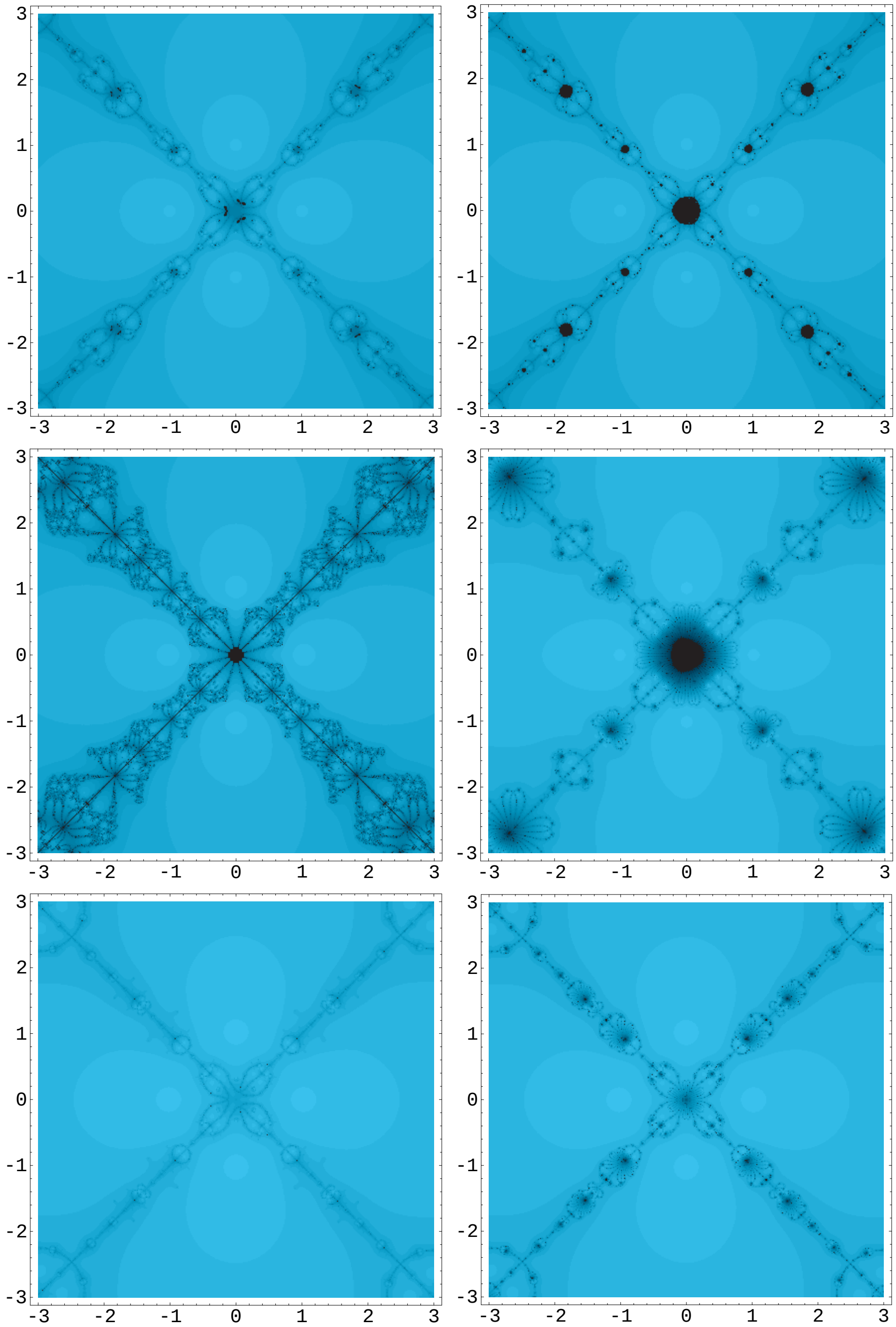

Figure 3: Top row: Methods without memory (18)(left) and (19)(right). Middle row: Methods (1)(left) and (3)(right). Bottom row: Methods (18) with (24)(left) and (19) with (24)(right). The results are for the polynomial $z^4 - 1$.

brighter colors (the less iterations, the brighter color). Black color denotes lack of convergence to any of the roots (with the maximum of iterations established) or convergence to the infinity. The fractals of our methods with memory are almost similar, we only give the basins of attraction of our methods with memory of the R-order 5.7016. The parameters used in iterative methods (18)–(19) without memory are $\gamma = 0.01$ and $\beta = -0.01$. The parameters $\lambda_0 = 0.01$ and $\beta = -0.01$ are used in method (3) with memory. Our methods with memory use the parameters $\gamma_0 = 0.01$ and $\beta_0 = -0.01$ in the first iteration.

Figures 1-3 show that the new methods with memory have very little diverging points compared to other methods. Figures 2-3 show that the basins of attraction for our methods with memory are larger than the other methods and the convergence speed of our methods with memory are faster than our methods (18)–(19) without memory. On the whole, we can see that the new methods with memory are better than the other methods in this paper.

6. Conclusions

In this paper, we have proposed a new family of two-step Newton-type iterative methods with and without memory for solving nonlinear equations. Using two self-correcting parameters calculated by Hermite interpolatory polynomials the R-order of convergence of the new Newton-type method with memory is increased from 4 to 5.7016 without any additional calculations. The main contribution of this paper is that we obtain the maximal order of Newton-type methods without any additional function evaluations. The new methods are compared in performance and computational efficiency with some existing methods by numerical examples. We observed that the computational efficiency indices of the new methods with memory are better than those of other existing two-step methods.

Acknowledgement

The project supported by the National Natural Science Foundation of China (Nos. 11071033 and 11201037). We would like to thank the referees for their helpful suggestions.

References

[1] G. Alefeld, J. Herzberger, *Introduction to Interval Computation*, Academic Press, New York, 1983.

[2] C. Chun, M. Y. Lee, B. Neta, J. Džunić, *On optimal fourth-order iterative methods free from second derivative and their dynamics*, Appl. Math. Comput. **218**(2012), 6427–6438.

[3] A. Cordero, J. R. Torregrosa, *Variants of Newton's Method using fifth-order quadrature formulas*, Appl .Math. Comput. **190**(2007), 686–698.

[4] J. Džunić, M. S. Petković, L. D. Petković, *Three-point methods with and without memory for solving nonlinear equations*, Appl. Math. Comput. **218**(2012), 4917–4927.

[5] B. NETA, *A New Family of Higher Order Methods for Solving Equations*, Int. J. Comput. Math. **14**(1983), 191–195.

[6] J. M. ORTEGA, W. C. RHEINBOLT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[7] A. M. OSTROWSKI, *Solutions of Equations and System of Equations*, Academic Press, New York, 1960.

[8] M. S. PETKOVIĆ, *A note on the priority of optimal multipoint methods for solving nonlinear equations*, Appl. Math. Comput. **219**(2013), 5249–5252.

[9] M. S. PETKOVIĆ, J. DŽUNIĆ, B. NETA, *Interpolatory multipoint methods with memory for solving nonlinear equations*, Appl. Math. Comput.**218** (2011), 2533–2541.

[10] M. G. SÁNCHEZ, À.GRAU, M. NOGUERA, *Ostrowski type methods for solving systems of nonlinear equations*, Appl. Math. Comput. **218**(2011), 2377–2385.

[11] J. R. SHARMA, R. K. GUHA, P. GUPTA, *Some efficient derivative free methods with memory for solving nonlinear equations*, Appl. Math. Comput. **219**(2012), 699–707.

[12] F. SOLEYMANI, M. SHARIFI, B. S. MOUSAVI, *An inprovement of Ostrowski's and King's techniques with optimal convergence order eight*, J. Optim. Theory. Appl. **153**(2012), 225–236.

[13] X. WANG, J. DŽUNIĆ, T. ZHANG, *On an efficient family of derivative free three-point methods for solving nonlinear equations*, Appl. Math. Comput. **219**(2012), 1749–1760.

[14] X. WANG, T. ZHANG, *A new family of Newton-type iterative methods with and without memory for solving nonlinear equations*, Calcolo (2012), 1–15, Doi:10.1007/s10092-012-0072-2.

[15] J. L. VARONA, *Graphic and numerical comparison between iterative methods*, The Math. Intelligencer, **24**(2002),37–46.