Numer Algor (2016) 73:1017–1036
DOI 10.1007/s11075-016-0127-6

ORIGINAL PAPER

Efficient derivative-free variants of Hansen-Patrick's family with memory for solving nonlinear equations

Munish Kansal[1] · V. Kanwar[1] · Saurabh Bhatia[1]

Received: 20 October 2015 / Accepted: 21 March 2016 / Published online: 29 March 2016

Abstract In this paper, we present a new tri-parametric derivative-free family of Hansen-Patrick type methods for solving nonlinear equations numerically. The proposed family requires only three functional evaluations to achieve optimal fourth order of convergence. In addition, acceleration of convergence speed is attained by suitable variation of free parameters in each iterative step. The self-accelerating parameters are estimated from the current and previous iteration. These self-accelerating parameters are calculated using Newton's interpolation polynomials of third and fourth degrees. Consequently, the R-order of convergence is increased from 4 to 7, without any additional functional evaluation. Furthermore, the most striking feature of this contribution is that the proposed schemes can also determine the complex zeros without having to start from a complex initial guess as would be necessary with other methods. Numerical experiments and the comparison of the existing robust methods are included to confirm the theoretical results and high computational efficiency.

Keywords Multipoint iterative methods · Derivative-free methods · Methods with memory · R-order of convergence · Computational efficiency

Mathematics Subject Classificaton (2010) 65H05

✉ V. Kanwar
vmithil@yahoo.co.in

Munish Kansal
mkmaths@gmail.com

Saurabh Bhatia
s_bhatia@pu.ac.in

[1] University Institute of Engineering and Technology, Panjab University, Chandigarh 160 014, India

1 Introduction

Finding rapidly and accurately the zeros of nonlinear functions is an interesting and challenging problem in the field of computational mathematics. In this study, we consider iterative methods for solving a nonlinear equation of the form $f(x) = 0$, where $f : I \subseteq \mathbb{R} \to \mathbb{R}$ is a scalar function defined on an open interval I. Analytical methods for solving such equations are almost non-existent and therefore, it is only possible to obtain approximate solutions by relying on numerical methods based on iterative procedure [1–27]. One of the most famous and basic tool for solving such equations is the Newton's method [1] given by $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},\ n \geq 0$. It converges quadratically for simple roots and linearly for multiple roots. In order to improve its local order of convergence, many higher-order methods have been proposed and analyzed in [2, 3]. One such well-known scheme is the classical cubically convergent Hansen-Patrick's family [4] defined by

$$x_{n+1} = x_n - \left[\frac{\alpha+1}{\alpha \pm \left(1-(\alpha+1)L_f(x_n)\right)^{1/2}}\right]\frac{f(x_n)}{f'(x_n)}, \tag{1.1}$$

where $L_f(x_n) = \frac{f''(x_n)f(x_n)}{f'^2(x_n)}$ and $\alpha \in \mathbb{R}\backslash\{-1\}$. This family includes Ostrowski's square-root method for ($\alpha = 0$), Euler's method for ($\alpha = 1$), Laguerre's method for $\left(\alpha = \frac{1}{\nu-1}, \nu \neq 1\right)$ and as a limiting case, Newton's method. Despite the cubic convergence, this scheme is considered less practical from a computational point of view because of the expensive second-order derivative evaluation. This fact motivated many researchers to investigate the idea of developing multipoint iterative methods for solving nonlinear equations numerically.

Multipoint iterative methods for solving nonlinear equations are of great practical importance since they circumvent the limitations of one-point methods regarding the convergence order and computational efficiency. Generally, multipoint iterative methods are divided into two categories: with memory and without memory methods. The main objective in the construction of the new iterative methods is to obtain the maximal computational efficiency. In other words, the aim is to attain convergence order as high as possible with fixed number of functional evaluation per iteration. According to the Kung-Traub conjecture [5], the order of convergence of any multipoint method without memory requiring n functional evaluations per iteration, cannot exceed the bound 2^{n-1}, called the optimal order. Consequently, convergence order of an optimal iterative method without memory consuming three functional evaluations cannot exceed four. Also, efficiency of an iterative method is measured by the efficiency index [6] defined as $E = p^{1/d}$, where p is the order of convergence and d is the number of functional evaluations required per step. King's family [7], Ostrowski's method [6] and Jarratt's method [8] are the well-known fourth-order multipoint methods without memory. Recently, Sharma et al. [9] introduced a modified two-step scheme of Hansen-Patrick's family given by

$$\begin{cases} y_n = x_n - \alpha\frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \left[\frac{\beta+1}{\beta \pm \{1-(\beta+1)H_f(x_n)\}^{1/2}}\right]\frac{f(x_n)}{f'(x_n)}, \end{cases} \tag{1.2}$$

where $H_f(x_n) = \frac{f''(y_n)f(x_n)}{f'^2(x_n)}$, $\beta\ (\neq -1)$, $\alpha \in \mathbb{R}$. In fact, the authors calculated second-order derivative f'' at y_n instead of x_n. It is worth mentioning that the family of methods (1.2) (except for $\alpha = 1/3$, $\beta = 1$) is not optimal in the sense of Kung-Traub conjecture, since it requires three functional evaluations $f(x_n)$, $f'(x_n)$ and $f''(y_n)$, per full iteration, having only third-order convergence. To remove the second-order derivative, several variants of Hansen-Patrick type methods free from second derivative have been proposed and analyzed in [10]. In [10], the authors proposed a new optimal fourth-order modification of Hansen-Patrick's family given by

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - \left[\dfrac{\alpha+1}{\alpha \pm \left\{ \frac{f(x_n)^2-(\alpha+3)f(x_n)f(y_n)-(\alpha^2-1)f(y_n)^2}{f(x_n)^2+(\alpha-1)f(x_n)f(y_n)} \right\}^{1/2}} \right] \dfrac{f(x_n)}{f'(x_n)}, \quad \alpha\ (\neq -1) \in \mathbb{R} \end{cases} \tag{1.3}$$

From computational point of view, the proposed class (1.3) requires only three functional evaluations viz. $f(x_n)$, $f'(x_n)$ and $f(y_n)$, per full iteration, to achieve an optimal efficiency index $E = 4^{1/3} \approx 1.587$. But, in spite of being optimal, it requires the evaluation of first-order derivative at each iterative step and hence cannot be applied to non-smooth functions.

On the other hand, the basic idea for the construction of multipoint methods with memory was introduced by Traub [1]. He improved a Steffensen-like method by the reuse of information from the previous iteration using secant approach. In fact, he proposed the following method with memory:

$$\begin{cases} \gamma_0 \text{ is given}, \ \gamma_n = \dfrac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}, \quad n \geq 1, \\ x_{n+1} = x_n - \dfrac{\gamma_n f(x_n)^2}{f(x_n + \gamma_n f(x_n)) - f(x_n)}, \end{cases} \tag{1.4}$$

having R-order of convergence [11] atleast $1 + \sqrt{2} \approx 2.414$. A similar approach was applied to higher order multipoint methods in [12–15]. Surprisingly, this particular class of methods with memory is not completely dealt with in the literature in spite of their high computational efficiency.

There are many higher-order iterative methods but most of them use derivatives in the iteration process, which is a serious disadvantage. To overcome this, we suggest and analyze methods for solving nonlinear equations which do not require the derivative of the function. The prime motive of this work is to present a new tri-parametric class of derivative-free methods without memory based on Hansen-Patrick's family having optimal fourth order of convergence. Each member of the proposed family supports Kung-Traub conjecture for $n = 3$. As a matter of fact, many higher-order derivative-free type methods without memory have been already derived in the literature using different techniques, see for instance [16–18] and the references cited therein. Hence, the proposed families can be regarded as an additional contribution to the subject but without additional advantage. However, we do not have any higher-order derivative-free variants of Hansen-Patrick type methods with memory till date.

With this aim, we further attempt to increase the convergence order of the proposed family by applying an accelerating procedure based on varying self-accelerating parameters calculated by Newton's interpolation polynomials in each iterative step. The R-order of convergence of the proposed two-point derivative-free methods with memory is 7. As a result, efficiency index increases from $E = 4^{1/3} \approx 1.587$ to $E = 7^{1/3} \approx 1.913$, which is even better than optimal sixteenth order methods without memory. It is noteworthy that the significant increase of convergence speed is achieved without additional functional evaluations. This means that the proposed methods with memory possess a very high computational efficiency, which is the main advantage of these methods in comparison to the methods without memory. Moreover, it is shown by way of illustration that the proposed schemes can determine the complex zeros without having to start from a complex number as would be necessary with other methods. It is found that the proposed methods are highly efficient in multi-precision computing environment.

2 Derivative-free two-point Hansen-Patrick's family and convergence analysis

In this section, we intend to develop a new derivative-free class of two-point Hansen-Patrick type methods having optimal fourth-order convergence.

Let $y_n = x_n - \frac{f(x_n)}{f'(x_n)}$ be the Newton's iterate. We consider Taylor's expansion of $f(y_n)$ about a point $x = x_n$ as follows:

$$f(y_n) \approx f(x_n) + f'(x_n)(y_n - x_n) + \frac{1}{2} f''(x_n)(y_n - x_n)^2,$$

which implies

$$f''(x_n) \approx \frac{2f'(x_n)^2 f(y_n)}{f(x_n)^2}. \tag{2.1}$$

Substituting the above approximate value of $f''(x_n)$ in scheme (1.1), we obtain

$$x_{n+1} = x_n - \left[\frac{\alpha + 1}{\alpha + \left(1 - \frac{2(\alpha+1) f(y_n)}{f(x_n)}\right)^{1/2}} \right] \frac{f(x_n)}{f'(x_n)}, \quad \alpha \in \mathbb{R}\backslash\{-1\}. \tag{2.2}$$

Re-writing (2.2) in *predictor-corrector* form, we get two-point iterative methods given by

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = y_n - \left[-1 + \frac{\alpha+1}{\alpha + \left(1 - \frac{2(\alpha+1) f(y_n)}{f(x_n)}\right)^{1/2}} \right] \frac{f(x_n)}{f'(x_n)}, \quad \alpha \in \mathbb{R}\backslash\{-1\}. \end{cases} \tag{2.3}$$

In order to obtain optimal derivative-free methods, we replace derivatives in both steps of family (2.3) by suitable approximations that use already available data.

Therefore, we introduce a new tri-parametric derivative-free family of iterative methods given by

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda f(w_n)}, \quad w_n = x_n + \gamma f(x_n), \ \gamma, \lambda \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[y_n, w_n] + \lambda f(w_n)} \left[-1 + \frac{\alpha+1}{\alpha + \left(1 - \frac{2(\alpha+1) f(y_n)}{f(x_n)}\right)^{1/2}} \right] H(\tau), \quad \tau = \dfrac{f(y_n)}{f(x_n)}, \end{cases} \tag{2.4}$$

where $\alpha \in \mathbb{R}\backslash\{-1\}$, $f[x, y] = \frac{f(x)-f(y)}{x-y}$ denotes a first-order divided difference (without index n) and H is a real variable weight function. Theorem 1 illustrates that under what conditions on weight function, convergence order of the family (2.4) will arrive at the optimal level four.

2.1 Convergence analysis

Theorem 1 *Assume that function $f : I \subseteq \mathbb{R} \to \mathbb{R}$ is sufficiently differentiable and has a simple zero $\xi \in I$. If an initial guess x_0 is sufficiently close to $\xi \in I$, then the iterative scheme defined by* (2.4) *has optimal fourth-order convergence when*

$$H(0) = 1, \ H'(0) = \frac{-(\alpha+1)}{2}, \ |H''(0)| < \infty, \ \alpha \in \mathbb{R}\backslash\{-1\}.$$

It satisfies the following error equation

$$e_{n+1} = \Theta_1 \left(1 + \gamma f'(\xi)\right)^2 (\lambda + c_2) e_n^4 + O(e_n)^5,$$

where

$$\begin{cases} \Theta_1 = -\frac{1}{4}\Big[\left(3 + 4\alpha + \alpha^2 + 2H''(0)\right)(1 + \gamma f'(\xi))\lambda^2 + 2a_1 \lambda c_2 + a_2 c_2^2 + 4c_3 \Big], \\ a_1 = -1 + 2H''(0) + 3\gamma f'(\xi) + 2H''(0)\gamma f'(\xi) + 4\alpha(1 + \gamma f'(\xi)) + \alpha^2(1 + \gamma f'(\xi)), \\ a_2 = -5 + 2H''(0) + 3\gamma f'(\xi) + 2H''(0)\gamma f'(\xi) + 4\alpha(1 + \gamma f'(\xi)) + \alpha^2(1 + \gamma f'(\xi)). \end{cases}$$

Proof Let $e_n = x_n - \xi$ be the error at n^{th} iteration and $c_n = \dfrac{1}{n!}\dfrac{f^{(n)}(\xi)}{f'(\xi)}$, $n = 2, 3, \ldots$. Taking taking into account that $f(\xi) = 0$, we can expand $f(x_n)$ and $f(w_n)$ about $x_n = \xi$. Therefore, we get

$$f(x_n) = f'(\xi)\left[e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4\right] + O(e_n^5), \tag{2.5}$$

and

$$f(w_n) = f'(\xi)\left[e_{n,w} + c_2 e_{n,w}^2 + c_3 e_{n,w}^3 + c_4 e_{n,w}^4\right] + O(e_{n,w}^5), \tag{2.6}$$

where $e_{n,w} := w_n - \xi$.

Now, using (2.5) and (2.6) in (2.4), we get

$$e_{n,y} = y_n - \xi = \left(1 + \gamma f'(\xi)\right)(\lambda + c_2)e_n^2 + \Big(\Big(-\left(2 + 2\gamma f'(\xi) + f'(\xi)^2\gamma^2\right)\lambda c_2 - \left(2 + 2\gamma f'(\xi) + f'(\xi)^2\gamma^2\right)c_2^2 - (1 + f'(\xi)\gamma)\left((1 + f'(\xi)\gamma)\lambda^2 - (2 + f'(\xi)\gamma)c_3\right)\Big)\Big)e_n^3 + O(e_n^4). \quad (2.7)$$

Expanding $f\left(x_n - \frac{f(x_n)}{f[x_n, w_n]}\right)$ about $x_n = \xi$, we have

$$f(y_n) = f'(\xi)\left[e_{n,y} + c_2 e_{n,y}^2 + c_3 e_{n,y}^3 + c_4 e_{n,y}^4\right] + O(e_{n,y}^5), \quad (2.8)$$

and

$$\tau = \frac{f(y_n)}{f(x_n)} = (1 + f'(\xi)\gamma)\left(\lambda + c_2\right)e_n + \Big(-\left(2 + 2f'(\xi)\gamma + f'(\xi)^2\gamma^2\right)\lambda c_2 - \left(2 + 2f'(\xi)\gamma + f'(\xi)^2\gamma^2\right)c_2^2 - (1 + f'(\xi)\gamma)c_2\left(\lambda + c_2\right) - (1 + f'(\xi)\gamma) \times \left((1 + f'(\xi)\gamma)\lambda^2 - (2 + f'(\xi)\gamma)c_3\right)\Big)e_n^2 + O(e_n^3). \quad (2.9)$$

Now, expanding the weight function $H(\tau)$ in the neighborhood of origin using Taylor expansion, we get

$$H(\tau) = H(0) + H'(0)\tau + \frac{1}{2!}H''(0)\tau^2 + \frac{1}{3!}H'''(0)\tau^3 + O(\tau^4). \quad (2.10)$$

Using (2.5)–(2.10) in scheme (2.4), we obtain the following error equation

$$\begin{aligned} e_{n+1} = &- c_2\left(H(0) - 1\right)\left(1 + \gamma f'(\xi)\right)(\lambda + c_2) \\ &+ \Big[- \left(2 + 2f'(\xi)\gamma + f'(\xi)^2\gamma^2\right)\lambda c_2 - \left(2 + 2f'(\xi)\gamma + f'(\xi)^2\gamma^2\right)c_2^2 \\ &- H'(0)(1 + f'(\xi)\gamma)^2\left(\lambda + c_2\right)^2 - (1 + f'(\xi)\gamma)\left((1 + f'(\xi)\gamma)\lambda^2 - (2 + f'(\xi)\gamma)c_3\right) \\ &- \frac{1}{2}H(0)\Big(2\left(- 1 + \alpha(1 + f'(\xi)\gamma)^2\right)\lambda c_2 \\ &+ \left(- 3 - 2f'(\xi)\gamma - f'(\xi)^2\gamma^2 + \alpha(1 + f'(\xi)\gamma)^2\right)c_2^2 + (1 + f'(\xi)\gamma) \\ &\times \left((-1 + \alpha)(1 + f'(\xi)\gamma)\lambda^2 + 2(2 + f'(\xi)\gamma)c_3\right)\Big)\Big]e_n^3 \\ &+ \Gamma_4 e_n^4 + O(e_n^5), \end{aligned} \quad (2.11)$$

where $\Gamma_4 = \Gamma_4\left(c_2, c_3, c_4, \alpha, \gamma, \lambda, H(0), H^i(0)\right)$ for $i = 1, 2$.

Therefore, to achieve fourth-order convergence, co-efficients of e_n^2 and e_n^3 should vanish simultaneously. Hence, substituting co-efficients of e_n^2 and e_n^3 in (2.11) equal to zero, we get the following conditions

$$H(0) = 1,\ H'(0) = \frac{-(\alpha + 1)}{2},\ |H''(0)| < \infty,\ \alpha \in \mathbb{R}\backslash\{-1\}.$$

Thus, the iterative scheme (2.4) satisfies the following error equation

$$e_{n+1} = \Theta_1 \left(1 + \gamma f'(\xi)\right)^2 (\lambda + c_2)e_n^4 + O(e_n)^5, \tag{2.12}$$

where

$$\begin{cases} \Theta_1 = -\frac{1}{4}\Big[\left(3 + 4\alpha + \alpha^2 + 2H''(0)\right)(1 + \gamma f'(\xi))\lambda^2 + 2a_1\lambda c_2 + a_2c_2^2 + 4c_3\Big], \\ a_1 = -1 + 2H''(0) + 3\gamma f'(\xi) + 2H''(0)\gamma f'(\xi) + 4\alpha(1 + \gamma f'(\xi)) + \alpha^2(1 + \gamma f'(\xi)), \\ a_2 = -5 + 2H''(0) + 3\gamma f'(\xi) + 2H''(0)\gamma f'(\xi) + 4\alpha(1 + \gamma f'(\xi)) + \alpha^2(1 + \gamma f'(\xi)). \end{cases}$$

This reveals that the modified derivative-free two-point Hansen-Patrick's family (2.4) attains fourth-order convergence requiring only three functional evaluations, viz., $f(x_n)$, $f(w_n)$ and $f(y_n)$, per step. Hence, optimal efficiency index of the proposed class is $E = \sqrt[3]{4} \approx 1.587$. This completes the proof. □

Remark 1 Error relation (2.12) plays an important role to construct new derivative-free iterative methods with memory in Section 4.

3 Special cases

In what follows, we present some concrete explicit representations of our proposed class (2.4) by choosing different weight functions satisfying all the conditions of the Theorem 1.

Case 1 Let us consider the following weight function

$$H_1(\tau) = 1 - \frac{\alpha + 1}{2}\tau. \tag{3.1}$$

Using the above weight function in scheme (2.4), we get

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda f(w_n)}, \quad w_n = x_n + \gamma f(x_n), \; \gamma, \lambda \in \mathbb{R}\backslash\{0\} \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[y_n, w_n] + \lambda f(w_n)} \left(-1 + \dfrac{\alpha + 1}{\alpha + \left(1 - \frac{2(\alpha+1)f(y_n)}{f(x_n)}\right)^{1/2}}\right)\left(1 - \dfrac{\alpha + 1}{2}\dfrac{f(y_n)}{f(x_n)}\right), \end{cases} \tag{3.2}$$

where $\alpha \in \mathbb{R}\backslash\{-1\}$.

This is a new tri-parametric optimal fourth-order derivative-free class of Hansen-Patrick type methods and one can easily get many new families of methods by choosing different values of parameters γ, λ and α. By fixing one of the free disposable parameters, we display some interesting special cases of family (3.2).

Sub special cases of optimal family (3.2)

(i) For $\alpha = 0$, family (3.2) reads as

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda f(w_n)}, \quad w_n = x_n + \gamma f(x_n), \ \gamma, \lambda \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[y_n, w_n] + \lambda f(w_n)} \left[-1 + \dfrac{1}{\sqrt{1 - \frac{2f(y_n)}{f(x_n)}}} \right] \left[1 - \dfrac{1}{2} \dfrac{f(y_n)}{f(x_n)} \right]. \end{cases} \tag{3.3}$$

This is a new optimal fourth-order derivative-free modification of Ostrowski's square-root method.

(ii) For $\alpha = 1$, family (3.2) reads as

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda f(w_n)}, \quad w_n = x_n + \gamma f(x_n), \ \gamma, \lambda \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[y_n, w_n] + \lambda f(w_n)} \left[-1 + \dfrac{2}{1 + \sqrt{1 - \frac{4f(y_n)}{f(x_n)}}} \right] \left[1 - \dfrac{f(y_n)}{f(x_n)} \right]. \end{cases} \tag{3.4}$$

This is a new optimal fourth-order derivative-free modification of Euler's method.

(iii) Taking $\alpha = \frac{1}{\nu - 1}$ $(\nu \neq 1)$, family (3.2) reads as

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda f(w_n)}, \quad w_n = x_n + \gamma f(x_n), \ \gamma, \lambda \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[y_n, w_n] + \lambda f(w_n)} \left[-1 + \dfrac{\nu}{1 + (\nu - 1)\sqrt{1 + \frac{2\nu f(y_n)}{(\nu-1) f(x_n)}}} \right] \left[1 + \dfrac{\nu}{2(1-\nu)} \dfrac{f(y_n)}{f(x_n)} \right]. \end{cases} \tag{3.5}$$

This is a new optimal fourth-order derivative-free modification of Laguerre's method.

Case 2 Let us consider the following weight function

$$H_2(\tau) = \frac{1}{1 + (1 + \alpha)\tau}. \tag{3.6}$$

Using the above weight function in scheme (2.4), we get

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda f(w_n)}, \quad w_n = x_n + \gamma f(x_n), \ \gamma, \lambda \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[y_n, w_n] + \lambda f(w_n)} \left[-1 + \dfrac{\alpha + 1}{\alpha + \left(1 - \frac{2(\alpha+1) f(y_n)}{f(x_n)} \right)^{1/2}} \right] \left[\dfrac{f(x_n)}{f(x_n) + (1 + \alpha) f(y_n)} \right], \end{cases} \tag{3.7}$$

where $\alpha \in \mathbb{R}\backslash\{-1\}$. This is another new tri-parametric optimal fourth-order derivative-free class of Hansen-Patrick type methods.

Sub special cases of optimal family (3.7)

(i) For $\alpha = 0$, family (3.7) reads as

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda f(w_n)}, \quad w_n = x_n + \gamma f(x_n), \ \gamma, \lambda \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[y_n, w_n] + \lambda f(w_n)} \left[-1 + \dfrac{1}{\sqrt{1 - \frac{2f(y_n)}{f(x_n)}}} \right] \left[\dfrac{f(x_n)}{f(x_n) + f(y_n)} \right]. \end{cases} \tag{3.8}$$

This is another new optimal fourth-order derivative-free modification of Ostrowski's square-root method.

(ii) For $\alpha = 1$, family (3.2) reads as

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda f(w_n)}, \quad w_n = x_n + \gamma f(x_n), \ \gamma, \lambda \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[y_n, w_n] + \lambda f(w_n)} \left[-1 + \dfrac{2}{1 + \sqrt{1 - \frac{4f(y_n)}{f(x_n)}}} \right] \left[\dfrac{f(x_n)}{f(x_n) + 2f(y_n)} \right]. \end{cases} \tag{3.9}$$

This is another new optimal fourth-order derivative-free modification of Euler's method.

(iii) Taking $\alpha = \frac{1}{\nu - 1}$ $(\nu \neq 1)$, family (3.2) reads as

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda f(w_n)}, \quad w_n = x_n + \gamma f(x_n), \ \gamma, \lambda \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[y_n, w_n] + \lambda f(w_n)} \left[-1 + \dfrac{\nu}{1 + (\nu - 1)\sqrt{1 + \frac{2\nu f(y_n)}{(\nu - 1) f(x_n)}}} \right] \left[\dfrac{f(x_n)}{f(x_n) + (\nu/(\nu - 1)) f(y_n)} \right]. \end{cases} \tag{3.10}$$

This is another new optimal fourth-order derivative-free modification of Laguerre's method.

Some more simple weight functions satisfying the conditions of Theorem (1) are given below:

$$H_3(\tau) = \frac{1}{1 + \frac{\alpha+1}{2}\tau + \beta_1 \tau^2}, \quad H_4(\tau) = \frac{1 + \tau}{1 + \frac{\alpha+3}{2}\tau}, \quad H_5(\tau) = \frac{1 + \beta_2 \tau^2}{1 + \frac{\alpha+1}{2}\tau},$$

where $\alpha \backslash \{-1\}$, β_1, β_2 are free disposable parameters.

4 Development of new methods with memory and convergence analysis

In this section, we are going to construct new iterative methods with memory from (2.4) using two self-accelerating parameters.

It is clear from error (2.12) that the order of convergence of the family (2.4) is four, when $\gamma \neq -1/f'(\xi)$ and $\lambda \neq -c_2$. Therefore, it is possible to increase the convergence speed of the proposed class (2.4), if $\gamma = -1/f'(\xi)$ and $\lambda =$

$-c_2 = -f''(\xi)/(2f'(\xi))$. However, the values of $f'(\xi)$ and $f''(\xi)$ are not available in practice and such acceleration is not possible. Instead of that, we could use approximations $\tilde{f}'(\xi) \approx f'(\xi)$ and $\tilde{f}''(\xi) \approx f''(\xi)$, calculated by already available information. Therefore, by setting $\gamma = -1/\tilde{f}'(\xi)$ and $\lambda = -c_2 = -\tilde{f}''(\xi)/(2\tilde{f}'(\xi))$, we can increase the convergence order without using any new functional evaluations. Hence, the main idea in constructing methods with memory consists of the calculation of the parameters $\gamma = \gamma_n$ and $\lambda = \lambda_n$ as the iteration proceeds by the formulas $\gamma_n = -1/\tilde{f}'(\xi)$ and $\lambda_n = -c_2 = -\tilde{f}''(\xi)/(2\tilde{f}'(\xi))$ for $n = 1, 2, 3, \ldots$. Further, it is also assumed that the initial estimates γ_0 and λ_0 should be chosen before starting the iterative process, for example, using one of the ways proposed in [1].
In what follows, we use symbols $\rightarrow$, O and $\sim$ according to the following convention: If $\lim_{n\to\infty} f(x_n) = C$, we write $f(x_n) \rightarrow C$ or $f \rightarrow C$, where C is a nonzero constant. If $\frac{f}{g} \rightarrow C$, we will write $f = O(g)$ or $f \sim g$.

Therefore, we approximate

$$\gamma_n = \frac{-1}{\tilde{f}'(\xi)} = \frac{-1}{N_3'(x_n)}, \tag{4.1}$$

and

$$\lambda_n = -\frac{N_4''(w_n)}{2N_4'(w_n)}, \tag{4.2}$$

in iterative scheme (2.4). Here, $N_3(t) = N_3(t; x_n, x_{n-1}, y_{n-1}, w_{n-1})$ and $N_4(t) = N_4(t; w_n, x_n, w_{n-1}, y_{n-1}, x_{n-1})$ are Newton interpolating polynomials of third and fourth degree, set through available nodal points $(x_n, x_{n-1}, y_{n-1}, w_{n-1})$ and $(w_n, x_n, w_{n-1}, y_{n-1}, x_{n-1})$, respectively.

If we use lower degree interpolating polynomials, then slower acceleration is achieved. Secondly, the other choices of nodes (of worse quality) gives approximations for γ_n and λ_n in (2.4) of somewhat less accuracy. Therefore, we have considered the best possible choices for nodal points to obtain the maximal order.

It is worth mentioning that the evaluation of the self-accelerating parameters γ_n and λ_n depends on the data available from the current and the previous iterations. Therefore, order of convergence will be increased significantly without using an extra functional evaluation. Finally, replacing fixed parameters γ and λ in (2.4) by the varying parameters γ_n and λ_n defined by (4.1) and (4.2), we shall obtain new methods with memory. Hence, with memory versions of derivative-free methods (2.4) can be presented as follows:

$$\begin{cases} x_0, \gamma_0, \lambda_0 \text{ are given, then } w_0 = x_0 + \gamma_0 f(x_0), \\ \gamma_n = \dfrac{-1}{N_3'(x_n)}, \; w_n = x_n + \gamma_n f(x_n), \; \lambda_n = -\dfrac{N_4''(w_n)}{2N_4'(w_n)}, \; n = 1, 2, \ldots, \\ y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda_n f(w_n)}, \; w_n = x_n + \gamma_n f(x_n), \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[y_n, w_n] + \lambda_n f(w_n)} \left[-1 + \dfrac{\alpha + 1}{\alpha + \left(1 - \frac{2(\alpha+1) f(y_n)}{f(x_n)}\right)^{1/2}} \right] H(\tau), \; \tau = \dfrac{f(y_n)}{f(x_n)}, \end{cases} \tag{4.3}$$

where the weight function $H(\tau)$ satisfy the conditions $H(0) = 1$, $H'(0) = \frac{-(\alpha+1)}{2}$, $|H''(0)| < \infty$ and $\alpha \in \mathbb{R}\backslash\{-1\}$.

In the next subsection, we will establish the convergence results for the new derivative-free with memory variants (4.3) which are based on Hansen-Patrick type methods.

4.1 Convergence analysis

Here, we attempt to prove that the R-order of convergence of a new derivative-free methods (4.3) with memory is seven. For this purpose, we state the following lemma [19].

Lemma 1 *Let* $\gamma_n = \dfrac{-1}{N_3'(x_n)}$ *and* $\lambda_n = -\dfrac{N_4''(w_n)}{2N_4'(w_n)}$, *where* $e_n = x_n - \xi$, $e_{n,w} = w_n - \xi$ *and* $e_{n,y} = y_n - \xi$, *then the following asymptotic relations hold:*

$$1 + \gamma_n f'(\xi) \sim \psi_1 e_{n-1} e_{n-1,w} e_{n-1,y} \text{ and } c_2 + \lambda_n \sim \psi_2 e_{n-1} e_{n-1,w} e_{n-1,y}, \quad (4.4)$$

where ψ_1 *and* ψ_2 *are some asymptotic constants.*

Now, we state the convergence theorem for the scheme (4.3).

Theorem 2 *If an initial guess* x_0 *is sufficiently close to the zero* ξ *of* $f(x)$ *and the parameters* γ_n *and* λ_n *in the iterative scheme* (4.3) *are recursively calculated by the forms given in* (4.1) *and* (4.2), *respectively, then the* R*-order of convergence of methods with memory* (4.3) *is at least seven.*

Proof Let $\{x_n\}$ be a sequence of approximations generated by an iterative method (IM). If this sequence converges to the zero ξ of f with the R-order ($\geq r$) of IM, then we write

$$e_{n+1} \sim D_{n,r} e_n^r, \quad e_n = x_n - \xi,$$

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

$$e_{n+1} \sim D_{n,r} \left(D_{n-1,r} e_{n-1}^r\right)^r = D_{n,r} D_{n-1,r}^r e_{n-1}^{r^2}. \quad (4.5)$$

Assume that the iterative sequences $\{w_n\}$ and $\{y_n\}$ have R-orders r_1 and r_2, respectively. Therefore, we obtain

$$e_{n,w} = w_n - \xi \sim D_{n,r_1} e_n^{r_1} \sim D_{n,r_1} \left(D_{n-1,r} e_{n-1}^r\right)^{r_1} = D_{n,r_1} D_{n-1,r}^{r_1} e_{n-1}^{r r_1}, \quad (4.6)$$

and

$$e_{n,y} = y_n - \xi \sim D_{n,r_2} e_n^{r_2} \sim D_{n,r_2} \left(D_{n-1,r} e_{n-1}^r\right)^{r_2} = D_{n,r_2} D_{n-1,r}^{r_2} e_{n-1}^{r r_2}. \quad (4.7)$$

Using (4.6), (4.7) and Lemma 1, we obtain

$$\begin{aligned} 1 + \gamma f'(\xi) &\sim \psi_1 e_{n-1,w} e_{n-1,y} e_{n-1} = \psi_1 D_{n-1,r_1} D_{n-1,r_2} e_{n-1}^{r_1+r_2+1}, \\ \lambda_n + c_2 &\sim \psi_2 D_{n-1,r_1} D_{n-1,r_2} e_{n-1}^{r_1+r_2+1}. \end{aligned} \quad (4.8)$$

In view of two-point methods (2.4) without memory, we have the following error relations

$$e_{n,w} = \left(1 + \gamma f'(\xi)\right) e_n + O(e_n)^2, \tag{4.9}$$

$$e_{n,y} = c_2 \left(1 + \gamma f'(\xi)\right) (\lambda + c_2) e_n^2 + O(e_n)^3, \tag{4.10}$$

$$e_{n+1} = \Theta_1 \left(1 + \gamma f'(\xi)\right)^2 (\lambda + c_2) e_n^4 + O(e_n)^5, \tag{4.11}$$

where

$$\begin{cases} \Theta_1 = -\frac{1}{4}\Big[\left(3 + 4\alpha + \alpha^2 + 2H''(0)\right)(1 + \gamma f'(\xi))\lambda^2 + 2a_1\lambda c_2 + a_2 c_2^2 + 4c_3\Big], \\ a_1 = -1 + 2H''(0) + 3\gamma f'(\xi) + 2H''(0)\gamma f'(\xi) + 4\alpha(1 + \gamma f'(\xi)) + \alpha^2(1 + \gamma f'(\xi)), \\ a_2 = -5 + 2H''(0) + 3\gamma f'(\xi) + 2H''(0)\gamma f'(\xi) + 4\alpha(1 + \gamma f'(\xi)) + \alpha^2(1 + \gamma f'(\xi)). \end{cases}$$

According to the error relations (4.9)–(4.11) with self-accelerating-parameters $\gamma = \gamma_n$ and $\lambda = \lambda_n$, we can write the corresponding error relations for the methods (2.4) with memory as follow:

$$e_{n,w} \sim \left(1 + \gamma_n f'(\xi)\right) e_n, \tag{4.12}$$

$$e_{n,y} \sim c_2 \left(1 + \gamma_n f'(\xi)\right) (\lambda_n + c_2) e_n^2, \tag{4.13}$$

$$e_{n+1} \sim a_{n,4} \left(1 + \gamma_n f'(\xi)\right)^2 (\lambda_n + c_2) e_n^4, \tag{4.14}$$

where $a_{n,4}$ is clear from (4.11) and depends on iteration index since γ_n and λ_n are re-calculated in each step.

Using Lemma 1 and (4.12)–(4.14), we get

$$e_{n,w} \sim \left(1 + \gamma_n f'(\xi)\right) e_n \sim \left(\psi_1 e_{n-1} e_{n-1,w} e_{n-1,y}\right) e_n \sim \psi_1 D_{n-1,r_1} D_{n-1,r_2} D_{n-1,r} e_{n-1}^{r+r_1+r_2+1}, \tag{4.15}$$

$$e_{n,y} \sim c_2 \left(1 + \gamma_n f'(\xi)\right) (\lambda_n + c_2) e_n^2 \sim c_2 \psi_1 \psi_2 D_{n-1,r_1}^2 D_{n-1,r_2}^2 D_{n-1,r}^2 e_{n-1}^{2r+2r_1+2r_2+2}, \tag{4.16}$$

$$e_{n+1} \sim a_{n,4} \left(1 + \gamma_n f'(\xi)\right)^2 (\lambda_n + c_2) e_n^4 \sim a_{n,4} \psi_1^2 \psi_2 D_{n-1,r_1}^3 D_{n-1,r_2}^3 D_{n-1,r}^4 e_{n-1}^{4r+3r_1+3r_2+3}. \tag{4.17}$$

Now, comparing the error exponents of e_{n-1} on the right hand sides of pairs (4.6)–(4.15), (4.7)–(4.16) and (4.5)–(4.17), respectively, we obtain the following system

$$\begin{cases} rr_1 - r - r_1 - r_2 = 1, \\ rr_2 - 2r - 2r_1 - 2r_2 = 2, \\ r^2 - 4r - 3r_1 - 3r_2 = 3. \end{cases}$$

Therefore, non-trivial solution of this system of equations is given by $r_1 = 2$, $r_2 = 4$ and $r = 7$. Thus, we can conclude that the lower bound of the R-order of the methods with memory (4.3) is *seven*. □

Remark 2 It can be easily seen that the improvement of convergence order from 4 to 7 (75 % of an improvement) is attained without any additional functional evaluations, which points to a very high computational efficiency of the proposed methods (4.3). Therefore, the efficiency index of the proposed methods (4.3) $E = 7^{1/3} \approx 1.913$

which is much higher than the E-values viz., $E = 4^{1/3} \approx 1.587$, $E = 8^{1/4} \approx 1.682$ and $E = 16^{1/5} \approx 1.741$ of the optimal fourth, eighth and sixteenth order methods, respectively.

Remark 3 We also emphasize that the further increase in the convergence speed may be obtained at the cost of introducing more self-accelerating parameters in the proposed iterative processes. However, keeping in mind that methods with memory have somewhat complex structure dealing with information from two successive iterations; we observe that the presented derivative-free Hansen-Patrick type methods with memory have somewhat simpler body structures using only two accelerating parameters.

5 Numerical examples and conclusion

In this section, we shall check the convergence behavior of newly proposed schemes (2.4) and (4.3) using different weight functions with some other methods having same order of convergence. All computations have been performed using the programming package *Mathematica* 7 [20] in multiple precision arithmetic environment. We have considered 2000 digits floating point arithmetic so as to minimize the round-off errors as much as possible. It is assumed that the initial estimates γ_0 and λ_0 should be chosen before starting the iterative process, and also x_0 is given suitably.

For comparisons, we have considered the following concrete methods:

Derivative-free fourth-order Kung-Traub method without memory (KTM4) [5]

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = y_n - \dfrac{f(y_n)f(w_n)}{\big[f(w_n) - f(y_n)\big]f[x_n, y_n]}. \end{cases} \tag{5.1}$$

Derivative-free fourth-order Zheng et al. method (ZLM4) without memory [21]

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \quad w_n = x_n + \beta f(x_n), \quad \beta \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = y_n - \dfrac{f(y_n)}{f[x_n, y_n] + (y_n - x_n)f[x_n, w_n, y_n]}. \end{cases} \tag{5.2}$$

Derivative-free fourth-order Soleymani et al. method (SSLT4) without memory [22]

$$\begin{cases} y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n]}, \ w_n = x_n + \gamma f(x_n), \quad \gamma \in \mathbb{R}\backslash\{0\}, \\ x_{n+1} = x_n - \dfrac{f(x_n) + f(y_n)}{f[x_n, w_n]} - \left(\dfrac{2f(x_n) + af(y_n)}{f[x_n, w_n]}\left(\dfrac{f(y_n)}{f(y_n)}\right)^2\right)\left(1 - \dfrac{\gamma f[x_n, w_n]}{2 + 2\gamma f[x_n, w_n]}\right), \ a \in \mathbb{R}. \end{cases} \tag{5.3}$$

Table 1 $f_1(x) = |x^2 - 4|,\ \xi = 2,\ x_0 = 2.25$

Without memory methods	$\lvert x_1 - \xi\rvert$	$\lvert x_2 - \xi\rvert$	$\lvert x_3 - \xi\rvert$	ρ_c
KTM4 (5.1) ($\beta = -0.01$)	0.8333e−4	0.1389e−17	0.1071e−72	4.0000
ZLM4 (5.2) ($\beta = -0.01$)	0.4413e−4	0.5460e−19	0.1280e−78	4.0000
SSLT4 (5.3) ($\gamma = -0.01,\ a = 5$)	**0.3066e−4**	**0.1295e−20**	**0.4119e−86**	**4.0000**
Our method (2.4) using $H_1(\tau)$ ($\alpha = 1/2$)	0.3278e−4	0.1210e−19	0.2252e−81	4.0000
Our method (2.4) using $H_3(\tau)$ ($\alpha = 1/4, \beta_1 = -1/2$)	**0.1717e−4**	**0.2804e−21**	**0.1996e−88**	**4.0000**

Bold face numbers denote the least error among the displayed methods in case of convergence to the desired root

Derivative-free seventh-order Cordero et al. method (CLBT7) with memory [23]

$$\begin{cases} x_0, \gamma_0, \lambda_0 \text{ are given, then } w_0 = x_0 + \gamma_0 f(x_0), \\ \gamma_n = \dfrac{-1}{N_3'(x_n)},\ w_n = x_n + \gamma_n f(x_n),\ \lambda_n = -\dfrac{N_4''(w_n)}{2N_4'(w_n)},\ n = 1, 2, \ldots, \\ y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda_n f(w_n)},\quad w_n = x_n + \gamma_n f(x_n), \\ x_{n+1} = y_n - \dfrac{f(x_n)}{f[x_n, y_n] + (y_n - x_n) f[x_n, w_n, y_n]}. \end{cases} \tag{5.4}$$

Derivative-free seventh-order Dzunic's method (D7) with memory [24]

$$\begin{cases} x_0, \gamma_0, \lambda_0 \text{ are given, then } w_0 = x_0 + \gamma_0 f(x_0), \\ \gamma_n = \dfrac{-1}{N_3'(x_n)},\ w_n = x_n + \gamma_n f(x_n),\ \lambda_n = -\dfrac{N_4''(w_n)}{2N_4'(w_n)},\ n = 1, 2, \ldots, \\ y_n = x_n - \dfrac{f(x_n)}{f[x_n, w_n] + \lambda_n f(w_n)},\quad w_n = x_n + \gamma_n f(x_n), \\ x_{n+1} = y_n - \dfrac{f(y_n)}{f[y_n, w_n] + \lambda_n f(w_n)} g(t_n),\quad t_n = \dfrac{f(y_n)}{f(x_n)}, \end{cases} \tag{5.5}$$

where $g(t)$ is a real-valued weight function such that $g(0) = 1, g'(0) = 1$ and $|g''(0)| < \infty$.

Table 2 $f_2(x) = x + e\sin x - M,\ \xi = 2,\ x_0 = M + e, e = 0.9995,\ M = 0.01$

Without memory methods	$\lvert x_1 - \xi\rvert$	$\lvert x_2 - \xi\rvert$	$\lvert x_3 - \xi\rvert$	ρ_c
KTM4 (5.1) ($\beta = -0.01$)	0.1219e+0	0.2257e−2	0.6781e−9	3.5156
ZLM4 (5.2) ($\beta = -0.01$)	0.9395e−1	0.4876e−3	0.6066e−12	3.7378
SSLT4 (5.3) ($\gamma = -0.01,\ a = 5$)	0.1196e+0	0.1602e−2	0.2762e−10	3.8923
Our method (2.4) using $H_1(\tau)$ ($\alpha = 1/2$)	**0.6957e−1**	**0.9289e−4**	**0.5172e−15**	**3.8436**
Our method (2.4) using $H_3(\tau)$ ($\alpha = 1/4, \beta_1 = -1/2$)	**0.7804e−1**	**0.6873e−4**	**0.5912e−17**	**3.8083**

Bold face numbers denote the least error among the displayed methods in case of convergence to the desired root

Table 3 $f_3(x) = \sin(\pi x)e^{x^2+x\cos x-1} + x\log(x\sin x + 1),\ \xi = 0,\ x_0 = 0.50$

Without memory methods	$\|x_1 - \xi\|$	$\|x_2 - \xi\|$	$\|x_3 - \xi\|$	ρ_c
KTM4 (5.1) ($\beta = -0.01$)	0.8766e−2	0.7426e−2	0.3801e−32	3.9979
ZLM4 (5.2) ($\beta = -0.01$)	0.8709e−2	0.1700e−8	0.2282e−35	4.0028
SSLT4 (5.3) ($\gamma = -0.01,\ a = 5$)	0.8683e−2	0.3081e−8	0.6065e−34	3.9829
Our method (2.4) using $H_1(\tau)$ ($\alpha = 1/2$)	**0.8111e−2**	**0.4249e−10**	**0.3741e−43**	**3.9902**
Our method (2.4) using $H_3(\tau)$ ($\alpha = 1/4,\ \beta_1 = -1/2$)	0.8089e−2	0.2229e−8	0.1401e−34	3.9921

Bold face numbers denote the least error among the displayed methods in case of convergence to the desired root

To check the theoretical order of convergence, we calculate the computational order of convergence [25] (COC) denoted by ρ_c using the following formula

$$\rho_c = \frac{\log(|f(x_n)/f(x_{n-1})|)}{\log(|f(x_{n-1})/f(x_{n-2})|)},$$

taking into consideration the last three approximations in the iteration process. We have considered variety of test functions of different nature to compute the errors $|x_n - \xi|$ of approximations. For example, the test function f_4 is a polynomial of *Wilkinson's type* with real zeros 1,2,3,4,5. It is well-known that this class of polynomials is ill-conditioned and small perturbations in polynomial coefficients cause drastic variations of zeros. Therefore, most of the iterative methods encounter serious difficulties in finding the zeros of *Wilkinson-like polynomials*.

In the second example, we consider *Kepler's* equation given by $f_2(x) = x - e\sin x - M = 0$, where $0 \le e \le 1$ and $0 \le M \le \pi$. A numerical study, for different

Table 4 $f_4(x) = \prod_{i=1}^{5}(x - i),\ \xi = 3,\ x_0 = 3.25$

With memory methods	$\|x_1 - \xi\|$	$\|x_2 - \xi\|$	$\|x_3 - \xi\|$	ρ_c
CLBT7 (5.4) ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.4231e−2	0.2727e−18	0.1454e−135	7.2432
D7(5.5) using $g(t) = 1 + t$, ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.7207e−2	0.5369e−17	0.4840e−126	7.2083
D7(5.5) using $g(t) = 1/(1-t)$, ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.5478e−2	0.1102e−17	0.4292e−131	7.2252
Our method (4.3) using $H_1(\tau)$ ($\alpha = 1/2$)	0.3945e−2	0.1889e−18	0.9667e−137	7.2252
Our method (4.3) using $H_1(\tau)$ ($\alpha = 1$)	0.3312e−2	0.7684e−19	0.1257e−139	7.2612
Our method (4.3) using $H_2(\tau)$ ($\alpha = 1/2$)	0.6816e−2	0.1421e−17	0.7832e−130	7.1590
Our method (4.3) using $H_2(\tau)$ ($\alpha = 1$)	0.1430e−1	0.2233e−16	0.1254e−116	6.7708
Our method (4.3) using $H_3(\tau)$ ($\alpha = 1/2, \beta_1 = -1/2$)	**0.1803e−2**	**0.3850e−20**	**0.2951e−149**	**7.3068**
Our method (4.3) using $H_3(\tau)$ ($\alpha = 1, \beta_1 = -1/2$)	**0.1914e−3**	**0.1569e−24**	**0.6215e−182**	**7.4647**

Bold face numbers denote the least error among the displayed methods in case of convergence to the desired root

Table 5 $f_5(x) = e^{-x^2+x+2} - 1,\ \xi = -1,\ x_0 = -0.8$

With memory methods	$\lvert x_1 - \xi \rvert$	$\lvert x_2 - \xi \rvert$	$\lvert x_3 - \xi \rvert$	ρ_c
CLBT7 (5.4) ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	**0.9764e−3**	**0.2109e−21**	**0.3538e−152**	**7.0060**
D7(5.5) using $g(t) = 1 + t$, ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.2525e−2	0.2258e−18	0.5913e−131	7.0145
D7(5.5) using $g(t) = 1/(1 - t)$, ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.1902e−2	0.3000e−19	0.4282e−137	7.0133
Our method (4.3) using $H_1(\tau)$ ($\alpha = 1/2$)	**0.6635e−3**	**0.8121e−23**	**0.4473e−162**	**6.9935**
Our method (4.3) using $H_1(\tau)$ ($\alpha = 1$)	**0.2976e−3**	**0.8014e−24**	**0.4015e−169**	**7.0638**
Our method (4.3) using $H_2(\tau)$ ($\alpha = 1/2$)	0.3192e−2	0.5515e−22	0.1904e−139	5.9432
Our method (4.3) using $H_2(\tau)$ ($\alpha = 1$)	0.2813e−2	0.1804e−18	0.2260e−133	7.0952
Our method (4.3) using $H_3(\tau)$ ($\alpha = 1/2, \beta_1 = -1/2$)	**0.2782e−5**	**0.2989e−32**	**0.4050e−228**	**7.2628**
Our method (4.3) using $H_3(\tau)$ ($\alpha = 1, \beta_1 = -1/2$)	0.1277e−2	0.1044e−20	0.2518e−147	7.0006

Bold face numbers denote the least error among the displayed methods in case of convergence to the desired root

values of M and e has been performed in [26]. Therefore, we take values $M = 0.01$ and $e = 0.9995$ that the authors consider out of the limit for getting convergence with Newton's method. In this case the solution is $\xi = 0.3899777749463621$. Also, test function f_1 is also included to show that our proposed methods are also applicable for *non-smooth functions*. The errors $|x_n - \xi|$ of approximations to the corresponding zeros of test functions and computational order of convergence ρ_c are displayed in Tables 1, 2, 3, 4, 5, 6, 7 and 8, where $A(-h)$ denotes $A \times 10^{-h}$. All the numerical results are calculated by taking initial estimates $\gamma_0 = \lambda_0 = -0.01$ in our schemes.

Table 6 $f_6(x) = e^{x^3-x} - \cos(x^2 - 1) + x^3 + 1,\ \xi = -1,\ x_0 = -1.5$

With memory methods	$\lvert x_1 - \xi \rvert$	$\lvert x_2 - \xi \rvert$	$\lvert x_3 - \xi \rvert$	ρ_c
CLBT7 (5.4) ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.2225e−2	0.8385e−17	0.1956e−120	7.1850
D7(5.5) using $g(t) = 1 + t$, ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.2380e−2	0.1135e−16	0.1634e−119	7.1810
D7(5.5) using $g(t) = 1/(1 - t)$, ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.2303e−2	0.9816e−17	0.5933e−120	7.1829
Our method (4.3) using $H_1(\tau)$ ($\alpha = 1/2$)	0.2215e−2	0.8249e−17	0.1757e−120	7.1852
Our method (4.3) using $H_1(\tau)$ ($\alpha = 1$)	0.2172e−2	0.7569e−17	0.9629e−121	7.1863
Our method (4.3) using $H_2(\tau)$ ($\alpha = 1/2$)	0.1023e−2	0.1587e−17	0.1391e−125	7.2965
Our method (4.3) using $H_2(\tau)$ ($\alpha = 1$)	**0.5126e−3**	**0.2444e−18**	**0.6158e−132**	**7.4143**
Our method (4.3) using $H_3(\tau)$ ($\alpha = 1/2, \beta_1 = -1/2$)	0.2132e−2	0.6965e−17	0.5384e−121	7.1873
Our method (4.3) using $H_3(\tau)$ ($\alpha = 1, \beta_1 = -1/2$)	0.2055e−2	0.5921e−17	0.1729e−121	7.1895

Bold face numbers denote the least error among the displayed methods in case of convergence to the desired root

Table 7 $f_7(x) = (x-1)^3 - 1,\ \xi = 2,\ x_0 = 3.5$

With memory methods	$\lvert x_1-\xi\rvert$	$\lvert x_2-\xi\rvert$	$\lvert x_3-\xi\rvert$	ρ_c
CLBT7 (5.4) ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.1991e+0	0.6470e−6	0.5272e−44	6.6595
D7(5.5) using $g(t) = 1+t$, ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.2929e+0	0.4546e−5	0.4456e−38	6.6952
D7(5.5) using $g(t) = 1/(1-t)$, ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.2683e+0	0.2839e−5	0.1652e−39	6.7301
Our method (4.3) using $H_1(\tau)$ ($\alpha = 1/2$)	0.1834e+0	0.3130e−6	0.3269e−46	6.8401
Our method (4.3) using $H_1(\tau)$ ($\alpha = 1$)	**0.1016e+0**	**0.1950e−7**	**0.1191e−54**	**7.0765**
Our method (4.3) using $H_2(\tau)$ ($\alpha = 1/2$)	0.2340e+0	0.4260e−7	0.7070e−53	6.6954
Our method (4.3) using $H_2(\tau)$ ($\alpha = 1$)	**0.1620e−1**	**0.5619e−15**	**0.1104e−122**	**8.0063**
Our method (4.3) using $H_3(\tau)$ ($\alpha = 1/2, \beta_1 = -1/2$)	0.1527e+0	0.1017e−6	0.1253e−49	6.8755
Our method (4.3) using $H_3(\tau)$ ($\alpha = 1, \beta_1 = -1/2$)	0.1728e+0	0.1065e−5	0.1728e−42	7.1674

Bold face numbers denote the least error among the displayed methods in case of convergence to the desired root

On the accounts of results obtained in the Tables 1–8, it can be concluded that the proposed methods are highly efficient as compared to the existing robust methods, when the accuracy is tested in the multi-precision digits. Additionally, the computational order of convergence (COC) of these methods also confirmed the above conclusions to a great extent.

Furthermore, we have also included two pathological examples (see, [27]) to show that our all proposed methods (2.4) and (4.3) will converge to the complex root without having to start with a complex number.

Example 1 $g_1(x) = x^4 + 4x^3 + 9x^2 + 4x + 8.$

Table 8 $f_3(x) = \sin(\pi x)e^{x^2+x\cos x-1} + x\log(x\sin x + 1),\ \xi = 0,\ x_0 = 0.50$

With memory methods	$\lvert x_1-\xi\rvert$	$\lvert x_2-\xi\rvert$	$\lvert x_3-\xi\rvert$	ρ_c
CLBT7 (5.4) ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.8121e−2	0.1820e−14	0.4854e−102	6.9211
D7(5.5) using $g(t) = 1+t$, ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.8214e−2	0.1755e−14	0.4063e−102	6.9146
D7(5.5) using $g(t) = 1/(1-t)$, ($\gamma_0 = -0.01, \lambda_0 = -0.01$)	0.8173e−2	0.1761e−14	0.4165e−102	6.9160
Our method (4.3) using $H_1(\tau)$ ($\alpha = 1/2$)	0.8111e−2	0.1769e−14	0.4294e−102	6.9179
Our method (4.3) using $H_1(\tau)$ ($\alpha = 1$)	0.8077e−2	0.1773e−14	0.4357e−102	6.9190
Our method (4.3) using $H_2(\tau)$ ($\alpha = 1/2$)	**0.8770e−2**	**0.2518e−14**	**0.5618e−104**	**7.1459**
Our method (4.3) using $H_2(\tau)$ ($\alpha = 1$)	0.8928e−2	0.3672e−14	0.2467e−101	7.0359
Our method (4.3) using $H_3(\tau)$ ($\alpha = 1/2, \beta_1 = -1/2$)	0.8068e−2	0.1774e−14	0.4372e−102	6.9193
Our method (4.3) using $H_3(\tau)$ ($\alpha = 1, \beta_1 = -1/2$)	0.8014e−2	0.1778e−14	0.4448e−102	6.9210

Bold face numbers denote the least error among the displayed methods in case of convergence to the desired root

The zeros here are complex and are not on the right half-plane. Starting with any real negative initial guess x_0 in (2.4) and (4.3) for any $\alpha \in \mathbb{R}\backslash\{-1\}$, we shall get a complex root. For instance, starting from the real initial guess $x_0 = -2.5$, the optimal fourth-order derivative method (2.4) using $H_1(\tau)$ for $(\alpha = 1/4, \gamma = -0.01, \lambda = -0.01)$ takes only 6 iterations to converge to the complex root $-2 - 2I$ with error in the approximation as $2.889e\text{-}359 + 5.24e\text{-}360I$ and method (2.4) using $H_3(\tau)$ for $(\alpha = 1/2, \beta_1 = -1/2, \gamma = -0.01, \lambda = -0.01)$ takes only 7 iterations to converge to the complex root $-2 - 2I$ with error in the approximation as $-2.365e\text{-}135 + 2.159e\text{-}135I$.

On the other hand, with memory method (4.3) using $H_1(\tau)$ for $(\alpha = 1/4, \gamma_0 = -0.01, \lambda_0 = -0.01)$ takes 6 iterations to converge to the complex root $-2-2I$ with error in the approximation as $2.852e\text{-}195 - 3.29e\text{-}196I$ and method (4.3) using $H_3(\tau)$ for $(\alpha = 1/2, \beta_1 = -1/2, \gamma_0 = -0.01, \lambda_0 = -0.01)$ takes 6 iterations to converge to the complex root $-2-2I$ with error in the approximation as $8.44e\text{-}122 - 1.573e\text{-}121I$. The other existing methods get no solution, no matter how many iterations are performed. This also demonstrates the advantage of our methods in finding complex roots without having to start with a complex initial guess.

Example 2 $g_2(x) = x^3 - 3x^2 + 2x + 0.4$.

In this pathological example, starting from the real initial guess $x_0 = 1.5$, the optimal fourth-order derivative method (2.4) using $H_1(\tau)$ for $(\alpha = 1/4, \gamma = -0.01, \lambda = -0.01)$ takes only 6 iterations to converge to the complex root $-2 - 2I$ with error in the approximation as $2.4263e\text{-}6 - 4.3698e\text{-}8I$ and method (2.4) using $H_3(\tau)$ for $(\alpha = 1/2, \beta_1 = -1/2, \gamma = -0.01, \lambda = -0.01)$ takes only 6 iterations to converge to the complex root $-2-2I$ with error in the approximation as $2.426e\text{-}6 - 4.3898e\text{-}8I$.

While, our new with memory scheme (4.3) using $H_1(\tau)$ for $(\alpha = 1/4, \gamma_0 = -0.01, \lambda_0 = -0.01)$ takes 6 iterations to converge to the complex root $-2-2I$ with error in the approximation as $2.426e\text{-}6 - 4.4e\text{-}8I$ and method (4.3) using $H_3(\tau)$ for $(\alpha = 1/2, \beta_1 = -1/2, \gamma_0 = -0.01, \lambda_0 = -0.01)$ takes 6 iterations to converge to the complex root $-2 - 2I$ with error in the approximation as $2.426e\text{-}6 - 4.4e\text{-}8I$. On the other hand, other existing methods fail to give complex roots starting from any real guess.

Similar numerical experiments have been carried out on variety of problems which confirm the above conclusions to a great extent. Finally, we can conclude from numerical experiments that new proposed schemes confirm the theoretical results and show consistent convergence behavior.

6 Concluding remarks

In this study, we contribute further to the development of the theory of iteration processes and propose new accurate and efficient higher-order derivative-free iterative methods with and without memory for solving nonlinear equations numerically. The significant increase in the convergence speed of the proposed methods is attained without additional functional evaluations, which points to a very high computationally efficiency. In other words, the efficiency index of the proposed family with memory is $E = 7^{1/3} \approx 1.913$, which is much better than optimal three, four and five-point methods without memory having efficiency indexes $E = 8^{1/4} \approx 1.681$, $E = 16^{1/5} \approx 1.741$, $E = 32^{1/6} \approx 1.781$, respectively. The another most striking feature of this contribution is that the proposed methods can locate the complex roots without having to start from a complex number as would be necessary with other methods. Numerical experiments and the comparison of the existing robust methods are included to confirm the theoretical results and high computational efficiency. Finally, we conclude with the remark that the presented derivative-free families of Hansen-Patrick type methods with memory would be valuable alternative for solving nonlinear equations.

References

1. Traub, J.F.: Iterative Methods for the Solution of Equations. Prentice-Hall, Englewood Cliffs (1964)
2. Weerakon, S., Fernando, T.G.I.: A variant of Newton's method with accelerated third-order convergence. Appl. Math. Lett. **13**, 87–93 (2000)
3. Amat, S., Busquier, S., Gutiérrez, J.M.: Geometric constructions of iterative functions to solve nonlinear equations. J. Comput. Appl. Math. **157**, 197–205 (2003)
4. Hansen, E., Patrick, M.: A family of root finding methods. Numer. Math. **27**, 257–269 (1977)
5. Kung, H.T., Traub, J.F.: Optimal order of one-point and multi-point iteration. J. Assoc. Comput. Math. **21**, 643–651 (1974)
6. Ostrowski, A.M.: Solutions of Equations and System of Equations. Academic Press, New York (1960)
7. King, R.F.: A family of fourth order methods for nonlinear equations. SIAM J. Numer. Anal. **10**, 876–879 (1973)
8. Jarratt, P.: Some efficient fourth-order multipoint methods for solving equations. BIT **9**, 119–124 (1969)
9. Sharma, J.R., Guha, R.K., Sharma, R.: Some variants of Hansen-Patrick method with third and fourth order convergence. Appl. Math. Comput. **214**, 171–177 (2009)
10. Kansal, M., Kanwar, V., Bhatia, S.: New modifications of Hansen-Patrick's family with optimal fourth and eighth orders of convergence. Appl. Math. Comput. **269**, 507–519 (2015)
11. Ortega, J.M., Rheinboldt, W.C.: Iterative Solutions of Nonlinear Equations in Several Variables. Academic Press, New York (1970)
12. Petković, M.S., Džunić, J., Petković, L.D.: A family of two-point methods with memory for solving nonlinear equations. Appl. Anal. Discrete Math. **5**, 298–317 (2011)
13. Sharifi, S., Siegmund, S., Salimi, M.: Solving nonlinear equations by a derivative-free form of the King's family with memory. Calcolo. doi:10.1007/s10092-015-0144-1
14. Sharma, J.R., Guha, R.K., Gupta, P.: Some efficient derivative free methods with memory for solving nonlinear equations. Appl. Math. Comput. **219**, 699–707 (2012)
15. Džunić, J., Petković, M.S., Petković, L.D.: Three-point methods with and without memory for solving nonlinear equations. Appl. Math. Comput. **218**, 4917–4927 (2012)
16. Cordero, A., Fardi, M., Ghasemi, M., Torregrosa, J.R.: Accelerated iterative methods for nding solutions of nonlinear equations and their dynamical behavior. Calcolo **51**, 17–30 (2014)
17. Cordero, A., Torregrosa, J.R., Vassileva, M.P.: Three-step iterative methods with optimal eighth-order convergence. J. Comput. Appl. Math. **235**, 3189–3194 (2011)
18. Andreu, C., Cambil, N., Cordero, A., Torregrosa, J.R.: A class of optimal eighth-order derivative free methods for solving the Danchick-Guass problem. Appl. Math. Comput. **232**, 237–246 (2014)
19. Dzunic, J., Petkovic, M.S.: On generalized multipoint root-solvers with memory. J. Comput. Appl. Math. **236**, 2909–2920 (2012)
20. Hazrat, R.: Mathematica: A Problem-Centered Approach. Springer, New York (2010)
21. Zheng, Q., Li, J., Huang, F.: An optimal Steffensen-type family for solving nonlinear equations. Appl. Math. Comput. **217**, 9592–9597 (2011)
22. Soleymani, F., Sharma, R., Li, X., Tohidi, E.: An optimized derivative-free form of the Potra-Pták method. Math. Comput. Modell. **56**, 97–104 (2012)

23. Cordero, A., Lotfi, T., Bakhtiari, P., Torregrosa, J.R.: An efficient two-parametric family with memory for nonlinear equations. Numer. Algoritm. **68**, 323–335 (2015)
24. Dzunic, J.: On efficient two-parameter methods for solving nonlinear equations. Numer. Algoritm. **63**(3), 549–569 (2013)
25. Jay, I.O.: A note on Q-order of convergence. BIT Numer. Math. **41**, 422–429 (2011)
26. Danby, J.M.A., Burkardt, T.M.: The solution of Kepler's equation. I. Celest. Mech. **31**, 95–107 (1983)
27. Otolorin, O.: A new Newton-like iterative method for roots of analytic functions. Int. J. Math. Ed. Sci. Tech. **36**, 539–572 (2005)