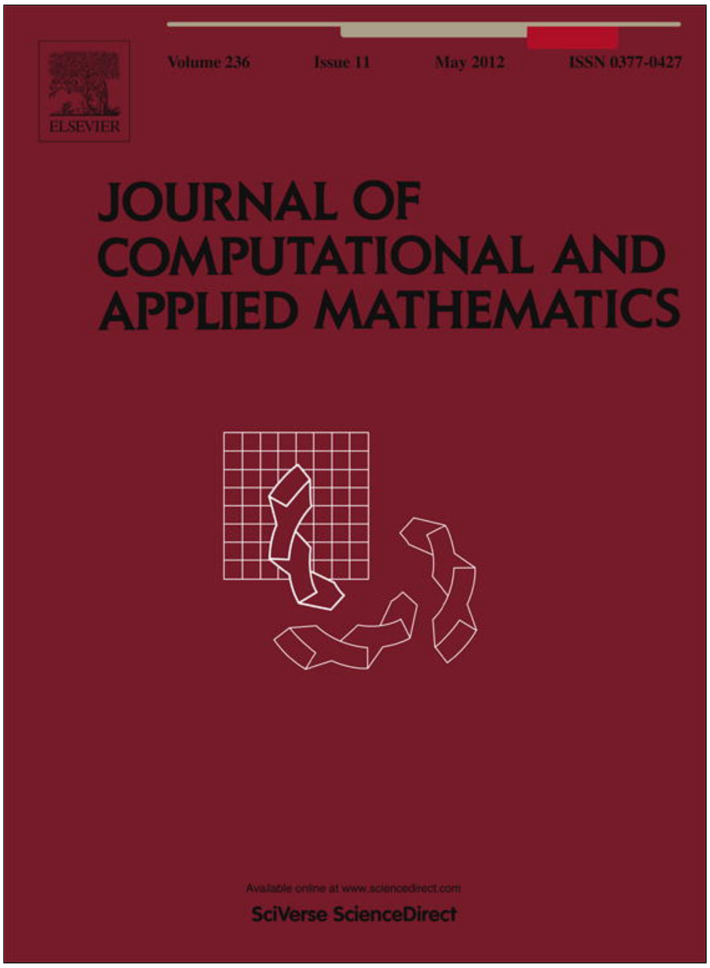
Volume 236
Issue 11
May 2012
ISSN 0377-0427
ELSEVIER
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
Available online at www.sciencedirect.com
SciVerse ScienceDirect

Journal of Computational and Applied Mathematics 236 (2012) 2909–2920

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

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

On generalized multipoint root-solvers with memory

Jovana Džunić, Miodrag S. Petković *

Faculty of Electronic Engineering, Department of Mathematics, University of Niš, 18000 Niš, Serbia

ARTICLE INFO

Article history:
Received 17 August 2011
Received in revised form 29 January 2012

MSC:
65H05

Keywords:
Nonlinear equations
Multipoint iterative methods
Iterative methods with memory
Acceleration of convergence
Computational efficiency

ABSTRACT

The improved versions of the Kung–Traub family and the Zheng–Li–Huang family of n-point derivative free methods for solving nonlinear equations are proposed. The convergence speed of the modified families is considerably accelerated by employing a self-correcting parameter. This parameter is calculated in each iteration using information from the current and previous iteration so that the proposed families can be regarded as the families with memory. The increase of convergence order is attained without any additional function evaluations meaning that these families with memory possess high computational efficiency. Numerical examples are included to confirm theoretical results and demonstrate convergence behaviour of the proposed methods.

1. Introduction

Multipoint iterative methods belong to the class of the most efficient methods for solving nonlinear equations of the form $f(x) = 0$. This class of methods was extensively studied in Traub's book [1] and some papers and books published in the second half of the twentieth century. The interest for multipoint methods has arisen in recent years mainly for two reasons: (1) root-solvers based on multipoint methods are of current interest since they overcome theoretical limits of one-point methods related to the convergence order and computational efficiency, and (2) implementation and convergence analysis of multipoint methods with the capability to generate root approximations of very high accuracy have become possible with significant progress of computer hardware (powerful processors) and software (multi-precision arithmetics, symbolic computation).

The highest possible computational efficiency of multipoint methods is closely connected to the optimality hypothesis of Kung and Traub [2] from 1974: *The order of convergence of any multipoint method without memory, based on $n + 1$ function evaluations per iteration, cannot exceed the bound 2^n*.

In this paper, we concentrate on the construction of multipoint methods with memory, a task which is seldom considered in the literature. We prove that methods with memory can achieve considerably faster convergence than the corresponding methods without memory, without additional function evaluations. In this manner the computational efficiency is significantly increased. The acceleration of convergence speed is attained by suitable variation of a free parameter in each iterative step. This *self-correcting parameter* is calculated using information from the current and the previous iteration by applying the secant-type method with "gliding" approximations and Newton's interpolating polynomials of second and third degree.

We restrict our attention to the Kung–Traub n-point family [2] and the Zheng–Li–Huang n-point family [3] for the following reasons:

(1) both families are optimal in the sense of the Kung–Traub conjecture;
(2) the order of convergence can be arbitrary high (in the form 2^n);

* Corresponding author.
E-mail addresses: msp@eunet.rs, msp@junis.ni.ac.rs (M.S. Petković).

0377-0427/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2012.01.035

(3) both families are adaptive since they allow the acceleration of convergence without altering their structure;
(4) these families are derivative free, which is convenient in all situations when the calculation of derivatives of f is complicated.

The paper is organized as follows. The Kung–Traub family [2] and the Zheng–Li–Huang family [3] of multipoint methods without memory are presented in Section 2. The key idea for the convergence acceleration, arising from the form of the expression for the asymptotic error relation, is discussed in Section 3. The families of multipoint methods with memory, based on the Kung–Traub's and Zheng–Li–Huang's n-point families, are derived through a self-correcting parameter which is calculated in each iteration using already known information from the previous and the current iteration. A similar idea was applied to the families of two-point and three-point methods in [4,5]. In Section 4, we determine the lower bound of the R-order of convergence of the proposed families with memory. A classical secant approach, exposed in Traub's book [1, p. 185–187] and extended in [6], provides the order $2^{n-1}\big(1+\sqrt{1+2^{1-n}}\,\big)$. An improved secant approach gives higher order 2^n+2^{j-1} ($j\in\{1,\ldots,n-1\}$). The application of Newton's interpolating polynomial of second degree provides even better results: the R-order is at least $11\cdot 2^{n-3}$ for $n\ge 3$, and at least $\frac{1}{2}(5+\sqrt{33})\approx 5.372$ for $n=2$. Even faster convergence is achieved by applying Newton's interpolating polynomial of third degree: the R-order is at least $23\cdot 2^{n-4}$ for $n\ge 4$, at least $\frac{1}{2}(11+\sqrt{137})\approx 11.352$ for $n=3$, and at least 6 for $n=2$. Numerical examples are given in Section 5.

2. Two families of arbitrary order without memory

Let α be a simple zero of a real function $f : D\subset \boldsymbol{R}\to\boldsymbol{R}$ and let x_0 be an initial approximation to α. In this section, we present two derivative free families of n-point methods without memory for solving nonlinear equations. Both families have similar structure, the order 2^n and require $n+1$ function evaluations per iteration, which means that they generate optimal methods in the sense of the Kung–Traub conjecture [2]. These families will be modified by our specific approach to very efficient generalized methods with memory.

Throughout this paper we often use *normalized Taylor series coefficients* for f,

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

To avoid higher order terms in some relations, which are without influence to the convergence order, we employ the notation used in Traub's book [1]: if $\{\varphi_k\}$ and $\{\omega_k\}$ are null sequences and $\varphi_k/\omega_k\to C$, where C is a nonzero constant, we shall write

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

This approach significantly simplifies both analysis and presentation.

Kung–Traub's family.

In 1974 Kung and Traub [2] stated the following derivative free family (shorter K–T family) of iterative methods without memory.

K–T family: for an initial approximation x_0, arbitrary $n\in\boldsymbol{N}$ and $k=0,1,\ldots$, define an iterative function $\psi_j(f)$ ($j=-1,0,\ldots,n$) as follows:

$$\begin{cases} y_{k,0}=\psi_0(f)(x_k)=x_k, & y_{k,-1}=\psi_{-1}(f)(x_k)=x_k+\gamma f(x_k),\quad \gamma\in\boldsymbol{R}\setminus\{0\},\\ y_{k,j}=\psi_j(f)(x_k)=\mathcal{R}_j(0), & j=1,\ldots,n,\ \text{for } n>0,\end{cases}\tag{1}$$

where $\mathcal{R}_j(t)$ is the inverse interpolating polynomial of degree at most j such that

$$\mathcal{R}_j(f(y_{k,m}))=y_{k,m},\quad m=-1,0,\ldots,j-1.$$

The Kung–Traub iterative method is defined by $x_{k+1}=y_{k,n}=\psi_n(f)(x_k)$ starting from x_0, where k is the iteration index. It was proved in [2] that the order of convergence of family (1) is 2^n ($n\ge 1$).

The approximation $y_{k,j}$ ($j<n$) at the j-th step within the k-th iteration will be called *intermediate* approximation with the associated intermediate error $\varepsilon_{k,j}=y_{k,j}-\alpha$. Following this terminology, $y_{k,n-1}$ is the *penultimate* approximation and $y_{k,n}$ ($=x_{k+1}$) is the *ultimate* approximation of the k-th iteration.

The following error relation for family (1), also called the *ultimate* error relation, was derived in [2]

$$\varepsilon_{k+1}=x_{k+1}-\alpha\sim (1+\gamma f'(\alpha))^{2^{n-1}}B_n(f)\varepsilon_k^{2^n},\tag{2}$$

where

$$B_n(f)=\Upsilon_n(f)\prod_{j=1}^{n-1}\Upsilon_j(f)^{2^{n-1-j}},\qquad \Upsilon_j(f)=\frac{(-1)^{j+1}\mathcal{F}^{(j+1)}(0)}{(j+1)!(\mathcal{F}'(0))^{j+1}},\tag{3}$$

and $\mathcal{F}$ is the inverse function of f. It is obvious from (1) and (2) that the *intermediate* error relation, similar to (2),

$$\varepsilon_{k,j}=y_{k,j}-\alpha\sim(1+\gamma f'(\alpha))^{2^{j-1}}B_j(f)\varepsilon_k^{2^j}\tag{4}$$

holds for each $1\le j\le n-1$. The case $j=n$ is also included in (4) having in mind that $\varepsilon_{k+1}=\varepsilon_{k,n}$.

Zheng–Li–Huang's family.

In the recent paper [3] Zheng, Li and Huang proposed another derivative free family (shorter Z–L–H family) of n-point methods of arbitrary order of convergence 2^n $(n \geq 1)$. This family is constructed using Newton's interpolation with forward divided differences.

Z–L–H family: for an initial approximation x_0, arbitrary $n \in \mathbf{N}$ and $k = 0, 1, \ldots$, the n-point methods are defined by

$$\begin{cases} y_{k,0} = x_k, \qquad y_{k,-1} = y_{k,0} + \gamma f(y_{k,0}), \quad \gamma \in \mathbf{R} \setminus \{0\}, \\ y_{k,1} = y_{k,0} - \dfrac{f(y_{k,0})}{f[y_{k,0}, y_{k,-1}]}, \\ y_{k,2} = y_{k,1} - \dfrac{f(y_{k,1})}{f[y_{k,1}, y_{k,0}] + f[y_{k,1}, y_{k,0}, y_{k,-1}](y_{k,1} - y_{k,0})}, \\ \vdots \\ y_{k,n} = y_{k,n-1} - \dfrac{f(y_{k,n-1})}{f[y_{k,n-1}, y_{k,n-2}] + \sum_{j=1}^{n-1} f[y_{k,n-1}, \ldots, y_{k,n-2-j}] \prod_{i=1}^{j} (y_{k,n-1} - y_{k,n-1-i})}. \end{cases} \tag{5}$$

The entries $y_{k,1}, \ldots, y_{k,n-1}$ are intermediate approximations of Z–L–H family, while $x_{k+1} = y_{k,n}$ is the ultimate approximation. $f[x, y] = (f(x) - f(y))/(x - y)$ denotes a divided difference. Divided differences of higher order are defined recursively by

$$f[x_0, x_1, \ldots, x_i] = \frac{f[x_1, \ldots, x_i] - f[x_0, \ldots, x_{i-1}]}{x_i - x_0} \quad (i > 1).$$

The following theorem was proved in [3].

Theorem 1. *Let a function $f : D \subset \mathbf{R} \to \mathbf{R}$ be sufficiently differentiable having a simple zero α in an open interval $I_f \subset D$. If x_0 is close enough to α, then the n-point family (5) converges with at least 2^n-th order and satisfies the error relation*

$$\varepsilon_{k+1} = x_{k+1} - \alpha = y_{k,n} - \alpha \sim D_n \varepsilon_k^{2^n} \quad (k = 0, 1, \ldots), \tag{6}$$

where

$$D_{-1} = 1 + \gamma f'(\alpha), \qquad D_0 = 1, \qquad D_1 = (1 + \gamma f'(\alpha))c_2, \tag{7}$$

$$D_m = D_{m-1}\Big[c_2 D_{m-1} + (-1)^{m-1} c_{m+1} D_{m-2} \cdots D_{-1}\Big] \quad (m = 2, \ldots, n). \tag{8}$$

As in the case of Kung–Traub's family (1), the *intermediate* error relations are given with

$$\varepsilon_{k,j} = y_{k,j} - \alpha \sim D_j \varepsilon_k^{2^j} \quad (j = 1, \ldots, n), \tag{9}$$

where constants D_j are calculated recursively by (7) and (8). The *ultimate* error relation is included in (9) for $j = n$.

We wish to show that constants D_m in the error relations (9) are of the form

$$D_m = (1 + \gamma f'(\alpha))^{2^{m-1}} d_m \quad (m = 1, \ldots, n), \tag{10}$$

where

$$d_{-1} = D_{-1} = 1, \qquad d_0 = D_0 = 1, \qquad d_1 = c_2, \tag{11}$$

$$d_m = d_{m-1}\Big[c_2 d_{m-1} + (-1)^{m-1} c_{m+1} d_{m-2} \cdots d_{-1}\Big] \quad (m = 2, \ldots, n). \tag{12}$$

For $m = 1$, assertion (10) is obvious. Let us assume that (10) and (12) are true for all $m < n$. According to (8) we find

$$\begin{aligned} D_n &= D_{n-1}\Big[c_2 D_{n-1} + (-1)^{n-1} c_{n+1} D_{n-2} \cdots D_{-1}\Big] \\ &= (1 + \gamma f'(\alpha))^{2^{n-2}} d_{n-1}\Big[c_2 (1 + \gamma f'(\alpha))^{2^{n-2}} d_{n-1} \\ &\quad + (-1)^{n-1} c_{n+1} (1 + \gamma f'(\alpha))^{2^{n-3}} d_{n-2} \cdots (1 + \gamma f'(\alpha)) d_1 d_0 (1 + \gamma f'(\alpha)) d_{-1}\Big] \\ &= (1 + \gamma f'(\alpha))^{2^{n-2}} d_{n-1}\Big[c_2 (1 + \gamma f'(\alpha))^{2^{n-2}} d_{n-1} + (-1)^{n-1} c_{n+1} (1 + \gamma f'(\alpha))^{2^{n-3} + 2^{n-2} + \cdots + 1 + 1} d_{n-2} \cdots d_{-1}\Big] \\ &= (1 + \gamma f'(\alpha))^{2^{n-2}} d_{n-1}\Big[c_2 (1 + \gamma f'(\alpha))^{2^{n-2}} d_{n-1} + (-1)^{n-1} c_{n+1} (1 + \gamma f'(\alpha))^{2^{n-2}} d_{n-2} \cdots d_{-1}\Big] \\ &= (1 + \gamma f'(\alpha))^{2^{n-1}} d_{n-1}\Big[c_2 d_{n-1} + (-1)^{n-1} c_{n+1} d_{n-2} \cdots d_{-1}\Big]. \end{aligned}$$

Hence, by induction, we conclude that the intermediate error relations can be written in the following form

$$\varepsilon_{k,j} \sim d_j(1+\gamma f'(\alpha))^{2^{j-1}}\varepsilon_k^{2^j} \quad (j=1,\ldots,n), \tag{13}$$

where d_j is defined by (11) and (12). Note that (13) includes the ultimate error relation for $j=n$, that is,

$$\varepsilon_{k+1}=\varepsilon_{k,n}=y_{k,n}-\alpha \sim d_n(1+\gamma f'(\alpha))^{2^{n-1}}\varepsilon_k^{2^n}. \tag{14}$$

Remark 1. Both families (1) and (5) of n-point methods have the same order of convergence 2^n and require $n+1$ function evaluations, which means that they support the Kung–Traub conjecture on the upper bound of the order of convergence in the class of methods without memory. These families are both derivative free, and have similar structure and the error relations of the same type, providing us to carry out the convergence analysis of both families simultaneously.

3. Derivative free families with memory

In this section, we show that the Kung–Traub family (1) and the Zheng–Li–Huang family (5) can be considerably accelerated without any additional function evaluations. The construction of new families of n-point derivative free methods is based on the variation of a free parameter γ in each iterative step. This parameter is calculated using information from the current and the previous iteration so that the presented methods may be regarded as the methods with memory.

As pointed out in [6], the factor $1+\gamma f'(\alpha)$ in the error relations (2) and (4) (for the K–T family), and (13) and (14) (for the Z–L–H family) plays the key role in derivation of the families with memory. The error relations (4) and (13) can be presented in the unique form

$$\varepsilon_{k,j} \sim a_{k,j}(1+\gamma f'(\alpha))^{2^{j-1}}\varepsilon_k^{2^j} \quad (j=1,\ldots,n), \tag{15}$$

where $\varepsilon_k = y_{k,0}-\alpha$ and $\varepsilon_{k,j}=y_{k,j}-\alpha\,(j=1,\ldots,n)$, k being the iteration index. Constants $a_{k,j}$ depend on the considered family and they can be determined recursively from (4) and (12). However, in this paper we concentrate on the lower bound of the R-order of the methods with memory so that the specific expressions of d_m and asymptotic error constants are out of our interest. The use of the unique relation (15) enables us to consider simultaneously both families with memory based on (1) and (5).

We observe from (2) and (14) that the order of convergence of families (1) and (5) is 2^n when $\gamma \neq -1/f'(\alpha)$. Obviously, if we could provide $\gamma=-1/f'(\alpha)$, the order of families (1) and (5) would exceed 2^n; more precisely, it is not difficult to show that the order of these families would be 2^n+2^{n-1}. However, the value $f'(\alpha)$ is not available in practice and we could use only an approximation $\widetilde{f}'(\alpha) \approx f'(\alpha)$, calculated by using available information. Then, setting $\gamma_k=-1/\widetilde{f}'(\alpha)$, we achieve that the order of convergence of the modified methods exceeds 2^n without using any new function evaluations.

The beneficial approach in approximating $\gamma \approx -1/f'(\alpha)$ is to use only available information, in other words, we wish to increase the convergence speed without additional function evaluations. We present three models for approximating $-1/f'(\alpha)$:

(I) $\widetilde{f}'(\alpha)=\frac{f(y_{k,0})-f(y_{k-1,j})}{y_{k,0}-y_{k-1,j}}$ (secant approach);

(II) $\widetilde{f}'(\alpha)=N_2'(y_{k,0})$ (Newton's interpolatory approach), where

$$N_2(t)=N_2(t;y_{k,0},y_{k-1,n-1},y_{k-1,n-2})$$

is Newton's interpolating polynomial of second degree, set through three best available approximations (nodes) $y_{k,0}, y_{k-1,n-1}, y_{k-1,n-2}$;

(III) $\widetilde{f}'(\alpha)=N_3'(x_k)$ (improved Newton's interpolatory approach), where

$$N_3(t)=N_3(t;y_{k,0},y_{k-1,n-1},y_{k-1,n-2},y_{k-1,n-3})$$

is Newton's interpolating polynomial of third degree, set through four best available approximations (nodes) $y_{k,0}, y_{k-1,n-1}, y_{k-1,n-2}, y_{k-1,n-3}$.

Using divided differences, we find

$$N_2'(y_{k,0})=f[y_{k,0},y_{k-1,n-1}]+f[y_{k,0},y_{k-1,n-1},y_{k-1,n-2}](y_{k,0}-y_{k-1,n-1}) \tag{16}$$

and

$$\begin{aligned}N_3'(y_{k,0}) &= f[y_{k,0},y_{k-1,n-1}]+f[y_{k,0},y_{k-1,n-1},y_{k-1,n-2}](y_{k,0}-y_{k-1,n-1})\\ &\quad+f[y_{k,0},y_{k-1,n-1},y_{k-1,n-2},y_{k-1,n-3}](y_{k,0}-y_{k-1,n-1})(y_{k,0}-y_{k-1,n-2}).\end{aligned} \tag{17}$$

Note that the Zheng–Li–Huang family (5) is very suitable for the application of Newton's interpolatory approaches (II) and (III) since divided differences are already calculated in the implementation of the iterative scheme (5).

Regarding the above methods (I), (II) and (III), we present the following three formulae for calculating the varying parameter γ_k:

$$\gamma_k = -\frac{y_{k,0} - y_{k-1,j}}{f(y_{k,0}) - f(y_{k-1,j})}, \quad \text{(method (I))}, \tag{18}$$

$$\gamma_k = -\frac{1}{N_2'(y_{k,0})} \quad \text{(method (II))}, \tag{19}$$

$$\gamma_k = -\frac{1}{N_3'(y_{k,0})} \quad \text{(method (III))}. \tag{20}$$

Replacing the fixed parameter γ in the iterative formulae (1) and (5) by the varying parameter γ_k calculated by (18), (19) or (20), we state the following families of multipoint methods with memory.

K–T family with memory: for an initial approximation x_0, arbitrary $n \in \mathbf{N}$ and γ_k calculated by (18), (19) or (20) and $k = 0, 1, \ldots,$ define iterative function $\psi_j(f)$ $(j = -1, 0, \ldots, n)$ as follows:

$$\begin{cases} y_{k,0} = \psi_0(f)(x_k) = x_k, \qquad y_{k,-1} = \psi_{-1}(f)(x_k) = x_k + \gamma_k f(x_k), \\ y_{k,j} = \psi_j(f)(x) = \mathcal{R}_j(0), \quad j = 1, \ldots, n, \ \text{for } n > 0, \\ x_{k+1} = y_{k,n} = \psi_n(f)(x_k). \end{cases} \tag{21}$$

Z–L–H family with memory: for an initial approximation x_0, arbitrary $n \in \mathbf{N}$, γ_k calculated by (18), (19) or (20) and $k = 0, 1, \ldots,$ the n-point methods are defined by

$$\begin{cases} y_{k,0} = x_k, \qquad y_{k,-1} = y_{k,0} + \gamma_k f(y_{k,0}), \\ y_{k,1} = y_{k,0} - \dfrac{f(y_{k,0})}{f[y_{k,0}, y_{k,-1}]}, \\ y_{k,2} = y_{k,1} - \dfrac{f(y_{k,1})}{f[y_{k,1}, y_{k,0}] + f[y_{k,1}, y_{k,0}, y_{k,-1}](y_{k,1} - y_{k,0})}, \\ \vdots \\ y_{k,n} = y_{k,n-1} - \dfrac{f(y_{k,n-1})}{f[y_{k,n-1}, y_{k,n-2}] + \sum_{j=1}^{n-1} f[y_{k,n-1}, \ldots, y_{k,n-2-j}] \prod_{i=1}^{j} (y_{k,n-1} - y_{k,n-1-i})}. \end{cases} \tag{22}$$

We use the term *method with memory* following Traub's classification [1, p. 8] and the fact that the evaluation of γ_k depends on the data available from the current and the previous iterative step.

4. *R*-order of convergence of the families with memory

To estimate the convergence rate of families (21) and (22), we use the concept of the *R*-order of convergence introduced in [7]. We distinguish three approaches for the calculation of the varying parameter γ_k given by formulae (18) (method (I)), (19) (method (II)) and (20) (method (III)).

Method (I)—secant approach.

Applying the secant approach (18), we have to estimate the factor $1 + \gamma_k f'(\alpha)$ in (15). For this purpose we use Taylor's expansion of f about its simple zero α (thus $f'(\alpha) \neq 0$),

$$f(x) = f'(\alpha)\Big(\varepsilon + c_2\varepsilon^2 + c_3\varepsilon^3 + \cdots\Big), \quad \varepsilon = x - \alpha. \tag{23}$$

Using (23) for $x = y_{k,0}$ and $x = y_{k-1,j}$, there follows from (18)

$$\begin{aligned} \gamma_k &= -\frac{y_{k,0} - y_{k-1,j}}{f(y_{k,0}) - f(y_{k-1,j})} \\ &= -\frac{\varepsilon_k - \varepsilon_{k-1,j}}{f'(\alpha)(\varepsilon_k - \varepsilon_{k-1,j} + c_2(\varepsilon_k^2 - \varepsilon_{k-1,j}^2) + O(\varepsilon_k^3 - \varepsilon_{k-1,j}^3))} \\ &= -\frac{1}{f'(\alpha)}(1 - c_2(\varepsilon_{k-1,j} + \varepsilon_k) + O(\varepsilon_{k-1,j}^2)). \end{aligned}$$

Hence,

$$1 + \gamma_k f'(\alpha) = c_2(\varepsilon_{k-1,j} + \varepsilon_k) + O(\varepsilon_{k-1,j}^2) \sim c_2\varepsilon_{k-1,j}. \tag{24}$$

Suppose that the R-order of convergence of the improved families with error relation (15) is r_j; then we may write

$$\varepsilon_{k+1} \sim A_{k,r_j}\varepsilon_k^{r_j}, \tag{25}$$

where A_{k,r_j} tends to the asymptotic error constant A_{r_j} when $k \to \infty$. Hence

$$\varepsilon_{k+1} \sim A_{k,r_j}\left(A_{k-1,r_j}\varepsilon_{k-1}^{r_j}\right)^{r_j} = A_{k,r_j}A_{k-1,r_j}^{r_j}\varepsilon_{k-1}^{r_j^2}. \tag{26}$$

In a similar fashion, if we suppose that $\{y_{k,j}\}$ is an iterative sequence of the R-order p_j for fixed $0 < j < n$, then

$$\varepsilon_{k,j} = y_{k,j} - \alpha \sim A_{k,p_j}\varepsilon_k^{p_j} \sim A_{k,p_j}\left(A_{k-1,r_j}\varepsilon_{k-1}^{r_j}\right)^{p_j} = A_{k,p_j}A_{k-1,r_j}^{p_j}\varepsilon_{k-1}^{r_j p_j}. \tag{27}$$

Combining (15), (24), (26) and (27), we arrive at

$$\begin{aligned}
\varepsilon_{k,j} &\sim a_{k,j}\big(1+\gamma_k f'(\alpha)\big)^{2^{j-1}}\varepsilon_k^{2^j} \\
&\sim a_{k,j}\left(c_2\varepsilon_{k-1,j}\right)^{2^{j-1}}\left(A_{k-1,r_j}\varepsilon_{k-1}^{r_j}\right)^{2^j} \\
&\sim a_{k,j}c_2^{2^{j-1}}\left(A_{k-1,p_j}\varepsilon_{k-1}^{p_j}\right)^{2^{j-1}}A_{k-1,r_j}^{2^j}\left(\varepsilon_{k-1}\right)^{r_j 2^j} \\
&= a_{k,j}c_2^{2^{j-1}}A_{k-1,p_j}^{2^{j-1}}A_{k-1,r_j}^{2^j}\left(\varepsilon_{k-1}\right)^{p_j 2^{j-1}+r_j 2^j}
\end{aligned} \tag{28}$$

and, in a similar way,

$$\begin{aligned}
\varepsilon_{k+1} &\sim a_{k,n}\big(1+\gamma_k f'(\alpha)\big)^{2^{n-1}}\varepsilon_k^{2^n} \\
&\sim a_{k,n}\left(c_2\varepsilon_{k-1,j}\right)^{2^{n-1}}\left(A_{k-1,r_j}\varepsilon_{k-1}^{r_j}\right)^{2^n} \\
&\sim a_{k,n}c_2^{2^{n-1}}A_{k-1,p_j}^{2^{n-1}}A_{k-1,r_j}^{2^n}\left(\varepsilon_{k-1}\right)^{p_j 2^{n-1}+r_j 2^n}.
\end{aligned} \tag{29}$$

Equating exponents of ε_{k-1} in (27) and (28), and then in (26) and (29), for $0 < j < n$ we form the system of equations with unknown orders p_j and r_j

$$\begin{cases} r_j p_j - p_j 2^{j-1} - r_j 2^j = 0, \\ r_j^2 - p_j 2^{n-1} - r_j 2^n = 0. \end{cases} \tag{30}$$

Positive solutions of this system are $p_j = 2^j + 2^{2j-n-1}$ and $r_j = 2^n + 2^{j-1}$. Hence, applying method (I), the R-order of the families (21) and (22) is at least $2^n + 2^{j-1}$.

In particular, let us consider successive approximations x_{k-1} and x_k in the secant method (18), that is,

$$\gamma_k = -\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}. \tag{31}$$

Such an accelerating method was first considered in Traub's book [1] and recently in [6]. Obviously, this approach gives the worst approximation to $f'(\alpha)$ and corresponds to the values $j = 0$ and $p_0 = 1$. In this case it is sufficient to consider only the second equation of (30), which reduces to the quadratic equation

$$r^2 - 2^n r - 2^{n-1} = 0.$$

The positive solution

$$r = 2^{n-1}\left(1 + \sqrt{1 + 2^{1-n}}\,\right)$$

gives the lower bound of the R-order of the families.

We summarize our results in the following theorem.

Theorem 2. *Let the varying parameter γ_k in the iterative formulae (21) and (22) be calculated by (18) for $j = 1, \ldots, n-1$ and (31) for $j = 0$. If an initial approximation x_0 is sufficiently close to a simple zero α of f, then the R-order of convergence of the families (21) and (22) of n-point methods with memory is at least $2^n + 2^{j-1}$ for $j = 1, \ldots, n-1$, and $2^{n-1}\left(1 + \sqrt{1 + 2^{1-n}}\,\right)$ for $j = 0$.*

Table 1
The lower bounds of the R-order.

n	Method (I)				Method (II)	Method (III)	No memory
	$j=0$	$j=1$	$j=2$	$j=3$			
2	**4.449** (11.2%)	**5** (25%)			**5.372** (34%)	**6** (50%)	**4**
3	**8.472** (6%)	**9** (12.5%)	**10** (25%)		**11** (37.5%)	**11.35** (41.9%)	**8**
4	**16.485** (3%)	**17** (6.25%)	**18** (12.5%)	**20** (25%)	**22** (37.5%)	**23** (43.7%)	**16**

Note that the use of the better intermediate approximation $y_{k-1,j}$ in the secant method (I) will give the better approximation to $f'(\alpha)$ and, consequently, higher order of the modified methods (21) and (22), based on (1) and (5) and dealing with the varying parameter γ_k. Some particular values of the R-order are given in Table 1.

Remark 2. The secant method (I) is, in fact, the derivative $N_1'(y_{k,0})$ of Newton's interpolating polynomial of first order at the nodes $y_{k,0}$ and $y_{k-1,j}$ so that relation (24) can be derived using Lemma 1 given below.

Now we estimate the R-order of convergence of families (21) and (22) with memory when Newton's interpolatory approaches (II) and (III) are applied. In our analysis we use the Bachman–Landau *o-notation*: for the sequences $\{a_k\}$ and $\{b_k\}$ which tend to 0 when $k \to \infty$ we write $a_k = o(b_k)$ if $\lim_{k\to\infty} a_k/b_k = 0$; in other words, a is dominated by b asymptotically. First we state the following assertion.

Lemma 1. *Let N_m be the Newton interpolating polynomial of the degree m that interpolates a function f at $m+1$ distinct interpolation nodes $y_{k,0}, y_{k-1,n-1}, \dots, y_{k-1,n-m}$ contained in an interval I and the derivative $f^{(m+1)}$ is continuous in I. Define the errors $\varepsilon_{k-1,n-j} = y_{k-1,n-j} - \alpha$ $(i \in \{1, \dots, m\})$ and assume that*

(1) *all nodes $y_{k,0}, y_{k-1,n-1}, \dots, y_{k-1,n-m}$ are sufficiently close to the zero α;*
(2) *the condition $\varepsilon_{k,0} = o\big(\varepsilon_{k-1,n-1} \cdots \varepsilon_{k-1,n-m}\big)$ holds.*

Then

$$N_m'(y_{k,0}) \sim f'(\alpha)\left(1 + (-1)^{m+1} c_{m+1} \prod_{j=1}^{m} \varepsilon_{k-1,n-j}\right). \tag{32}$$

Proof. Taylor's series of derivatives of f at the points $y_{k,0} \in I$ and $d \in I$ about the zero α of f give

$$f'(y_{k,0}) = f'(\alpha)\big(1 + 2c_2\varepsilon_{k,0} + 3c_3\varepsilon_{k,0}^2 + \cdots\big), \tag{33}$$

$$f^{(m+1)}(d) = f'(\alpha)\left((m+1)!c_{m+1} + \frac{(m+2)!}{1!} c_{m+2}\varepsilon_d + \cdots\right), \tag{34}$$

where $\varepsilon_d = d - \alpha$ and d is the point that appears in the formula for the error of the Newton interpolation

$$f(t) - N_m(t) = \frac{f^{(m+1)}(d)}{(m+1)!}(t - y_{k,0}) \prod_{j=1}^{m} \big(t - y_{k-1,n-j}\big) \quad (d \in I). \tag{35}$$

After differentiating (35) we obtain at the point $t = y_{k,0}$

$$N_m'(y_{k,0}) = f'(y_{k,0}) - \frac{f^{(m+1)}(d)}{(m+1)!} \prod_{j=1}^{m} (y_{k,0} - y_{k-1,n-j}). \tag{36}$$

Substituting (33) and (34) into (36) and bearing in mind the conditions of Lemma 1, after some elementary calculations we come to relation (32). □

Method (II)—Newton's interpolation of second degree.

First, let us consider the case $n \geq 3$ and assume that the R-orders of the iterative sequences $\{y_{k,n-2}\}$, $\{y_{k,n-1}\}$ and $\{y_k\}$ are at least p, q and r, respectively, that is,

$$\varepsilon_{k,n-2} \sim A_{k,p}\varepsilon_k^p, \qquad \varepsilon_{k,n-1} \sim A_{k,q}\varepsilon_k^q, \qquad \varepsilon_{k+1} \sim A_{k,r}\varepsilon_k^r.$$

Hence

$$\varepsilon_{k,n-2} \sim A_{k,p}\left(A_{k-1,r}\varepsilon_{k-1}^r\right)^p = A_{k,p}A_{k-1,r}^p\varepsilon_{k-1}^{rp}, \tag{37}$$

$$\varepsilon_{k,n-1} \sim A_{k,q}\left(A_{k-1,r}\varepsilon_{k-1}^r\right)^q = A_{k,q}A_{k-1,r}^q\varepsilon_{k-1}^{rq}, \tag{38}$$

$$\varepsilon_{k+1} \sim A_{k,r}\left(A_{k-1,r}\varepsilon_{k-1}^r\right)^r = A_{k,r}A_{k-1,r}^r\varepsilon_{k-1}^{r^2}. \tag{39}$$

In view of Lemma 1 for $m = 2$ we have

$$N_2'(y_{k,0}) \sim f'(\alpha)\big(1 - c_3\varepsilon_{k-1,n-2}\varepsilon_{k-1,n-1}\big).$$

According to this and (19) we find

$$1 + \gamma_k f'(\alpha) \sim c_3\varepsilon_{k-1,n-2}\varepsilon_{k-1,n-1}. \tag{40}$$

Using (40) and the previously derived relations, we obtain the error relations for the intermediate approximations

$$\begin{aligned}\varepsilon_{k,n-2} &\sim a_{k,n-2}\Big(1+\gamma_k f'(\alpha)\Big)^{2^{n-3}}\varepsilon_k^{2^{n-2}}\\ &\sim a_{k,n-2}\Big(c_3\varepsilon_{k,n-2}\varepsilon_{k,n-1}\Big)^{2^{n-3}}\Big(A_{k-1,r}\varepsilon_{k-1}^r\Big)^{2^{n-2}}\\ &\sim a_{k,n-2}c_3^{2^{n-3}}A_{k-1,r}^{2^{n-2}}\Big(A_{k-1,p}A_{k-1,q}\Big)^{2^{n-3}}\Big(\varepsilon_{k-1}\Big)^{(p+q)2^{n-3}+r2^{n-2}}\end{aligned} \tag{41}$$

and

$$\begin{aligned}\varepsilon_{k,n-1} &\sim a_{k,n-1}\Big(1+\gamma_k f'(\alpha)\Big)^{2^{n-2}}\varepsilon_k^{2^{n-1}}\\ &\sim a_{k,n-1}\Big(c_3\varepsilon_{k,n-2}\varepsilon_{k,n-1}\Big)^{2^{n-2}}\Big(A_{k-1,r}\varepsilon_{k-1}^r\Big)^{2^{n-1}}\\ &\sim a_{k,n-1}c_3^{2^{n-2}}A_{k-1,r}^{2^{n-1}}\Big(A_{k-1,p}A_{k-1,q}\Big)^{2^{n-2}}\Big(\varepsilon_{k-1}\Big)^{(p+q)2^{n-2}+r2^{n-1}}.\end{aligned} \tag{42}$$

Similarly,

$$\begin{aligned}\varepsilon_{k+1} &\sim a_{k,n}\Big(1+\gamma_k f'(\alpha)\Big)^{2^{n-1}}\varepsilon_k^{2^{n}}\\ &\sim a_{k,n}\Big(c_3\varepsilon_{k,n-2}\varepsilon_{k,n-1}\Big)^{2^{n-1}}\Big(A_{k-1,r}\varepsilon_{k-1}^r\Big)^{2^{n}}\\ &\sim a_{k,n}c_3^{2^{n-1}}A_{k-1,r}^{2^{n}}\Big(A_{k-1,p}A_{k-1,q}\Big)^{2^{n-1}}\Big(\varepsilon_{k-1}\Big)^{(p+q)2^{n-1}+r2^{n}}.\end{aligned} \tag{43}$$

Equating error exponents of ε_{k-1} in three pairs of error relations (37)∧ (41), (38)∧ (39) and (42)∧ (43), we form the following system of equations in unknown orders p, q and r,

$$\begin{cases} rp - (p+q)2^{n-3} - r2^{n-2} = 0,\\ rq - (p+q)2^{n-2} - r2^{n-1} = 0,\\ r^2 - (p+q)2^{n-1} - r2^{n} = 0.\end{cases} \tag{44}$$

Positive solutions of this system are $p = 11\cdot 2^{n-5}$, $q = 11\cdot 2^{n-4}$, $r = 11\cdot 2^{n-3}$. Therefore, the R-order of convergence of families (21)∧ (19) and (22)∧ (19) is at least $11\cdot 2^{n-3}$ for $n \ge 3$. For example, the R-order of the three-point families (21) and (22) is at least 11, the four-point families have the R-order at least 22, etc. (see Table 1), assuming that γ_k is calculated by (19).

The case $n = 2$ slightly differs from the previous analysis; Newton's interpolating polynomial is constructed at the nodes x_{k-1} $(=y_{k-1,0})$, y_{k-1} $(=y_{k-1,1})$ and x_k $(=y_{k,0})$ and we may formally take $p = 1$ in (44) and remove the first equation. Then the system of Eqs. (44) reduces to

$$\begin{cases} rq - (q+1) - 2r = 0,\\ r^2 - 2(q+1) - 4r = 0,\end{cases}$$

with the solutions $q = \frac{1}{4}(5+\sqrt{33})$ and $r = \frac{1}{2}(5+\sqrt{33})$. Therefore, families (21)∧ (19) and (22)∧ (19) of two-point methods with memory have the R-order at least $\frac{1}{2}(5+\sqrt{33}) \approx 5.372$.

According to the previous study we can state the following convergence theorem.

Theorem 3. *Let the varying parameter γ_k in the iterative formulae (21) and (22) be calculated by (19). If an initial approximation x_0 is sufficiently close to a simple zero α of f, then the R-order of convergence of families (21) and (22) of n-point methods with memory is at least $11\cdot 2^{n-3}$ for $n \ge 3$ and at least $\frac{1}{2}(5+\sqrt{33}) \approx 5.372$ for $n = 2$.*

Method (III)—Newton's interpolation of third degree.

The calculation of γ_k by (20) uses more information compared to (19) and we expect to achieve faster convergence. The presented convergence analysis confirms our assumption.

Let $n \geq 4$ and assume that the R-order of the iterative sequences $\{y_{k,n-3}\}$, $\{y_{k,n-2}\}$, $\{y_{k,n-1}\}$ and $\{y_k\}$ is at least p, q, s and r, respectively, that is,

$$\varepsilon_{k,n-3} \sim A_{k,p}\varepsilon_k^p, \qquad \varepsilon_{k,n-2} \sim A_{k,q}\varepsilon_k^q, \qquad \varepsilon_{k,n-1} \sim A_{k,s}\varepsilon_k^s, \qquad \varepsilon_{k+1} \sim A_{k,r}\varepsilon_k^r.$$

Hence

$$\varepsilon_{k,n-3} \sim A_{k,p}\left(A_{k-1,r}\varepsilon_{k-1}^r\right)^p = A_{k,p}A_{k-1,r}^p\varepsilon_{k-1}^{rp}, \tag{45}$$

$$\varepsilon_{k,n-2} \sim A_{k,q}\left(A_{k-1,r}\varepsilon_{k-1}^r\right)^q = A_{k,p}A_{k-1,r}^q\varepsilon_{k-1}^{rq}, \tag{46}$$

$$\varepsilon_{k,n-1} \sim A_{k,s}\left(A_{k-1,r}\varepsilon_{k-1}^r\right)^s = A_{k,s}A_{k-1,r}^s\varepsilon_{k-1}^{rs}, \tag{47}$$

$$\varepsilon_{k+1} \sim A_{k,r}\left(A_{k-1,r}\varepsilon_{k-1}^r\right)^r = A_{k,r}A_{k-1,r}^r\varepsilon_{k-1}^{r^2}. \tag{48}$$

According to Lemma 1 for $m = 3$, we have

$$N_3'(y_{k,0}) \sim f'(\alpha)\big(1 + c_4\varepsilon_{k-1,n-3}\varepsilon_{k-1,n-2}\varepsilon_{k-1,n-1}\big).$$

From the last relation and (20) we find

$$1 + \gamma_k f'(\alpha) \sim c_4\varepsilon_{k-1,n-3}\varepsilon_{k-1,n-2}\varepsilon_{k-1,n-1}. \tag{49}$$

Combining (49) and the previously derived relations, we derive the following error relations

$$\begin{aligned} \varepsilon_{k,n-3} &\sim a_{k,n-3}\Big(1 + \gamma_k f'(\alpha)\Big)^{2^{n-4}} \varepsilon_k^{2^{n-3}} \\ &\sim a_{k,n-3}c_4^{2^{n-4}}A_{k-1,r}^{2^{n-3}}\Big(A_{k-1,p}A_{k-1,q}A_{k-1,s}\Big)^{2^{n-4}}\Big(\varepsilon_{k-1}\Big)^{2^{n-3}r+2^{n-4}(p+q+s)}, \end{aligned} \tag{50}$$

$$\begin{aligned} \varepsilon_{k,n-2} &\sim a_{k,n-2}\Big(1 + \gamma_k f'(\alpha)\Big)^{2^{n-3}} \varepsilon_k^{2^{n-2}} \\ &\sim a_{k,n-2}c_4^{2^{n-3}}A_{k-1,r}^{2^{n-2}}\Big(A_{k-1,p}A_{k-1,q}A_{k-1,s}\Big)^{2^{n-3}}\Big(\varepsilon_{k-1}\Big)^{2^{n-2}r+2^{n-3}(p+q+s)}, \end{aligned} \tag{51}$$

$$\begin{aligned} \varepsilon_{k,n-1} &\sim a_{k,n-1}\Big(1 + \gamma_k f'(\alpha)\Big)^{2^{n-2}} \varepsilon_k^{2^{n-1}} \\ &\sim a_{k,n-1}c_4^{2^{n-2}}A_{k-1,r}^{2^{n-1}}\Big(A_{k-1,p}A_{k-1,q}A_{k-1,s}\Big)^{2^{n-2}}\Big(\varepsilon_{k-1}\Big)^{2^{n-1}r+2^{n-2}(p+q+s)}, \end{aligned} \tag{52}$$

$$\begin{aligned} \varepsilon_{k+1} &\sim a_{k,n}\Big(1 + \gamma_k f'(\alpha)\Big)^{2^{n-1}} \varepsilon_k^{2^{n}} \\ &\sim a_{k,n}c_4^{2^{n-1}}A_{k-1,r}^{2^{n}}\Big(A_{k-1,p}A_{k-1,q}A_{k-1,s}\Big)^{2^{n-1}}\Big(\varepsilon_{k-1}\Big)^{2^{n}r+2^{n-1}(p+q+s)}. \end{aligned} \tag{53}$$

In a similar way as before, equating exponents of ε_{k-1} in four pairs of error relations (45)∧ (50), (46)∧ (51), (47)∧ (48) and (52)∧ (53), we form the following system of equations,

$$\begin{cases} rp - (p+q+s)2^{n-4} - r2^{n-3} = 0, \\ rq - (p+q+s)2^{n-3} - r2^{n-2} = 0, \\ rs - (p+q+s)2^{n-2} - r2^{n-1} = 0, \\ r^2 - (p+q+s)2^{n-1} - r2^{n} = 0. \end{cases} \tag{54}$$

Positive solutions of this system are $p = 23 \cdot 2^{n-7}$, $q = 23 \cdot 2^{n-6}$, $r = 23 \cdot 2^{n-4}$. Therefore, the R-order of convergence of families (21)∧ (20) and (22)∧ (20) is at least $23 \cdot 2^{n-4}$ for $n \geq 4$. For example, the R-order of the four-point families (21) and (22) is at least 23.

To find the R-order of the three-point families (21)∧ (20) and (22)∧ (20), we put $n = 3$ and $p = 1$ in system (54), remove the first equation and solve the system of three equations

$$\begin{cases} rq - (1+q+s) - 2r = 0, \\ rs - 2(1+q+s) - 4r = 0, \\ r^2 - 4(1+q+s) - 8r = 0. \end{cases}$$

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

n	Method (I)				Method(II)	Method (III)	No memory
	$j=0$	$j=1$	$j=2$	$j=3$			
2	1.645	1.710			1.751	1.817	1.587
3	1.706	1.732	1.778		1.821	1.836	1.682
4	1.759	1.762	1.783	1.820	1.856	1.872	1.741

Positive solutions are $q = \frac{1}{8}\left(11+\sqrt{137}\right)$, $s = \frac{1}{4}\left(11+\sqrt{137}\right)$, $r = \frac{1}{2}\left(11+\sqrt{137}\right)$. Therefore, in this particular case the R-order is at least $\frac{1}{2}\left(11+\sqrt{137}\right) \approx 11.352$.

It remains to examine the case $n = 2$. The corresponding Newton interpolating polynomial is constructed through the points $y_{k-1,-1} = x_{k-1} + \gamma_{k-1} f(x_{k-1})$, x_{k-1}, $y_{k-1,1}$ and x_k. Analysis of the sequences $\{y_{k,-1}\}$, $\{y_{k-1,1}\}$ and $\{x_k\}$ (of orders p, q and r) and the same argumentation as above lead to the system

$$\begin{cases} rp - r - p - q - 1 = 0, \\ rq - 2r - p - q - 1 = 0, \\ r^2 - 4r - 2p - 2q - 2 = 0. \end{cases}$$

Hence, we find $r = 6$ and conclude that the lower bound of the R-order of the two-point methods with memory (21)∧ (20) and (22)∧ (20) is at least six.

Our results are summarized in the following theorem.

Theorem 4. *Let the varying parameter γ_k in the iterative formulae (21) and (22) be calculated by (20). If an initial approximation x_0 is sufficiently close to a simple zero α of f, then the R-order of convergence of families (21) and (22) of n-point methods with memory is at least $23 \cdot 2^{n-4}$ for $n \geq 4$, at least $\frac{1}{2}(11+\sqrt{137}) \approx 11.352$ for $n = 3$ and 6 for $n = 2$.*

Remark 3. From Theorems 3 and 4, we conclude that the multipoint methods for $n > 3$ are only of theoretical importance; indeed, multipoint methods with extraordinary fast convergence produce root approximations of considerable accuracy, not required in solving most practical problems. However, in this paper we have studied general families and general results on the convergence rate as a contribution to the general theory of iterative processes, emphasizing important particular cases $n = 2$ and $n = 3$.

The lower bounds of the R-order of families (21) and (22) for γ_k calculated by (18)–(20) are given in Table 1 for several entries of j and n.

From Table 1 we observe that the R-order of convergence of families (21) and (22) with memory is considerably increased related to the corresponding basic families (1) and (5) without memory (entries in the last column). The increment in percentage is also displayed. It is evident that the self-corrections (19) and (20), obtained by Newton's interpolation with divided differences, give the best results. It is worth noting that the improvement of convergence order in all cases is attained without any additional function evaluations, which points to a very high computational efficiency of the proposed methods with memory. Several values of the efficiency index

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

where r is the R-order of the considered iterative method (IM) and θ is the number of function evaluations per iterations, are given in Table 2. Numerical examples given in the next section entirely confirm the theoretical results presented in Theorems 2–4.

Remark 4. It is clear that the use of Newton's interpolatory polynomials of higher order than 3 can provide further increase of convergence order of n-point methods for $n \geq 3$. For example, using the described convergence analysis it is not difficult to prove that the R-order of the fourth-order families (21) and (22) with memory is 12 if the self-correcting parameter is calculated as

$$\gamma_k = -\frac{1}{N_4'(x_k)} \tag{55}$$

(see numerical results in Tables 3 and 4). However, for the reasons given in Remark 3, we are not interested in root-finders of extremely high order.

5. Numerical examples

We have tested families (1) and (5) without memory and the corresponding families (21) and (22) with memory using the programming package *Mathematica* with multiple-precision arithmetic. We regard that a proper challenge in designing root-finding methods is to develop iterative methods of as high as possible computational efficiency rather than very fast but

Table 3
$f(x) = e^{-x^2}(x-2)(1+x^3+x^6)$, $x_0 = 1.8$, $\alpha = 2$.

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	r_c(56)
K–T $n=2$	1.59(−3)	2.89(−11)	3.20(−42)	3.998
(57)	1.59(−3)	7.57(−13)	5.36(−54)	4.414
(59)	1.59(−3)	1.69(−14)	2.90(−69)	4.990
(60)	1.59(−3)	1.14(−15)	4.60(−81)	5.384
(61)	1.59(−3)	1.85(−17)	1.05(−100)	5.973
Z–L–H $n=2$	1.34(−3)	8.42(−12)	1.34(−44)	3.999
(57)	1.34(−3)	2.33(−13)	2.07(−56)	4.411
(59)	1.34(−3)	5.04(−15)	6.85(−72)	4.978
(60)	1.34(−3)	3.16(−16)	5.36(−84)	5.367
(61)	1.34(−3)	2.52(−18)	1.68(−106)	5.988
K–T $n=3$	6.43(−6)	2.01(−40)	1.80(−316)	8.000
(57)	6.43(−6)	1.38(−43)	3.13(−362)	8.459
(58)	6.43(−6)	6.86(−47)	1.50(−415)	8.998
(59)	6.43(−6)	2.53(−51)	1.39(−505)	10.004
(60)	6.43(−6)	3.20(−58)	3.11(−634)	11.013
(61)	6.43(−6)	7.82(−63)	3.12(−704)	11.274
(55)	6.43(−6)	4.27(−61)	4.82(−723)	11.996
Z–L–H $n=3$	7.20(−7)	2.50(−49)	5.23(−389)	7.999
(57)	7.20(−7)	1.91(−52)	3.73(−438)	8.463
(58)	7.20(−7)	8.96(−56)	1.66(−495)	8.992
(59)	7.20(−7)	1.76(−60)	9.34(−597)	10.003
(60)	7.20(−7)	9.29(−68)	1.69(−737)	10.999
(61)	7.20(−7)	9.26(−70)	7.11(−783)	11.339
(55)	7.20(−7)	2.29(−76)	1.08(−907)	11.962

expensive methods. Nevertheless, for demonstration of convergence behaviour of the proposed methods and comparison purpose, we present in Tables 3 and 4 approximations of high accuracy but only for two-point and three-point methods. Multipoint methods for $n > 3$ are very seldom required for solving practical problems.

The errors $|x_k - \alpha|$ of approximations to the zeros are given in Tables 3 and 4, where $A(-h)$ denotes $A \times 10^{-h}$. These tables include the values of the computational order of convergence r_c calculated by the formula [8]

$$r_c = \frac{\log |f(x_k)/f(x_{k-1})|}{\log |f(x_{k-1})/f(x_{k-2})|}, \tag{56}$$

taking into consideration the last three approximations in the iterative process. We have chosen the following test functions:

$$f(x) = e^{-x^2}(x-2)\big(1+x^3+x^6\big), \qquad \alpha = 2, \qquad x_0 = 1.8,$$

$$f(x) = \cos 2x + e^{x^2-1}\sin x - 2, \qquad x_0 = 1.33, \qquad \alpha = 1.447794857468\ldots.$$

For better readability, in this section we display explicitly five accelerating formulae for the calculation of the varying parameter γ_k, previously given by (18)–(20) and (31):

$$\gamma_k = -\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}, \tag{57}$$

$$\gamma_k = -\frac{y_{k,0} - y_{k-1,n-2}}{f(y_{k,0}) - f(y_{k-1,n-2})}, \tag{58}$$

$$\gamma_k = -\frac{y_{k,0} - y_{k-1,n-1}}{f(y_{k,0}) - f(y_{k-1,n-1})}, \tag{59}$$

$$\gamma_k = -\frac{1}{N_2'(y_{k,0})} = -\frac{1}{N_2'(x_k)}, \tag{60}$$

$$\gamma_k = -\frac{1}{N_3'(y_{k,0})} = -\frac{1}{N_3'(x_k)}. \tag{61}$$

In all numerical examples, the initial value $\gamma_0 = 0.01$ was used.

From Tables 3 and 4 and many tested examples we can conclude that all implemented methods produce approximations of great accuracy. Good initial approximations were obtained using an efficient method given in [9]. We observe that the methods with memory considerably increase the accuracy of obtained results. The quality of the calculation of γ_k by

Table 4
$f(x) = \cos 2x + e^{x^2-1}\sin x - 2$, $x_0 = 1.33$, $\alpha = 1.44779\ldots$.

Methods	$\lvert x_1 - \alpha\rvert$	$\lvert x_2 - \alpha\rvert$	$\lvert x_3 - \alpha\rvert$	r_c(56)
K–T $n = 2$	4.56(−3)	5.76(−9)	1.50(−32)	3.996
(57)	4.56(−3)	3.38(−10)	1.25(−41)	4.406
(59)	4.56(−3)	2.67(−11)	2.16(−52)	4.989
(60)	4.56(−3)	1.46(−12)	2.66(−63)	5.342
(61)	4.56(−3)	4.85(−13)	8.32(−73)	5.991
Z–L–H $n = 2$	8.84(−4)	1.84(−12)	3.48(−47)	3.999
(57)	8.84(−4)	1.52(−13)	5.94(−57)	4.444
(59)	8.84(−4)	8.89(−15)	8.83(−70)	5.001
(60)	8.84(−4)	1.93(−16)	3.95(−84)	5.346
(61)	8.84(−4)	1.34(−17)	2.03(−100)	5.993
K–T $n = 3$	7.71(−5)	8.06(−31)	1.14(−238)	7.999
(57)	7.71(−5)	2.77(−33)	9.03(−274)	8.454
(58)	7.71(−5)	1.73(−35)	3.46(−311)	8.995
(59)	7.71(−5)	3.93(−39)	5.44(−382)	9.998
(60)	7.71(−5)	1.73(−45)	1.27(−491)	10.975
(61)	7.71(−5)	1.07(−46)	1.55(−521)	11.344
(55)	7.71(−5)	2.98(−46)	3.13(−543)	12.001
Z–L–H $n = 3$	2.18(−6)	1.46(−44)	5.78(−350)	7.999
(57)	2.18(−6)	9.88(−47)	1.33(−388)	8.474
(58)	2.18(−6)	3.40(−49)	1.53(−434)	9.002
(59)	2.18(−6)	2.01(−55)	7.29(−546)	10.002
(60)	2.18(−6)	2.81(−61)	3.21(−665)	11.003
(61)	2.18(−6)	2.99(−67)	9.74(−754)	11.279
(55)	2.18(−6)	4.44(−66)	2.01(−782)	12.001

(57)–(61) can also be noticed from Tables 3 and 4: Newton's interpolation of higher degree evidently gives the best results, which is expected having in mind that this approach provides the highest order of convergence. From the last column of Tables 3 and 4 we observe that the computational order of convergence r_c, calculated by (56), matches very well the theoretical order given in Theorems 2–4.

We end this paper with the conclusion that the considerable increase of the R-order of convergence (even up to 50%, see Table 1) of families (21) and (22) with memory is attained without any additional function evaluations per iteration, indicating a very high computational efficiency of the proposed methods with memory. Finally, note that the order of n-point methods (21) and (22) with memory is higher than 2^n ($n \geq 2$), but it does not refute the Kung–Traub conjecture since this hypothesis is related only to the methods *without memory* such as (1) and (5).

Acknowledgement

This work was supported by the Serbian Ministry of Science under Grant 174022.

References

[1] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
[2] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, J. ACM 21 (1974) 643–651.
[3] Q. Zheng, J. Li, F. Huang, Optimal Steffensen-type families for solving nonlinear equations, Appl. Math. Comput. 217 (2011) 9592–9597.
[4] 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.
[5] 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.
[6] 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.
[7] J.M. Ortega, W.C. Rheiboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
[8] M.S. Petković, Remarks on "On a general class of multipoint root-finding methods of high computational efficiency", SIAM J. Numer. Math. 49 (2011) 1317–1319.
[9] B.I. Yun, A non-iterative method for solving non-linear equations, Appl. Math. Comput. 198 (2008) 691–699.