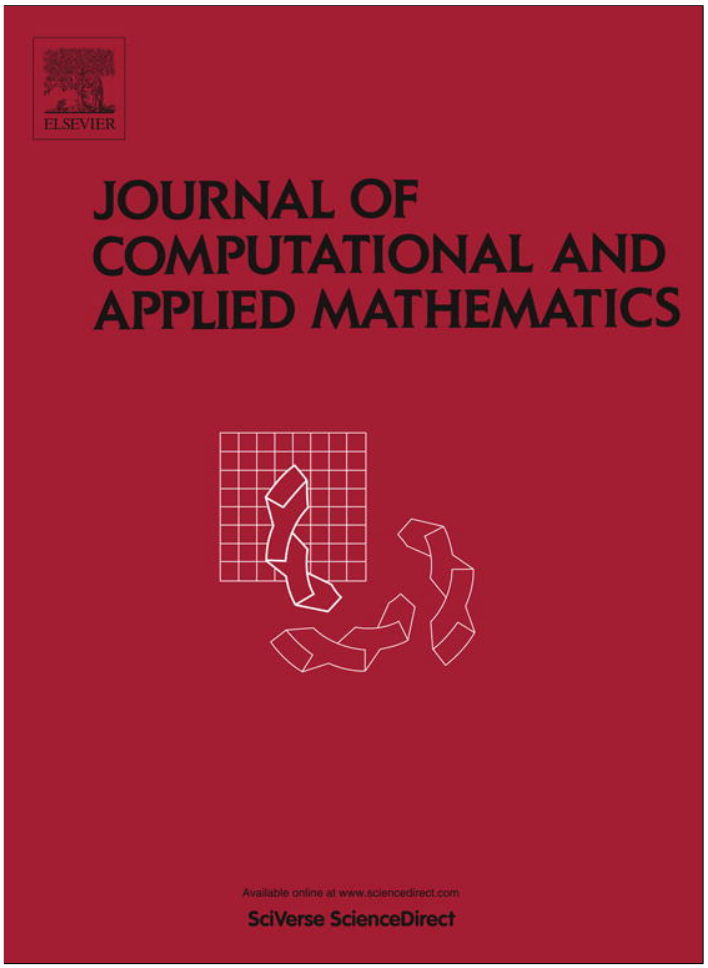

Journal of Computational and Applied Mathematics 255 (2014) 362–375

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

On generalized biparametric multipoint root finding methods with memory☆

Jovana Džunić, Miodrag S. Petković*

Faculty of Electronic Engineering, Department of Mathematics, University of Niš, P.O. Box 73, 18000 Niš, Serbia

ARTICLE INFO

Article history:
Received 14 March 2013
Received in revised form 20 May 2013

MSC:
65H05

Keywords:
Nonlinear equations
Root-finders
Multipoint methods
Convergence acceleration
Computational efficiency

ABSTRACT

A general family of biparametric n-point methods with memory for solving nonlinear equations is proposed using an original accelerating procedure with two parameters. This family is based on derivative free classes of n-point methods without memory of interpolatory type and Steffensen-like method with two free parameters. The convergence rate of the presented family is considerably increased by self-accelerating parameters which are calculated in each iteration using information from the current and previous iteration and Newton's interpolating polynomials with divided differences. The improvement of convergence order is achieved without any additional function evaluations so that the proposed family has a high computational efficiency. Numerical examples are included to confirm theoretical results and demonstrate convergence behavior of the proposed methods.

1. Introduction

The main drawbacks of one-point methods for solving nonlinear equations of the form $f(x) = 0$, such as Newton's, Halley's and Laguerre's methods, are their theoretical limits related to the convergence order, computational and informational efficiency. To overcome these disadvantages, so-called *multipoint methods* were constructed in the second half of the twentieth century. One of the fundamental papers from that period, written by Kung and Traub [1], presented two families of multipoint methods with arbitrary order of convergence of the form 2^n, which require exactly $n + 1$ function evaluations (FE for short). According to the conjecture presented in the same paper [1], proved for some classes of multipoint methods by Woźniakowski [2], the bound 2^n cannot be exceeded without additional information. This bound, often called *optimal order*, is concerned with multipoint methods that use only information from the current iteration, referred to as *methods without memory*. Further advance in designing multipoint optimal methods have become possible with significant progress of computer hardware (powerful processors) and software (symbolic computation and multi-precision arithmetics). Indeed, a classic pencil-and-paper fashion was not sufficient to design and analyze methods of higher order. For this reason, new multipoint methods of high computational efficiency have appeared in very recent years; see, e.g., [3–17] and [18].

In this paper we use optimal multipoint methods without memory as the base for constructing considerably faster methods employing information from the current and previous iteration. Following Traub's classification (see [19, pp. 8–9]), this class of root-finders are called *methods with memory*. Surprisingly enough, although methods with memory possess very high computational efficiency, they were considered very seldom in the literature. Neta's method [20] and recent results presented in [21–24] offer an advance in this topic.

☆ This work was supported by the Serbian Ministry of Science under grant 174022.
* Corresponding author. Tel.: +38118529105.
E-mail addresses: msp@eunet.rs, msp@junis.ni.ac.rs (M.S. Petković).

0377-0427/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cam.2013.05.013

Our main goal is to present a general approach to the construction and convergence analysis of n-point methods with memory. The acceleration of convergence rate is attained by suitable variation of one or two free parameters in each iterative step. These self-accelerating parameters are calculated using information from the current and previous iteration by applying Newton's interpolating polynomials. Since considerable acceleration of convergence is obtained without additional FE, the computational efficiency of improved multipoint methods is significantly increased.

The paper is organized as follows. In Section 2 we present a general family of derivative free n-point methods based on the Steffensen-like iteration function [25]

$$\phi(x) = x - \frac{f(x)}{f[x, x+\gamma f(x)] + pf(x+\gamma f(x))}$$

with real parameters γ ($\neq 0$) and p, where $f[x, y] = (f(x) - f(y))/(x - y)$ denotes a divided difference. The presented family of Steffensen's type is of interpolatory type in a wide sense (it can relay on derivative estimate) and has the order of convergence at most 2^n costing $n+1$ function evaluations. The presented procedure of derivative estimation can successfully be applied to any derivative free $(n-1)$-step optimal method to obtain an n-step optimal root-finder. The derived expression of the asymptotic error constant, depending on the parameters γ and p, gives a clear motivation for the acceleration of n-point methods. Some estimates necessary for the convergence analysis of the family with memory are presented in Section 3. In Sections 4 and 5 we study the convergence rate of the proposed family with memory attained by varying self-accelerating parameters γ and p. An appropriate calculation of γ (with fixed p) in each iterative step provides maximal order $1.5 \cdot 2^n$, while a combined variation of both parameters γ and p gives maximal order $1.75 \cdot 2^n$, which is the order improvement of 50% and 75%, respectively, relative to the n-point families without memory.

Two families based on the methods investigated in [15,25], rising from the general scheme and their particular cases, are also presented in Section 6. A discussion of computational efficiency and numerical examples are given in Section 7.

2. Two-parameter n-point methods without and with memory

Let α be a simple real zero of a real function $f : D \subset \boldsymbol{R} \to \boldsymbol{R}$ and let x_0 be an initial approximation to α. Normalized coefficients of Taylor series of the function f will be denoted with

$$c_j = \frac{f^{(j)}(\alpha)}{j!f'(\alpha)}, \quad j = 2, 3, \ldots.$$

To avoid higher order terms in some relations, which do not influence the convergence order, we employ the O- and o-notation. If $\{\varphi_k\}$ and $\{\omega_k\}$ are null sequences and $\varphi_k/\omega_k \to C$ when $k \to \infty$, where C is a nonzero constant, we shall write

$$\varphi_k = O(\omega_k) \quad \text{or} \quad \varphi_k \sim C\omega_k.$$

If $\varphi_k/\omega_k \to 0$ when $k \to \infty$, we write $\varphi_k = o(\omega_k)$; in other words, φ is dominated by ω asymptotically. This approach significantly simplifies both the convergence analysis and presentation.

In this paper we consider biparametric multipoint methods in a general form

$$\begin{cases} y_{k,1} = \varphi_1(f)(x_k) = x_k + \gamma f(x_k), \\ y_{k,2} = \varphi_2(f)(x_k) = x_k - \dfrac{f(x_k)}{f[x_k, y_{k,1}] + pf(y_{k,1})}, \\ y_{k,j} = \varphi_j(f)(x_k), \quad j = 3, \ldots, n, \\ x_{k+1} = y_{k,n+1} = \varphi_{n+1}(f)(x_k), \quad k = 0, 1, \ldots, \end{cases} \tag{1}$$

where $\gamma \neq 0$ and p are real parameters. The first two steps of the iterative scheme (1) define the two-parameter Steffensen-like method, investigated in [25]

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, x_k+\gamma f(x_k)] + pf(x_k+\gamma f(x_k))}, \quad k = 0, 1, \ldots. \tag{2}$$

The next $n-1$ steps $y_{k,j} = \varphi_j(f)(x_k)$, $j = 3, \ldots, n+1$, use inverse interpolatory iteration functions

$$y_{k,j} = \varphi_j(f)(x_k) = R(0), \quad \big(R(f(y_{k,i})) = y_{k,i},\ i = 0, \ldots, j-1,\ y_{k,0} = x_k\big), \tag{3}$$

or iteration functions of the form

$$y_{k,j} = \varphi_j(f)(x_k) = y_{k,j-1} - \frac{f(y_{k,j-1})}{P_j'(y_{k,j-1}; y_{k,0}, y_{k,1}, \ldots, y_{k,j-1})}, \tag{4}$$

where $P_j(t; y_{k,0}, y_{k,1}, \ldots, y_{k,j-1})$ is Newton's interpolating polynomial or a rational interpolating function that use available FE at the points $y_{k,0}, y_{k,1}, \ldots, y_{k,j-1}$. We restrict our investigation to these two types of interpolatory functions because of

their low computational cost. Formulas (3) and (4) give explicit definitions of the iterative steps of the iteration scheme (1). From (3) and (4) it is obvious that $y_{k,j}$ depends not only on $y_{k,j-1}$ but also on all $y_{k,i}$ for $0 \le i \le j-1$. Note that any other interpolating function of the same quality (satisfying the same interpolating conditions) would give equally good results in terms of convergence of the method (1). For more details on basic interpolatory iteration functions see the book [19, Chapter 4].

Recall that the original Steffensen's method is given by the iterative formula

$$x_{k+1} = x_k - \frac{f(x_k)^2}{f(x_k + f(x_k)) - f(x_k)};$$

see [26]. It is easy to show (for details see [25]) that the error relations of the Steffensen-like method (2) are given by

$$\varepsilon_{k,1} \sim (1 + \gamma f'(\alpha))\varepsilon_k, \qquad \varepsilon_{k+1} \sim (c_2 + p)\varepsilon_k\varepsilon_{k,1} \sim (c_2 + p)(1 + \gamma f'(\alpha))\varepsilon_k^2, \tag{5}$$

where $\varepsilon_k = x_k - \alpha$, $\varepsilon_{k,1} = y_{k,1} - \alpha$. We shall see later that this error relation has a key role in accelerating convergence order of multipoint methods with memory when we use a suitable calculation of the parameters p and γ to minimize the factors $c_2 + p$ and $1 + \gamma f'(\alpha)$. However, if the parameters γ and p have constant values during the iterative process, then the order of convergence of the Steffensen-like method (2) is two and the order of the corresponding n-point method (1) cannot exceed 2^n.

The form (1) (with $p = 0$) of multipoint methods are most commonly used for solving scalar equations. Kung and Traub were the first to develop general optimal n-point families based on the inverse interpolatory polynomials; see [1]. Interpolatory iteration functions have a common error relation of the form

$$\varepsilon_{k,j} := y_{k,j} - \alpha \sim a_j \prod_{i=0}^{j-1} \varepsilon_{k,i}^{\lambda_i}, \tag{6}$$

where

$$\varepsilon_{k,0} = \varepsilon_k = x_k - \alpha, \qquad \varepsilon_{k,n+1} = \varepsilon_{k+1} = \varepsilon_{k+1,0} = x_{k+1} - \alpha$$

and λ_i is the number of information on the function f $(f, f', \ldots, f^{(\lambda_i - 1)})$ taken at the point $y_{k,i}$ (see, e.g., [19,27]). Constants a_j depend on the type of the applied interpolation.

In what follows we will assume that the information on f, used in the iterative scheme (1), is $f(y_{k,j})$ $(j = 0, \ldots, n)$, that is, $\lambda_j = 1$ for any j. Such an information-sample procedure is one of the two proved to give the optimal order of convergence for methods with the error relation (6); see [1]. Thus (6) for the case $\lambda_j = 1$ comes down to

$$\varepsilon_{k,j} := y_{k,j} - \alpha \sim a_j \prod_{i=0}^{j-1} \varepsilon_{k,i}. \tag{7}$$

Now we will show that general interpolatory type iterations (4) preserve error relation form (7). The relation (7) holds for $j = 1, 2$, based on (5) for $a_1 = 1 + \gamma f'(\alpha)$ and $a_2 = c_2 + p$. Assuming that (7) holds for all $1 \le i \le j-1$ and $j > 2$, let $P_j(t) = P_j(t; y_{k,0}, y_{k,1}, \ldots, y_{k,j-1})$ be a minimal degree interpolating polynomial or a rational function (restriction imposed because of their low computational complexity), that coincides with f at the points $y_{k,i}$, $i = 0, \ldots, j-1$. Due to Cauchy's mean value theorem, there exists a point ξ_t contained in the minimal interval defined by the points $y_{k,0}, y_{k,1}, \ldots, y_{k,j-1}$ such that

$$f(t) - P_j(t) = \frac{(f - P_j)^{(j)}(\xi_t)}{j!} \prod_{i=0}^{j-1}(t - y_{k,i}) \sim \frac{(f - P_j)^{(j)}(\alpha)}{j!} \prod_{i=0}^{j-1}(t - y_{k,i}) \tag{8}$$

holds. After differentiating (8) at the point $y_{k,j}$, having in mind relations (7) for $1 \le i \le j-1$, we obtain

$$\begin{aligned} P_j'(y_{k,j}) &\sim f'(y_{k,j}) - \frac{(f - P_j)^{(j)}(\alpha)}{j!} \prod_{i=0}^{j-2}(y_{k,j} - y_{k,i}) \\ &= f'(y_{k,j}) - \frac{(f - P_j)^{(j)}(\alpha)}{j!} \prod_{i=0}^{j-2}(\varepsilon_{k,j} - \varepsilon_{k,i}) \\ &\sim f'(y_{k,j}) \left(1 + (-1)^j \frac{(f - P_j)^{(j)}(\alpha)}{j! f'(\alpha)} \prod_{i=0}^{j-2} \varepsilon_{k,i}\right). \end{aligned}$$

Therefore

$$P_j'(y_{k,j}) = f'(y_{k,j}) \left(1 + O\left(\prod_{i=0}^{j-2} \varepsilon_{k,i}\right)\right). \tag{9}$$

The error estimate is given by

$$\begin{aligned}\varepsilon_{k,j} &= \varepsilon_{k,j-1} - \frac{f(y_{k,j-1})}{P_j'(y_{k,j-1})}\\ &= \varepsilon_{k,j-1} - \frac{f(y_{k,j-1})}{f'(y_{k,j-1})}\left(1 + O\left(\prod_{i=0}^{j-2}\varepsilon_{k,i}\right)\right)\\ &= \varepsilon_{k,j-1} - (\varepsilon_{k,j-1} + O(\varepsilon_{k,j-1}^2))\left(1 + O\left(\prod_{i=0}^{j-2}\varepsilon_{k,i}\right)\right).\end{aligned}$$

Hence we obtain

$$\varepsilon_{k,j} = O\left(\prod_{i=0}^{j-1}\varepsilon_{k,i}\right)$$

and conclude that (7) also holds for all j when general interpolatory type iterations (4) are applied.

Using a standard error-estimate procedure that involves Taylor's series and (7), we derive the following error relations for the n-point method (1)

$$\varepsilon_{k,j} = y_{k,j} - \alpha \sim a_j \prod_{i=0}^{j-1}\varepsilon_{k,i}, \quad j = 1, \ldots, n+1, \tag{10}$$

where

$$a_1 = 1 + \gamma f'(\alpha), \qquad a_2 = c_2 + p, \tag{11}$$

and a_j ($j \geq 3$) depend on the type of interpolation used at the j-th step. For example,

$$a_j = \begin{cases} (-1)^j c_j + c_2 a_{j-1}, & \text{for Newton's interpolating polynomial in (4) [28],}\\ (-1)^{j+1}\dfrac{\mathcal{F}^{(j)}(0)}{j!\mathcal{F}'(0)^j}, & \text{for inverse polynomial interpolatory iteration (3) [1],}\end{cases}$$

where $\mathcal{F}$ is the inverse function of f.

Using (10) we prove by induction that the following is valid:

$$\varepsilon_{k,j} \sim \left(a_j \prod_{i=1}^{j-1} a_i^{2^{j-i-1}}\right)\varepsilon_{k,0}^{2^{j-1}}.$$

Taking into account (11), from the last relation we obtain

$$\begin{cases}\varepsilon_{k,1} = y_{k,1} - \alpha \sim (1 + \gamma f'(\alpha))\varepsilon_{k,0},\\ \varepsilon_{k,2} = y_{k,2} - \alpha \sim (c_2 + p)(1 + \gamma f'(\alpha))\varepsilon_{k,0}^2,\\ \varepsilon_{k,j} = y_{k,j} - \alpha \sim B_j(c_2 + p)^{2^{j-3}}(1 + \gamma f'(\alpha))^{2^{j-2}}\varepsilon_{k,0}^{2^{j-1}}, \quad j = 3, \ldots, n+1,\end{cases} \tag{12}$$

where $B_3 = a_3$, $B_j = a_j \prod_{i=3}^{j-1} a_i^{2^{j-i-1}}$ ($j > 3$). Assuming that parameters γ and p in (1) are constants, the family of n-point methods defined by (1) can be regarded as the method without memory that requires exactly $n+1$ function evaluations. For $j = n+1$, we get from the third relation of (12)

$$\begin{aligned}\varepsilon_{k,n+1} &= \varepsilon_{k+1} = x_{k+1} - \alpha \sim B_{n+1}(c_2 + p)^{2^{n-2}}(1 + \gamma f'(\alpha))^{2^{n-1}}\varepsilon_{k,0}^{2^n}\\ &= B_{n+1}(c_2 + p)^{2^{n-2}}(1 + \gamma f'(\alpha))^{2^{n-1}}\varepsilon_k^{2^n}.\end{aligned} \tag{13}$$

Therefore, the order of the family of n-point methods (1) without memory is 2^n, which means that this family is optimal in the sense of the Kung–Traub conjecture.

In the construction of multipoint methods Hermitian type of information is often used. This type of information means that if we use the derivative $f^{(m)}(y)$ at a certain point y, then all $f^{(j)}(y)$, $0 \leq j \leq m-1$, are used as well. Note that for any multipoint scheme that consumes n FE of Hermitian type having error relations

$$\varepsilon_{k,j} = O\big(\varepsilon_k^{2^{j-1}}\big), \quad j = 1, \ldots, n,$$

a new step of the form (3) or (4) can be appended to this scheme to obtain a new iterative scheme of optimal order 2^n with one new FE $f(y_{k,n})$. Thus, by means of the steps (3) and (4) we can always construct multipoint methods of arbitrary order of convergence.

Considering the error relation (13) we observe that a suitable minimization of the factors $c_2 + p = f''(\alpha)/(2f'(\alpha)) + p$ and $1 + \gamma f'(\alpha)$ could provide the order of the method (1) greater than 2^n. However, the values $f'(\alpha)$ and $f''(\alpha)$ are not available in practice. For this reason, we are forced to use only approximations $\widetilde{f}'(\alpha) \approx f'(\alpha)$ and $\widetilde{f}''(\alpha) \approx f''(\alpha)$, calculated by available information, endeavoring to attain sufficiently good approximations of the mentioned factors, for example,

$$\frac{\widetilde{f}''(\alpha)}{2\widetilde{f}'(\alpha)} + p = O(\varepsilon_{k,0}) \quad \text{and} \quad 1 + \gamma\widetilde{f}'(\alpha) = O(\varepsilon_{k,0}).$$

Then, setting $\gamma = \gamma_k = -1/\widetilde{f}'(\alpha)$ or $p_k = -\widetilde{f}''(\alpha)/(2\widetilde{f}'(\alpha)) = -\widetilde{c}_2$, or both γ_k and p_k in (1), we achieve that the order of convergence of the modified method exceeds 2^n without using any new function evaluations. Since the information from the current and previous iteration are used, the iterative scheme (1), modified in this way, becomes a method with memory. The more accurate information is used in this procedure, the greater order of the method with memory is achieved. In this paper we will use Newton's interpolating polynomials with best available approximations (nodes) to attain as high as possible order of convergence.

Remark 1. For simplicity, in the sequel we will use the denotation $\widetilde{f}'(\alpha)$ and $\widetilde{f}''(\alpha)$ although these quantities depend on the iteration index k. Contrary, to avoid any confusion and distinguish methods with and without memory, the constants a_j will be denoted with $a_{k,j}$ always when these quantities are calculated in each iteration (methods with memory).

Remark 2. We note that a similar accelerating process was applied in [21] to the Kung–Traub family without derivative [1] (constructed by inverse interpolation) and the Zheng–Li–Huang family [28] (constructed by Newton's interpolation), but only with one parameter γ. Observe that the general iterative scheme (1) contains these families as special cases.

Our model for approximating $f'(\alpha)$ and c_2 uses Newton's interpolation with divided differences

$$\widetilde{f}'(\alpha) = N_m'(y_{k,0}) \quad \text{and} \quad \widetilde{c}_2 = \frac{N_{m+1}''(y_{k,1})}{2N_{m+1}'(y_{k,1})},$$

where

$$N_m(\tau) = N_m(\tau; y_{k,0}, y_{k-1,n-j_1}, \ldots, y_{k-1,n-j_m}),$$
$$N_{m+1}(\tau) = N_{m+1}(\tau; y_{k,1}, y_{k,0}, y_{k-1,n-j_1}, \ldots, y_{k-1,n-j_m}), \quad 0 \le j_1 < j_2 < \cdots < j_m \le n,$$

are Newton's interpolating polynomials set through $m+1$ and $m+2$ available approximations from the current and previous iteration. Evidently, the fastest acceleration will be obtained when best available approximations are used as nodes for Newton's interpolating polynomials; see the recent results presented in [21–23]. For this reason we will restrict our consideration to the case

$$N_m(\tau) = N_m(\tau; y_{k,0}, y_{k-1,n}, \ldots, y_{k-1,n-m+1}), \tag{14}$$
$$N_{m+1}(\tau) = N_{m+1}(\tau; y_{k,1}, y_{k,0}, y_{k-1,n}, \ldots, y_{k-1,n-m+1}) \tag{15}$$

for $m \le n+1$. Therefore, the formulas for calculating γ_k and p_k are given by

$$\gamma_k = -\frac{1}{N_m'(y_{k,0})}, \quad m \ge 1, \tag{16}$$

$$p_k = -\frac{N_{m+1}''(y_{k,1})}{2N_{m+1}'(y_{k,1})}, \quad m \ge 1, \tag{17}$$

where N_m and N_{m+1} are defined by (14) and (15), respectively.

Now we replace constant parameters γ and p in the iterative formula (1) by the varying γ_k and p_k defined by (16) and (17). Then the family of n-point methods with memory, following from (1), becomes

$$\begin{cases} y_{k,1} = x_k + \gamma_k f(x_k), \\ y_{k,2} = x_k - \dfrac{f(x_k)}{f[x_k, y_{k,1}] + p_k f(y_{k,1})}, \\ y_{k,j} = \varphi_j(f)(x_k), \quad j = 3, \ldots, n, \\ x_{k+1} = y_{k,n+1} = \varphi_{n+1}(f)(x_k), \quad k = 0, 1, \ldots. \end{cases} \tag{18}$$

According to (10), we can write the following error relations for the family of n-point methods with memory (18)

$$\begin{cases} \varepsilon_{k,1} = y_{k,1} - \alpha \sim (1 + \gamma_k f'(\alpha))\varepsilon_{k,0}, \\ \varepsilon_{k,2} = y_{k,2} - \alpha \sim (c_2 + p_k)\varepsilon_{k,0}\varepsilon_{k,1}, \\ \varepsilon_{k,j} = y_{k,j} - \alpha \sim a_{k,j}\displaystyle\prod_{i=0}^{j-1}\varepsilon_{k,i}, \quad j = 3, \ldots, n+1, \end{cases} \tag{19}$$

where $a_{k,j} = 1 + \gamma_k f'(a)$, $a_{k,2} = c_2 + p_k$ and $a_{k,j}$ ($j \ge 3$) depends on the derivatives of f at α, γ_k and p_k, and the type of interpolation applied in (18).

3. Estimation of convergence factors

In this section we estimate the convergence factors $1+\gamma_k f'(\alpha)$ and c_2+p_k that appear in the error relation (19). These results are necessary to determine the order of convergence of the family of n-point methods (18), which is the subject of Sections 4 and 5.

Let $\{x_k\}$ be a sequence of approximations to the zero α, generated by an iterative method (IM). If this sequence converges to the zero α of f with order r, we will write

$$\varepsilon_{k+1,0}=\varepsilon_{k,n+1}\sim D_{k,n+1}\varepsilon_{k,0}^{r}, \tag{20}$$

where $D_{k,n+1}$ tends to the asymptotic error constant D_{n+1} of (IM) when $k\to\infty$. Similar to (20), we have

$$\varepsilon_{k,j}\sim D_{k,j}\varepsilon_{k,0}^{r_j},\quad 1\le j\le n, \tag{21}$$

for the iterative sequence $\{y_{k,j}\}$. Replacing relations (20) and (21) into the third relation of (19) leads to

$$\varepsilon_{k,j}\sim a_{k,j}\prod_{i=0}^{j-1}\varepsilon_{k,i}\sim a_{k,j}\varepsilon_{k,0}\prod_{i=1}^{j-1}D_{k,i}\varepsilon_{k,0}^{r_i}\sim A_{k,j}\varepsilon_{k,0}^{1+r_1+\cdots+r_{j-1}},\quad 3\le j\le n+1, \tag{22}$$

where we set $A_{k,j}=a_{k,j}\prod_{i=1}^{j-1}D_{k,i}$.

After equating exponents of the error $\varepsilon_{k,0}$ in pairs of relations (20) $\wedge$ (22) for $j=n+1$ and (21) $\wedge$ (22) for $3\le j\le n$, we obtain the system of equations

$$\begin{cases} r=1+r_1+r_2+\cdots+r_n,\\ r_j=1+r_1+r_2+\cdots+r_{j-1},\quad 3\le j\le n,\end{cases} \tag{23}$$

which results in

$$\begin{cases} r=2r_n=2^{n-2}r_3=2^{n-2}(1+r_1+r_2),\\ r_j=2r_{j-1}=2^{j-3}r_3=2^{j-3}(1+r_1+r_2),\quad 3\le j\le n,\end{cases} \tag{24}$$

for $n\ge 2$. In particular, when $n=2$ then $r=r_3$.

Orders r_1 and r_2 are directly influenced by the varying parameters γ_k and p_k and the rest of convergence analysis will be focused on determining r_1 and r_2 that would give the sought order r in the end.

In order to determine orders r_1 and r_2 we use the first two relations of (19) and the estimates of the factors $1+\gamma_k f'(\alpha)$ and c_2+p_k. According to (16) and (17), both of these factors involve Newton's interpolation. The error relation (8) becomes

$$f(t)-N_s(t)=\frac{f^{(s+1)}(\zeta)}{(s+1)!}\prod_{j=0}^{s}(t-t_j) \tag{25}$$

for Newton's interpolating polynomial of degree s, set through the nodes $t_0,\dots,t_s$, where ζ is some point from the interval defined by the interpolating nodes. The relation (25) is the basis for our error analysis.

Estimation of $1+\gamma_k f'(\alpha)$. We estimate the polynomial (14) for $1\le m\le n+1$. After differentiating (25) at the point $t=y_{k,0}$, where $s=m$ and $t_j=y_{k-1,n+1-j}$, $j=0,\dots,m$, we obtain by Taylor's expansion about the zero α

$$\begin{aligned} N_m'(y_{k,0}) &\sim f'(y_{k,0})+(-1)^{m+1}\frac{f^{(m+1)}(\alpha)}{(m+1)!}\prod_{j=0}^{m-1}\varepsilon_{k-1,n-j}\\ &\sim f'(\alpha)\left(1+c_2\varepsilon_{k,0}+(-1)^{m+1}c_{m+1}\prod_{j=0}^{m-1}\varepsilon_{k-1,n-j}\right),\end{aligned}$$

since $\varepsilon_{k,0}=O\left(\prod_{j=0}^{n}\varepsilon_{k-1,j}\right)$. Hence

$$\begin{aligned} 1+\gamma_k f'(\alpha) &= 1-\frac{f'(\alpha)}{N_m'(y_{k,0})}\sim 1-\frac{f'(\alpha)}{f'(\alpha)\left(1+c_2\varepsilon_{k,0}+(-1)^{m+1}c_{m+1}\prod_{j=0}^{m-1}\varepsilon_{k-1,n-j}\right)}\\ &\sim (-1)^{m+1}c_{m+1}\prod_{j=0}^{m-1}\varepsilon_{k-1,n-j}+c_2\varepsilon_{k,0}.\end{aligned} \tag{26}$$

According to (26), (19) and (21), and taking $r_0=1$, we estimate

$$1+\gamma_k f'(\alpha)\sim L_m\varepsilon_{k-1,0}^{r_{n-m+1}+\cdots+r_n},\quad 1\le m\le n+1, \tag{27}$$

where $L_m = (-1)^{m+1} c_{m+1} \prod_{j=1}^{m-1} D_{k-1,n-j}$ for $1 \le m \le n$ and

$$L_{n+1} = \left((-1)^n c_{n+2} + c_2 a_{k-1,n+1}\right) \prod_{j=1}^{n} D_{k-1,j}.$$

Estimation of $c_2 + p_k$. Consider now the interpolating polynomial (15). Differentiating twice (25) (for $s = m+1$) at the point $t = y_{k,1}$, we obtain

$$\begin{aligned} f'(y_{k,1}) - N'_{m+1}(y_{k,1}) &= \frac{f^{(m+2)}(\zeta_{k,m+1})}{(m+2)!}(y_{k,1} - y_{k,0}) \prod_{j=0}^{m-1}(y_{k,1} - y_{k-1,n-j}) \\ &\sim (-1)^{m+1} \frac{f^{(m+2)}(\alpha)}{(m+2)!} \varepsilon_{k,0} \prod_{j=0}^{m-1} \varepsilon_{k-1,n-j}, \end{aligned} \tag{28}$$

and

$$\begin{aligned} f''(y_{k,1}) - N''_{m+1}(y_{k,1}) &= \frac{2f^{(m+2)}(\zeta_{k,m+1})}{(m+2)!} \left[\prod_{j=0}^{m-1}(y_{k,1} - y_{k-1,n-j}) + (y_{k,1} - y_{k,0}) \sum_{j=0}^{m-1} \prod_{i=0, i\neq j}^{m-1} (y_{k,1} - y_{k-1,n-i}) \right] \\ &\sim (-1)^m \frac{2f^{(m+2)}(\alpha)}{(m+2)!} \prod_{j=0}^{m-1} \varepsilon_{k-1,n-j}. \end{aligned} \tag{29}$$

Using (28), (29) and Taylor's series, we derive

$$\begin{aligned} \frac{N''_{m+1}(y_{k,1})}{2N'_{m+1}(y_{k,1})} &\sim \frac{1}{2} \frac{f''(y_{k,1}) - (-1)^m \frac{2f^{(m+2)}(\alpha)}{(m+2)!} \prod_{j=0}^{m-1} \varepsilon_{k-1,n-j}}{f'(y_{k,1}) - (-1)^{m+1} \frac{f^{(m+2)}(\alpha)}{(m+2)!} \varepsilon_{k,0} \prod_{j=0}^{m-1} \varepsilon_{k-1,n-j}} \\ &\sim \frac{1}{2} \frac{f''(\alpha)\left(1 + \frac{3c_3}{c_2}\varepsilon_{k,1}\right) - (-1)^m \frac{2f^{(m+2)}(\alpha)}{(m+2)!} \prod_{j=0}^{m-1} \varepsilon_{k-1,n-j}}{f'(\alpha)\left(1 + 2c_2\varepsilon_{k,1}\right) - (-1)^{m+1} \frac{f^{(m+2)}(\alpha)}{(m+2)!} \varepsilon_{k,0} \prod_{j=0}^{m-1} \varepsilon_{k-1,n-j}} \\ &\sim \frac{f''(\alpha)}{2f'(\alpha)} \left(1 - (-1)^m \frac{c_{m+2}}{c_2} \prod_{j=0}^{m-1} \varepsilon_{k-1,n-j}\right). \end{aligned}$$

Hence, we estimate

$$c_2 + p_k = \frac{f''(\alpha)}{2f'(\alpha)} - \frac{N''_{m+1}(y_{k,1})}{2N'_{m+1}(y_{k,1})} \sim (-1)^m \frac{c_{m+2}}{c_2} \prod_{j=0}^{m-1} \varepsilon_{k-1,n-j}. \tag{30}$$

From (10), (21) and (30) it follows

$$c_2 + p_k \sim K_m \varepsilon_{k-1,0}^{r_{n-m+1} + \cdots + r_n}, \quad 1 \le m \le n+1, \tag{31}$$

where

$$K_m = (-1)^m \frac{c_{m+2}}{c_2} \prod_{j=0}^{m-1} D_{k-1,n-j}, \quad 1 \le m \le n+1, \text{ and } D_{k-1,0} = 1,\ r_0 = 1.$$

From (27) and (31) we conclude that the factors $1 + \gamma_k f'(\alpha)$ and $c_2 + p_k$ are of the same order of accuracy, that is

$$1 + \gamma_k f'(\alpha) \sim L_m \varepsilon_{k-1,0}^{\rho_m}, \tag{32}$$

$$c_2 + p_k \sim K_m \varepsilon_{k-1,0}^{\rho_m}, \tag{33}$$

where, in view of (23) for $n \ge 2$,

$$\rho_m = r_{n-m+1} + \cdots + r_n = \begin{cases} r - r_{n-m+1}, & 1 \le m < n, \\ r - 1, & m = n, \\ r, & m = n+1. \end{cases} \tag{34}$$

4. Order of convergence: variation of γ_k

First we will analyze the influence of the variable parameter γ_k to the convergence rate of the family (18) of n-point methods when $p_k = p$ is fixed. Since the term $c_2 + p$ is not relevant to the order of convergence of (18), in this case, with regard to (19), we associate particular relation

$$r_2 = r_1 + 1 \tag{35}$$

with (23) so that the relations (24) become

$$r = 2^{n-1}(1 + r_1) \quad \text{and} \quad r_j = 2^{j-2}(1 + r_1), \quad 2 \le j \le n,\ n \ge 1. \tag{36}$$

Combining (20) and (21) for $j = 1$ gives the estimate

$$\varepsilon_{k,1} \sim D_{k,1} D_{k-1,n+1}^{r_1} \varepsilon_{k-1,0}^{rr_1}. \tag{37}$$

Taking into account (32), (20) and the first relation of (19), we obtain

$$\varepsilon_{k,1} \sim L_m \varepsilon_{k-1,0}^{\rho_m} D_{k-1,n+1} \varepsilon_{k-1,0}^{r} \sim L_m D_{k-1,n+1} \varepsilon_{k-1,0}^{\rho_m + r}. \tag{38}$$

Equating exponents of the error $\varepsilon_{k-1,0}$ in the relations (37) and (38), with regard to (34), gives

$$rr_1 = r + \rho_m = \begin{cases} 2r - r_{n-m+1}, & 1 \le m < n, \\ 2r - 1, & m = n, \\ 2r, & m = n + 1. \end{cases}$$

According to the last relation and (36), we find

$$r_1 = \begin{cases} 2 - 2^{-m}, & 1 \le m < n, \\ \dfrac{1}{2}\left(1 + 2^{-n/2}\sqrt{9 \cdot 2^n - 8}\right), & m = n, \\ 2, & m = n + 1. \end{cases} \tag{39}$$

In this manner, from (39) and (36) we obtain the order of convergence of the n-point method with memory (18) for $n \ge 1$

$$r = 2^{n-1}(1 + r_1) = \begin{cases} 3 \cdot 2^{n-1} - 2^{n-m-1}, & 1 \le m < n, \\ 3 \cdot 2^{n-2} + 2^{\frac{n}{2}-2}\sqrt{9 \cdot 2^{2n} - 8}, & m = n, \\ 2^n + 2^{n-1} = 1.5 \cdot 2^n, & m = n + 1. \end{cases} \tag{40}$$

From the third relation of (40) we note that the improvement of convergence order of (18) is even 50% related to the order of the method without memory (1).

We summarize our results in the following theorem.

Theorem 1. *If an initial approximation x_0 is sufficiently close to a simple zero α of a function f, then the convergence order of the family of n-point methods with memory (18) with the varying γ_k, calculated by (16), is given by (40) for $1 \le m \le n + 1$.*

5. Order of convergence: variation of both γ_k and p_k

Relations (19), (23) and (24) will still make the crucial part of our convergence analysis when both parameters γ_k and p_k in (18) vary as the iteration proceeds. Recall that the factors $1 + \gamma_k f'(\alpha)$ and $c_2 + p_k$ are of the same order of accuracy $\varepsilon_{k-1,0}^{\rho_m}$ (see (32), (33)), where ρ_m is defined by (34). Hence,

$$\varepsilon_{k,1} = O\big(\varepsilon_{k-1,0}^{\rho_m} \varepsilon_{k,0}\big) \quad \text{and} \quad \varepsilon_{k,2} = O\big(\varepsilon_{k-1,0}^{\rho_m} \varepsilon_{k,1} \varepsilon_{k,0}\big) = O\big(\varepsilon_{k,1}^2\big), \tag{41}$$

yielding $r_2 = 2r_1$ and $r_3 = 1 + 3r_1$. Therefore, Eqs. (24) for $n \ge 2$ become

$$r = 2^{n-2}(1 + 3r_1) \quad \text{and} \quad r_j = 2^{j-3}(1 + 3r_1), \quad 3 \le j \le n. \tag{42}$$

As in the previous section, equating exponents of the error $\varepsilon_{k-1,0}$ in the relations (37) and (38) and having in mind (42), we obtain

$$r = \begin{cases} 2^n + 2^{n-1} + 2^{n-2} - 3 \cdot 2^{n-m-2} = 2^{n-m-2}(7 \cdot 2^m - 3), & 1 \le m < n, \\ 7 \cdot 2^{n-3} + 2^{\frac{n}{2}-3}\sqrt{49 \cdot 2^n - 48}, & m = n, \\ 2^n + 2^{n-1} + 2^{n-2} = 1.75 \cdot 2^n, & m = n + 1,\ n \ge 2. \end{cases} \tag{43}$$

We observe from the third relation of (43) that the improvement of convergence order of the family with memory (18) is even 75% related to the order of the method without memory (1).

Above results can be summarized in the following theorem.

Theorem 2. *Let x_0 be an initial approximation sufficiently close to a simple zero α of a function f. Then the convergence order of the family of n-point methods ($n \ge 2$) with memory (18) with the varying γ_k and p_k, calculated by (16) and (17), is given by (43) for $1 \le m \le n + 1$.*

Table 1
Order of convergence of multipoint methods.

$n \to$	2	3	4
$m = 1$			
γ_k	**5** (25%)	**10** (25%)	**20** (25%)
γ_k, p_k	**5.5** (37.5%)	**11** (37.5%)	**22** (37.5%)
$m = 2$			
γ_k	**5.65** (41.14%)	**11** (37.5%)	**22** (37.5%)
γ_k, p_k	**6.54** (63.5%)	**12.5** (56.25%)	**25** (56.25%)
$m = 3$			
γ_k	**6** (50%)	**11.66** (45.7%)	**23** (43.75%)
γ_k, p_k	**7** (75%)	**13.56** (69.5%)	**26.5** (65.625%)
$m = 4$			
γ_k		**12** (50%)	**23.66** (47.89%)
γ_k, p_k		**14** (75%)	**27.56** (72.25%)
Without memory	**4**	**8**	**16**

Table 2
The efficiency indices of multipoint methods with/without memory.

$n \to$	2	3	4
$m = 1$			
γ_k	**1.710**	**1.778**	**1.821**
γ_k, p_k	**1.765**	**1.821**	**1.856**
$m = 2$			
γ_k	**1.781**	**1.821**	**1.856**
γ_k, p_k	**1.870**	**1.880**	**1.904**
$m = 3$			
γ_k	**1.817**	**1.848**	**1.873**
γ_k, p_k	**1.913**	**1.919**	**1.926**
$m = 4$			
γ_k		**1.861**	**1.883**
γ_k, p_k		**1.934**	**1.941**
Without memory	**1.587**	**1.682**	**1.741**

Remark 3. We did not consider the case when the parameter γ is constant and p varies as the iteration proceeds. Following the presented procedure it is not difficult to show that the best improvement of convergence rate does not exceed 25% in this case, which is considerably smaller than the acceleration attained in Section 4 (varying γ_k) and Section 5 (varying both γ_k and p_k). For this reason, we omit the approach which varies only parameter p_k.

Remark 4. Note that the order improvement of 100% of multipoint methods with memory (18) is possible only if all available information from all the previous iterations are used; see [29]. The two self-accelerating parameters technique obtains 75% of improvement with only one previous iteration storage in memory. Similarly introduced additional self-accelerating parameters can give even higher acceleration in the convergence speed. If we use m parameters based on the information from the current and one previous iteration, then the highest obtainable order of convergence would be $2^n \sum_{i=0}^{m} 2^{-i}$, which is $\left(1 - 1/2^m\right)100\%$ of an improvement. However such a procedure requires higher and higher derivative estimation and excessive raise in computational cost with not much of an effect (increase in convergence order drops exponentially).

Remark 5. We emphasize that iterative formulas convenient for practical problems are those for $n = 1, 2$ and 3. Multipoint methods of very high order generate very accurate approximations to the zeros, not needed in practice. In Table 1 we give a review of the convergence rate of the considered n-point methods with and without memory, together with the percentage improvements of convergence order related to the order of the corresponding methods without memory.

6. Examples of derivative free n-point methods

In this section we give some examples of derivative free n-point methods. The same basic formulas hold for the methods with and without memory; the essential difference is that methods with memory use free parameters γ_k and p_k calculated

Table 3
Two-point methods with and without memory with $h(u,v) = 1+u+v+(u+v)^2$ and $g(u) = 1/(1-u-u^2)$ for $f(x) = e^x \sin(x) + \log(x^2+1)$, $\alpha = 0$, $x_0 = 0.3$, $\gamma_0 = 0.01$, $p_0 = 0$.

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	$\lvert x_4-\alpha\rvert$	r_c (50)
(47) with $h(u,v)$	1.57(−2)	4.93(−7)	4.41(−25)	2.83(−97)	4.000
γ_k, $m=1$	1.57(−2)	1.11(−8)	2.03(−39)	5.57(−193)	4.996
γ_k, $m=2$	1.57(−2)	7.01(−10)	8.45(−55)	7.10(−309)	5.656
γ_k, $m=3$	1.57(−2)	7.09(−10)	7.43(−54)	9.83(−318)	6.000
K–T, $n=2$	1.55(−2)	7.91(−7)	6.11(−24)	2.19(−92)	4.000
γ_k, $m=1$	1.55(−2)	1.67(−8)	2.08(−38)	6.26(−188)	5.000
γ_k, p_k $m=1$	1.55(−2)	1.29(−10)	3.94(−58)	6.49(−327)	5.657
γ_k, $m=2$	1.55(−2)	6.04(−10)	6.29(−55)	4.93(−309)	5.649
γ_k, p_k $m=2$	1.55(−2)	2.66(−12)	2.47(−80)	2.83(−526)	6.555
γ_k, $m=3$	1.55(−2)	6.13(−10)	3.03(−54)	4.46(−320)	6.000
γ_k, p_k $m=3$	1.55(−2)	1.49(−12)	1.57(−82)	2.32(−572)	7.000
(49) with $g(u)$	1.18(−2)	1.33(−7)	2.34(−27)	2.20(−106)	4.000
γ_k, $m=1$	1.18(−2)	5.25(−9)	6.94(−41)	2.57(−200)	5.001
γ_k, p_k $m=1$	1.18(−2)	2.69(−12)	8.97(−67)	4.86(−374)	5.641
γ_k, $m=2$	1.18(−2)	1.08(−10)	8.52(−59)	9.44(−331)	5.653
γ_k, p_k $m=2$	1.18(−2)	2.68(−14)	2.41(−90)	1.23(−587)	6.539
γ_k, $m=3$	1.18(−2)	1.11(−10)	1.08(−58)	8.88(−347)	6.000
γ_k, p_k $m=3$	1.18(−2)	7.47(−15)	1.33(−99)	6.97(−693)	7.000
Z–L–H, $n=2$	1.09(−2)	9.67(−8)	6.55(−28)	1.38(−108)	4.000
γ_k, $m=1$	1.09(−2)	2.01(−9)	5.26(−43)	6.46(−211)	5.000
γ_k, p_k $m=1$	1.09(−2)	1.85(−12)	1.42(−67)	1.59(−378)	5.642
γ_k, $m=2$	1.09(−2)	3.24(−11)	3.59(−61)	1.25(−343)	5.654
γ_k, p_k $m=2$	1.09(−2)	1.59(−14)	9.25(−92)	7.20(−597)	6.540
γ_k, $m=3$	1.09(−2)	3.34(−11)	3.49(−62)	4.63(−368)	6.000
γ_k, p_k $m=3$	1.09(−2)	4.11(−15)	1.99(−101)	1.18(−705)	7.000

in each iteration using (16) and (17). Aside from the Kung–Traub n-point derivative free family [1] and the Zheng–Li–Huang family [28], two new families with a rich structure will be tested in numerical examples.

Uniparametric family of n-point methods with memory. First of the new n-point families is relied on the following fourth-order two-point family of methods proposed in [15]

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f[x_k, w_k]}, \\ x_{k+1} = y_k - h(u_k, v_k)\dfrac{f(y_k)}{f[x_k, w_k]}, \end{cases} \qquad (k = 0, 1, \ldots), \tag{44}$$

where $u_k = \frac{f(y_k)}{f(x_k)}$, $v_k = \frac{f(y_k)}{f(w_k)}$, $w_k = x_k + \gamma f(x_k)$. The weight function h of two variables u and v satisfies the following conditions:

$$\begin{aligned} &h(0,0) = h_u(0,0) = h_v(0,0) = 1, \qquad h_{vv}(0,0) = 2, \\ &|h_{uu}(0,0)| < \infty, \qquad |h_{uv}(0,0)| < \infty, \end{aligned} \tag{45}$$

where subscripts represent appropriate partial derivatives with respect to u and v. It was proved in [15] that the error relation of the family of two-point methods (44) is given by

$$\begin{aligned} \varepsilon_{k+1} = x_{k+1} - \alpha = -c_2(1+\gamma f'(\alpha))^2 \Big[& c_3 + c_2^2\big(-4 + h_{uu}(0,0)/2 + h_{uv}(0,0) \\ & + (h_{uu}(0,0)/2 - 1)\gamma f'(\alpha)\big)\Big]\varepsilon_k^4 + O(\varepsilon_k^5). \end{aligned} \tag{46}$$

Several simple forms of the weight function $h(u,v)$ are given below

$$h(u,v) = 1+u+v, \qquad h(u,v) = \frac{1+u}{1-v}, \qquad h(u,v) = \frac{1}{1-u-v},$$
$$h(u,v) = (1+u)(1+v), \qquad h(u,v) = \frac{1}{(1-u)(1-v)}.$$

Table 4
Two-point methods with and without memory with $h(u,v) = 1+u+v+(u+v)^2$ and $g(u) = 1/(1-u-u^2)$ for $f(x) = x^2-(1-x)^{25}$, $\alpha = 0.1437,\ldots,$ $x_0 = 0.25$, $\gamma_0 = 0.01$, $p_0 = 0$.

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	$\lvert x_4-\alpha\rvert$	r_c (50)
(47) with $h(u,v)$	4.99(−3)	2.21(−7)	3.06(−25)	1.11(−96)	4.000
γ_k, $m=1$	4.99(−3)	2.62(−9)	1.58(−40)	4.82(−196)	4.981
γ_k, $m=2$	4.99(−3)	8.57(−10)	1.85(−48)	1.46(−265)	5.615
γ_k, $m=3$	4.99(−3)	2.37(−11)	5.81(−60)	1.85(−351)	5.997
K–T, $n=2$	3.92(−3)	1.00(−7)	4.76(−26)	2.41(−99)	4.000
γ_k, $m=1$	3.92(−3)	9.35(−10)	3.63(−42)	3.09(−204)	5.000
γ_k, p_k $m=1$	3.92(−3)	1.16(−10)	2.01(−52)	5.39(−288)	5.641
γ_k, $m=2$	3.92(−3)	2.18(−10)	8.60(−51)	9.33(−279)	5.642
γ_k, p_k $m=2$	3.92(−3)	1.66(−11)	8.56(−66)	9.55(−421)	6.538
γ_k, $m=3$	3.92(−3)	2.34(−12)	2.59(−65)	4.69(−383)	6.000
γ_k, p_k $m=3$	3.92(−3)	1.26(−13)	3.83(−85)	8.14(−586)	7.000
(49) with $g(u)$	4.35(−3)	6.11(−8)	1.85(−27)	1.56(−105)	4.000
γ_k, $m=1$	4.35(−3)	1.22(−9)	1.56(−41)	4.52(−201)	5.002
γ_k, p_k $m=1$	4.35(−3)	3.29(−10)	3.74(−50)	1.81(−275)	5.641
γ_k, $m=2$	4.35(−3)	2.88(−10)	4.32(−50)	8.63(−275)	5.642
γ_k, p_k $m=2$	4.35(−3)	4.30(−11)	1.33(−63)	5.15(−407)	6.540
γ_k, $m=3$	4.35(−3)	7.54(−13)	2.90(−68)	9.24(−401)	6.000
γ_k, p_k $m=3$	4.35(−3)	7.65(−14)	9.09(−88)	1.93(−605)	7.003
Z–L–H, $n=2$	4.28(−3)	4.86(−8)	7.13(−28)	3.28(−107)	4.000
γ_k, $m=1$	4.28(−3)	4.09(−10)	5.11(−44)	1.70(−213)	4.999
γ_k, p_k $m=1$	4.28(−3)	2.83(−10)	1.85(−50)	3.55(−277)	5.642
γ_k, $m=2$	4.28(−3)	9.00(−11)	9.81(−53)	4.46(−289)	5.632
γ_k, p_k $m=2$	4.28(−3)	3.82(−11)	7.34(−64)	1.12(−408)	6.541
γ_k, $m=3$	4.28(−3)	3.06(−13)	6.25(−71)	5.93(−417)	5.998
γ_k, p_k $m=3$	4.28(−3)	9.53(−14)	3.36(−87)	1.82(−601)	7.001

The new family of optimal n-point derivative free methods without memory, based on the two-point method (44), is given by the following iterative scheme

$$\begin{cases} y_{k,1} = x_k + \gamma f(x_k), \\ y_{k,2} = x_k - \dfrac{f(x_k)}{f[x_k, y_{k,1}]}, \\ y_{k,3} = y_{k,2} - h(u_k, v_k)\dfrac{f(y_{k,2})}{f[x_k, y_{k,1}]}, \quad u_k = \dfrac{f(y_{k,2})}{f(x_k)},\ v_k = \dfrac{f(y_{k,2})}{f(y_{k,1})}, \\ y_{k,j} = y_{k,j-1} - \dfrac{f(y_{k,j-1})}{N'_{j-1}(y_{k,j-1})}, \quad j = 4,\ldots,n+1, \\ x_{k+1} = y_{k,n+1}, \quad k = 0, 1, \ldots, \end{cases} \tag{47}$$

where

$$N_{j-1}(\tau) = N_{j-1}(\tau; y_{k,j-1}, \ldots, y_{k,1}, x_k) \tag{48}$$

is Newton's interpolating polynomial of degree $j-1$ set through the nodes $y_{k,j-1},\ldots,y_{k,1},x_k$, for $j = 4,\ldots,n+1$. When the parameter γ is fixed during the iterative process, the n-point method (47) (without memory) can be regarded as an extension of the three-point family proposed in [22].

The n-point method (47) does not involve the parameter p so that we can vary the parameter γ only using (16). In this way the uniparametric family of n-point methods with memory is obtained. Its order of convergence can be increasing up to 50% relative to the order of the family without memory (47); see Theorem 1.

Biparametric family of n-point methods with memory. Another new n-point iterative scheme uses the two-parameter Steffensen-like method and has the form

$$\begin{cases} y_{k,1} = x_k + \gamma f(x_k), \\ y_{k,2} = x_k - \dfrac{f(x_k)}{f[x_k, y_{k,1}] + pf(y_{k,1})}, \\ y_{k,3} = y_{k,2} - g(u_k)\dfrac{f(y_{k,2})}{f[y_{k,2}, y_{k,1}] + pf(y_{k,1})}, \quad u_k = \dfrac{f(y_{k,2})}{f(x_k)}, \\ y_{k,j} = y_{k,j-1} - \dfrac{f(y_{k,j-1})}{N'_{j-1}(y_{k,j-1})}, \quad j = 4,\ldots,n+1, \\ x_{k+1} = y_{k,n+1}, \quad k = 0, 1, \ldots, \end{cases} \tag{49}$$

Table 5
Three-point methods with and without memory with $h(u,v) = 1+u+v+(u+v)^2$ and $g(u) = 1/(1-u-u^2)$ for $f(x) = e^x \sin(x)+\log(x^2+1)$, $\alpha = 0$, $x_0 = 0.3$, $\gamma_0 = 0.01$, $p_0 = 0$.

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	r_c (50)
(47) with $h(u,v)$	2.59(−5)	1.30(−48)	6.41(−525)	11.000
γ_k, $m=1$	2.59(−5)	3.80(−65)	1.17(−942)	14.666
γ_k, $m=2$	2.59(−5)	2.65(−60)	9.78(−825)	13.901
γ_k, $m=3$	2.59(−5)	2.65(−66)	3.83(−981)	15.000
γ_k, $m=4$	2.59(−5)	2.84(−67)	6.41(−979)	14.714
K–T, $n=3$	8.13(−4)	2.16(−22)	5.45(−171)	7.999
γ_k, $m=1$	8.13(−4)	1.73(−28)	1.88(−275)	10.009
γ_k, p_k $m=1$	8.13(−4)	1.06(−33)	9.02(−363)	11.011
γ_k, $m=2$	8.13(−4)	5.52(−34)	1.30(−368)	11.092
γ_k, p_k $m=2$	8.13(−4)	2.47(−41)	3.37(−524)	12.870
γ_k, $m=3$	8.13(−4)	9.10(−34)	1.83(−389)	11.876
γ_k, p_k $m=3$	8.13(−4)	1.28(−41)	9.77(−566)	13.864
γ_k, $m=4$	8.13(−4)	1.38(−33)	8.37(−391)	11.999
γ_k, p_k $m=4$	8.13(−4)	3.23(−41)	7.04(−565)	14.001
(49) with $g(u)$	2.36(−4)	9.92(−28)	9.53(−215)	8.000
γ_k, $m=1$	2.36(−4)	1.21(−33)	1.31(−326)	10.002
γ_k, p_k $m=1$	2.36(−4)	1.67(−41)	4.76(−451)	11.024
γ_k, $m=2$	2.36(−4)	6.62(−42)	5.62(−455)	11.000
γ_k, p_k $m=2$	2.36(−4)	1.05(−51)	5.22(−652)	12.677
γ_k, $m=3$	2.36(−4)	5.42(−41)	1.14(−472)	11.782
γ_k, p_k $m=3$	2.36(−4)	1.27(−52)	1.72(−710)	13.629
γ_k, $m=4$	2.36(−4)	1.95(−40)	2.21(−473)	11.998
γ_k, p_k $m=4$	2.36(−4)	1.73(−51)	4.16(−713)	14.037
Z–L–H, $n=3$	2.00(−4)	2.67(−28)	2.67(−219)	8.000
γ_k, $m=1$	2.00(−4)	5.29(−35)	7.85(−341)	10.001
γ_k, p_k $m=1$	2.00(−4)	2.67(−42)	8.04(−460)	11.024
γ_k, $m=2$	2.00(−4)	1.10(−43)	1.64(−474)	10.974
γ_k, p_k $m=2$	2.00(−4)	6.91(−53)	7.40(−667)	12.669
γ_k, $m=3$	2.00(−4)	1.37(−42)	7.69(−491)	11.745
γ_k, p_k $m=3$	2.00(−4)	1.43(−49)	8.00(−635)	12.963
γ_k, $m=3$	2.00(−4)	5.74(−42)	2.07(−492)	11.998
γ_k, p_k $m=3$	2.00(−4)	1.72(−52)	3.90(−727)	14.036

where $N_{j-1}(\tau)$ is given in (48). The family (49) without memory reaches the optimal order of convergence 2^n when the two-point family

$$\begin{cases} y_{k,1} = x_k + \gamma f(x_k), \\ y_{k,2} = x_k - \dfrac{f(x_k)}{f[x_k, y_{k,1}] + pf(y_{k,1})}, \\ x_{k+1} = y_{k,2} - g(u_k)\dfrac{f(y_{k,2})}{f[y_{k,2}, y_{k,1}] + pf(y_{k,1})}, \quad u_k = \dfrac{f(y_{k,2})}{f(x_k)}, \end{cases} \qquad (k = 0, 1, \ldots)$$

is of optimal order four. This optimal order is achieved under the conditions

$$g(0) = 1, \qquad g'(0) = 1, \qquad |g''(0)| < \infty;$$

see [25].

By varying parameters γ and p in (49) using (16) and (17) we obtain the family of n-point methods with memory. The order of this family can be increased up to 75% relative to the family (49) without memory; see Theorem 2. Some examples of the weight function g of simple form and approximately of the same computational cost are as follows:

$$g(u) = \frac{1+\eta_1 u}{1+(\eta_1-1)u}, \qquad g(u) = \frac{1}{1-u+\eta_2 u^2}, \qquad g(u) = \left(1+\frac{u}{q}\right)^q,$$

where $\eta_1, \eta_2 \in \boldsymbol{R}$, $q \in \boldsymbol{Q}$. The values $q = -2, -1, 1, 2$ are more convenient values for q.

Explicit iterative formulas of the Kung–Traub family and the Zheng–Li–Huang family are given in [1,28], respectively, and also in [21]. For this reason, they will not be displayed here.

Table 6
Three-point methods with and without memory with $h(u,v) = 1+u+v+(u+v)^2$ and $g(u) = 1/(1-u-u^2)$ for $f(x) = x^2-(1-x)^{25}$, $\alpha = 0.1437,\ldots,$ $x_0 = 0.25$, $\gamma_0 = 0.01$, $p_0 = 0$.

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	r_c (50)
(47) with $h(u,v)$	8.21(−6)	9.38(−36)	2.74(−275)	8.000
γ_k, $m=1$	8.21(−6)	2.67(−41)	5.04(−400)	10.108
γ_k, $m=2$	8.21(−6)	5.52(−45)	4.13(−479)	11.082
γ_k, $m=3$	8.21(−6)	1.99(−46)	8.68(−523)	11.728
γ_k, $m=4$	8.21(−6)	4.14(−48)	4.04(−560)	12.105
K–T, $n=3$	2.15(−4)	1.38(−24)	4.25(−186)	7.999
γ_k, $m=1$	2.15(−4)	1.54(−30)	5.66(−290)	9.923
γ_k, p_k $m=1$	2.15(−4)	2.32(−33)	5.76(−349)	10.895
γ_k, $m=2$	2.15(−4)	1.60(−33)	2.25(−350)	10.877
γ_k, p_k $m=2$	2.15(−4)	8.14(−40)	4.50(−485)	12.570
γ_k, $m=3$	2.15(−4)	2.19(−34)	4.35(−381)	11.560
γ_k, p_k $m=3$	2.15(−4)	2.82(−40)	2.57(−522)	13.434
γ_k, $m=4$	2.15(−4)	5.12(−35)	3.41(−400)	11.925
γ_k, p_k $m=4$	2.15(−4)	1.74(−40)	8.52(−544)	13.945
(49) with $g(u)$	9.57(−6)	3.30(−35)	6.66(−271)	8.000
γ_k, $m=1$	9.57(−6)	4.94(−40)	5.12(−386)	10.091
γ_k, p_k $m=1$	9.57(−6)	1.39(−43)	6.23(−461)	11.030
γ_k, $m=2$	9.57(−6)	8.83(−44)	1.96(−465)	11.086
γ_k, p_k $m=2$	9.57(−6)	2.69(−49)	2.82(−602)	12.697
γ_k, $m=3$	9.57(−6)	2.87(−45)	1.23(−509)	11.749
γ_k, p_k $m=3$	9.57(−6)	1.23(−51)	9.84(−675)	13.578
γ_k, $m=4$	9.57(−6)	4.60(−47)	3.73(−547)	12.103
γ_k, p_k $m=4$	9.57(−6)	2.69(−54)	2.56(−738)	14.089
Z–L–H, $n=3$	1.12(−5)	1.10(−34)	9.57(−267)	8.000
γ_k, $m=1$	1.12(−5)	1.68(−40)	4.96(−392)	10.095
γ_k, p_k $m=1$	1.12(−5)	5.51(−43)	1.41(−454)	11.033
γ_k, $m=2$	1.12(−5)	3.20(−44)	1.16(−470)	11.064
γ_k, p_k $m=2$	1.12(−5)	8.24(−49)	3.37(−596)	12.691
γ_k, $m=3$	1.12(−5)	1.01(−45)	9.41(−515)	11.713
γ_k, p_k $m=3$	1.12(−5)	9.90(−50)	1.41(−620)	12.958
γ_k, $m=4$	1.12(−5)	1.53(−47)	1.15(−555)	12.138
γ_k, p_k $m=4$	1.12(−5)	8.18(−54)	1.49(−731)	14.080

7. Computational aspects

In this section we first consider computational efficiency of multipoint methods with memory obtained by the proposed acceleration techniques. Several values of the efficiency index

$$E(IM) = r^{1/\theta},$$

where r is the order of the considered iterative method (IM) and θ is the number of new function evaluations per iterations (see [19, p. 263]), are given in Table 2.

From Table 2 we conclude that the use of the self-accelerating parameter γ_k considerably increases computational efficiency of multipoint methods. An additional increase is attained if both parameters γ_k and p_k are calculated in each iteration. We note that two-point methods with memory, which are of the greatest practical interest, are even more efficient than the three-point optimal methods without memory. We again emphasize that great informational and computational efficiency is the main advantage of the proposed methods with memory.

We have tested particular methods of the newly proposed families (47) and (49), and members of the Kung–Traub family [1] and the Zheng–Li–Huang family [28] with and without memory for $n=2$ and $n=3$. Good initial approximations were calculated using an efficient non-iterative method based on numerical integration, given in [30]. We employed the computational software package *Mathematica* with multiple-precision arithmetic.

All numerical examples have shown exceptional convergence speed of the tested methods. The errors $\lvert x_k-\alpha\rvert$ are given in Tables 3 and 4 for the first four iterations (two-point methods) and in Tables 5 and 6 for the first three iterations (three-point methods), where the denotation $A(-h)$ means $A\times 10^{-h}$. K–T and Z–L–H are the abbreviations for the Kung–Traub method and the Zheng–Li–Huang method, respectively. These tables include the values of the computational order of convergence r_c calculated by the formula (see [24, p. 5])

$$r_c = \frac{\log\lvert f(x_k)/f(x_{k-1})\rvert}{\log\lvert f(x_{k-1})/f(x_{k-2})\rvert}, \tag{50}$$

taking into consideration the last three approximations in the iterative process. Although this formula for r_c is derived for methods without memory, it gives mainly a good estimate of convergence order for methods with memory.

From Tables 3–6 and many tested examples we can conclude that the proposed methods with memory produce approximations of great accuracy. Usually two iterations are quite sufficient for solving most real-life problems even in the case when two-point methods are applied. Results of the third and fourth iteration are given to demonstrate very fast convergence of the presented methods. The more accurate information are used in the accelerating procedure (that is, greater m), the greater order of the designed method with memory is achieved. Entries in the last column of Tables 3–6 show that the computational order of convergence r_c, calculated by (50), matches very well the theoretical order given in Theorems 1 and 2.

Remark 6. The computational order of convergence r_c perfectly coincides with the theoretical order only when initial approximations are reasonably close to the sought zeros; otherwise, the convergence at the beginning of the iterative process can be slow. For this reason, the choice of good initial approximations is of great importance in the application of iterative methods, including multipoint methods. Most authors do not discuss this important subject. We note that an efficient method for finding initial approximations of great accuracy was recently proposed in [30,31]. A combination of this localization method and fast iterative multipoint methods is the only way to achieve satisfactory results in practice.

We conclude this paper with the comment that the considerable increase of the order of convergence up to 50% (using a suitable variation of the parameter γ_k) and even up to 75% (using the self-acceleration of two parameters γ_k and p_k) is attained without any additional function evaluations per iteration. This means that the proposed methods with memory possess a very high computational efficiency, not recorded at present in theory and practice of iterative processes for solving nonlinear equations.

References

[1] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, J. ACM 21 (1974) 643–651.
[2] H. Woźniakowski, Maximal order of multipoint iterations using n evaluations, in: J.F. Traub (Ed.), Analytic Computational Complexity, Academic Press, New York, 1976, pp. 75–107.
[3] W. Bi, H. Ren, Q. Wu, Three-step iterative methods with eight-order convergence for solving nonlinear equations, J. Comput. Appl. Math. 225 (2009) 105–112.
[4] W. Bi, Q. Wu, H. Ren, A new family of eight-order iterative methods for solving nonlinear equations, Appl. Math. Comput. 214 (2009) 236–245.
[5] C. Chun, Some fourth-order iterative methods for solving nonlinear equations, Appl. Math. Comput. 195 (2008) 454–459.
[6] 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.
[7] C. Chun, B. Neta, Certain improvements of Newton's method with fourth-order convergence, Appl. Math. Comput. 215 (2009) 821–828.
[8] A. Cordero, J.R. Torregrosa, M.P. Vassileva, Three-step iterative methods with optimal eighth-order convergence, J. Comput. Appl. Math. 235 (2011) 3189–3194.
[9] Y.H. Geum, Y.I. Kim, A multi-parameter family of three-step eighth-order iterative methods locating a simple root, Appl. Math. Comput. 215 (2010) 3375–3382.
[10] Y.H. Geum, Y.I. Kim, A uniparametric family of three-step eighth-order multipoint iterative methods for simple roots, Appl. Math. Lett. 24 (2011) 929–935.
[11] G.V. Milovanović, A.S. Cvetković, A note on three-step iterative methods for nonlinear equations, Studia Univ. Babeş-Bolyai Math. 52 (2007) 37–146.
[12] B. Neta, M.S. Petković, Construction of optimal order nonlinear solvers using inverse interpolation, Appl. Math. Comput. 217 (2010) 2448–2455.
[13] M.S. Petković, On a general class of multipoint root-finding methods of high computational efficiency, SIAM J. Numer. Anal. 47 (2010) 4402–4414.
[14] M.S. Petković, J. Džunić, B. Neta, Interpolatory multipoint methods for solving nonlinear equations, Appl. Math. Comput. 218 (2011) 2533–2541.
[15] M.S. Petković, S. Ilić, J. Džunić, Derivative free two-point methods with and without memory for solving nonlinear equations, Appl. Math. Comput. 217 (2010) 1887–1895.
[16] H. Ren, Q. Wu, W. Bi, A class of two-step Steffensen's type methods with fourth-order convergence, Appl. Math. Comput. 209 (2009) 206–210.
[17] J.R. Sharma, R. Sharma, A new family of modified Ostrowski's methods with accelerated eighth order convergence, Numer. Algorithms 54 (2010) 445–458.
[18] M.S. Petković, L.D. Petković, Families of optimal multipoint methods for solving nonlinear equations: a survey, Appl. Anal. Discrete Math. 4 (2010) 1–22.
[19] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
[20] B. Neta, A new family of higher order methods for solving equations, Int. J. Comput. Math. 14 (1983) 191–195.
[21] J. Džunić, M.S. Petković, On generalized multipoint root-solvers with memory, J. Comput. Appl. Math. 236 (2012) 2909–2920.
[22] 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.
[23] M.S. Petković, J. Džunić, L.D. Petković, A family of two-point methods with memory for solving nonlinear equations, Appl. Anal. Discrete Math. 5 (2011) 298–317.
[24] M.S. Petković, B. Neta, L.D. Petković, J. Džunić, Multipoint Methods for Solving Nonlinear Equations, Elsevier, Amsterdam, Boston, Heidelberg, 2013.
[25] J. Džunić, On efficient two-parameter methods for solving nonlinear equations, Numer. Algorithms (2012) http://dx.doi.org/10.1007/s11075-012-9641-3. Published online: 07 September.
[26] I.F. Steffensen, Remarks on iteration, Skand. Aktuarietidskr. 16 (1933) 64–72.
[27] J. Herzberger, Über Matrixdarstellungen für Iterationverfahren bei nichtlinearen Gleichungen, Computing 12 (1974) 215–222.
[28] Q. Zheng, J. Li, F. Huang, Optimal Steffensen-type families for solving nonlinear equations, Appl. Math. Comput. 217 (2011) 9592–9597.
[29] R. Brent, S. Winograd, P. Wolfe, Optimal iterative processes for root-finding, Numer. Math. 80 (1973) 327–341.
[30] B.I. Yun, A non-iterative method for solving non-linear equations, Appl. Math. Comput. 198 (2008) 691–699.
[31] B.I. Yun, M.S. Petković, Iterative methods based on the signum function approach for solving nonlinear equations, Numer. Algorithms 52 (2009) 649–662.