Applicable Analysis and Discrete Mathematics
available online at http://pefmath.etf.rs

Appl. Anal. Discrete Math. **5** (2011), 298–317. doi:10.2298/AADM110905021P

A FAMILY OF TWO-POINT METHODS WITH MEMORY FOR SOLVING NONLINEAR EQUATIONS

Miodrag S. Petković, Jovana Džunić, Ljiljana D. Petković

An efficient family of two-point derivative free methods with memory for solving nonlinear equations is presented. It is proved that the convergence order of the proposed family is increased from 4 to at least $2+\sqrt{6}\approx 4.45$, 5, $\frac{1}{2}\big(5+\sqrt{33}\,\big)\approx 5.37$ and 6, depending on the accelerating technique. The increase of convergence order is attained using a suitable accelerating technique by varying a free parameter in each iteration. The improvement of convergence rate is achieved without any additional function evaluations meaning that the proposed methods with memory are very efficient. Moreover, the presented methods are more efficient than all existing methods known in literature in the class of two-point methods and three-point methods of optimal order eight. Numerical examples and the comparison with the existing two-point methods are included to confirm theoretical results and high computational efficiency.

1. INTRODUCTION

The main goal and motivation in constructing iterative methods for solving nonlinear equations is to attain as high as possible order of convergence with minimal computational cost. The most efficient existing root-solvers are based on multipoint iterations, first studied in Traub's book [**16**] and some papers and books published in the 1960s and 1970s (see, e.g., [**2**], [**3**], [**4**], [**6**], [**8**], [**9**], [**11**]). Multipoint iterative methods have again became an interesting and challenging task at present since they overcome theoretical limits of one-point methods concerning

2010 Mathematics Subject Classification. 65H05.
Keywords and Phrases. Nonlinear equations, Iterative methods, Multipoint methods with memory, Acceleration of convergence, Computational efficiency.

the convergence order and computational efficiency. The highest possible computational efficiency of these methods is closely connected to the hypothesis of Kung and Traub [**6**] from 1974. They have conjectured that the order of convergence of any multipoint method without memory, consuming n function evaluations per iteration, cannot exceed the bound 2^{n-1} (called *optimal order*). Multipoint methods with this property are usually called *optimal methods*. An extensive (but not exhausting) list of optimal methods may be found, for example, in [**12**] and [**13**].

Let α be a simple real zero of a real function $f : D \subset \mathbb{R} \to \mathbb{R}$ and let x_0 be an initial approximation to α. In many practical situations it is preferable to avoid calculations of derivatives of f. First multipoint derivative free methods were developed by Kung and Traub in [**6**] in 1974 with arbitrary order 2^{n-1} $(n \geq 2)$ requiring $n+1$ function evaluations. For $n = 2$ one obtains the derivative free method

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

of Steffensen-type with quadratic convergence (see [**16**, p. 185]). Taking $n = 3$ in the Kung-Traub family, the following derivative free two-point family of fourth order methods

$$\begin{cases} y_k = x_k - \dfrac{\gamma f(x_k)^2}{f(x_k + \gamma f(x_k)) - f(x_k)}, \\[2ex] x_{k+1} = y_k - \dfrac{f(y_k) f(x_k + \gamma f(x_k))}{\big(f(x_k + \gamma f(x_k)) - f(y_k)\big) f[x_k, y_k]}, \end{cases} \quad (k = 0, 1, \ldots), \tag{2}$$

is obtained, where $f[x, y] = \big[f(x) - f(y)\big]/(x - y)$ is a divided difference and γ is a nonzero constant. The family (2) requires three function evaluations and has the fourth order of convergence, which means that it supports the Kung-Traub conjecture. Its efficiency index is $E(2) \approx 1.587$.

In this paper we follow a basic principle of numerical analysis that a genuine ranking of numerical algorithms can be attained using computational efficiency that is always directly proportional to the quality of an algorithm and inversely proportional to its computational cost. In the case of root-finders, very fast convergence or approximations of great accuracy are of irrelevant importance if their computational cost is too high.

The aim of this paper is to state a two-point family with memory of very high computational efficiency. We start from a family of two-point methods without memory with order 4, derived in [**14**], and increase the convergence order to $2+\sqrt{6} \approx 4.45$, 5, $\frac{1}{2}\big(5+\sqrt{33}\big) \approx 5.37$ and 6 (depending on the accelerating technique) without additional calculations. In this manner we obtain new methods for finding simple roots of nonlinear equations, whose computational efficiency is higher than the efficiency of existing methods known in literature in the class of two-point methods and even higher than the efficiency of optimal three-point methods of order eight.

The main idea is based on the use of suitable two-valued functions and the variation of a free parameter γ in each iterative step. This parameter is calculated using information from the current and previous iteration so that the developed methods may be regarded as *methods with memory* following TRAUB's classification [**16**, p. 8].

An additional motivation for studying methods with memory arises from a surprising fact that such classes of methods have been considered in literature very seldom in spite of their high computational efficiency. We cite pioneering results of TRAUB [**16**, pp. 185–187], the three-point method of NETA [**9**] and the recently developed method with memory [**14**] with the order $2+\sqrt{5}\approx 4.236$.

The paper is organized as follows. In Section 2 we present a family of two-point methods without memory, which is an extended version of the family derived in [**14**]. None calculation of derivatives are requested. A modification of the KUNG-TRAUB method (2) appears as a special case of this family. Using a varying parameter, which is recursively calculated in each iteration, we develop a family of two-point methods with memory. In Section 3 we state convergence theorems which show that the R-order of convergence of the proposed family with memory is at least $2+\sqrt{6}\approx 4.45$ if a standard secant approach is applied, at least five if a new method called *improved secant approach* is employed and even $\frac{1}{2}\big(5+\sqrt{33}\,\big)\approx 5.372$ and 6 using Newton's interpolatory polynomials of the second and third degree, respectively. We emphasize that the increase of the order of convergence is obtained without any additional function evaluations, which points to very high computational efficiency. Indeed, the efficiency index 1.71 of the proposed fifth order two-point methods with memory is higher than the efficiency index 1.68 of optimal three-point methods of order eight, while the methods with Newton's interpolation (with the efficiency indices 1.75 and 1.817) are even more efficient. Numerical examples and the comparison with the existing optimal two-point methods are given in Section 4.

2. FAMILY OF TWO-POINT METHODS WITH MEMORY

To construct a family of derivative-free two-point methods, let us start from the doubled Newton method

$$\text{(3)}\qquad \begin{cases} y_k = x_k - \dfrac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \dfrac{f(y_k)}{f'(y_k)} \end{cases} \qquad (k=0,1,\ldots)$$

and substitute derivatives $f'(x)$ and $f'(y)$ by suitable approximations. This is often used model in designing two-point methods. Let

$$\varphi(x)=\frac{f(x+\gamma f(x))-f(x)}{\gamma f(x)},$$

be a function that appears in the Steffensen-like method (1), where γ is an arbitrary real constant. Obviously, $\varphi(x)$ is an approximation to the first derivative $f'(x)$ assuming that $|f(x)|$ is small enough.

It is natural to approximate $f'(x) \approx \varphi(x)$ in (3). The derivative $f'(y)$ in the second step of (3) will be approximated by $f'(y) \approx \varphi(x)/h(u,v)$, where $h(u,v)$ is at least two-times differentiable function that depends on two real variables

$$u = u(x,y) = \frac{f(y)}{f(x)}, \quad v = v(x,y) = \frac{f(y)}{f(x+\gamma f(x))}. \tag{4}$$

Now from the iterative scheme (3) we state the family of two-point iterative methods

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{\varphi(x_k)}, \\ x_{k+1} = y_k - h(u_k, v_k)\dfrac{f(y_k)}{\varphi(x_k)}, \end{cases} \quad (k = 0, 1, \ldots), \tag{5}$$

where

$$u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(x_k + \gamma f(x_k))}.$$

The weight function h should be determined in such way that the order of convergence of the two-point method (5) is as high as possible, which is the subject of the following theorem.

Theorem 1. *If an initial approximation x_0 is sufficiently close to a zero α of f and the weight function h appearing in* (5) *satisfies the conditions*

$$\begin{aligned} &h(0,0) = h_u(0,0) = h_v(0,0) = 1, \quad h_{vv}(0,0) = 2, \\ &|h_{uu}(0,0)| < \infty, \quad |h_{uv}(0,0)| < \infty, \end{aligned} \tag{6}$$

then the error relation related to the family of two-point methods (5) *is given by*

$$\begin{aligned} \varepsilon_{k+1} &= x_{k+1} - \alpha \\ &= -a_2(1+\gamma f'(\alpha))^2 \Big[a_3 + a_2^2\big(-4 + h_{uu}(0,0)/2 + h_{uv}(0,0) \\ &\quad + (h_{uu}(0,0)/2 - 1)\gamma f'(\alpha)\big)\Big]\varepsilon_k^4 + O(\varepsilon_k^5). \end{aligned} \tag{7}$$

Proof. Introduce the abbreviations

$$\varepsilon_k = x_k - \alpha, \quad \varepsilon_{k,y} = y_k - \alpha, \quad q = \gamma f'(\alpha), \quad a_k = \frac{f^{(k)}(\alpha)}{k!f'(\alpha)} \ (k = 2, 3, \ldots).$$

In what follows we will derive the error relation (7) of the family of two-point methods (5), which is essential to our study.

Using Taylor series about the root α, we obtain

$$f(x_k) = f'(\alpha)\Big(\varepsilon_k + a_2\varepsilon_k^2 + a_3\varepsilon_k^3 + a_4\varepsilon_k^4\Big) + O(\varepsilon_k^5) \tag{8}$$

and

$$
\begin{aligned}
(9)\qquad f(x_k+\gamma f(x_k)) = f'(\alpha)\Big(&(1+q)\varepsilon_k + a_2(1+3q+q^2)\varepsilon_k^2\\
&+ (2a_2^2q(1+q) + a_3(1+4q+3q^2+q^3))\varepsilon_k^3\\
&+ (a_4(1+5q+6q^2+4q^3+q^4) + a_2^3q^2\\
&+ a_2a_3q(5+8q+3q^2))\varepsilon_k^4\Big) + O(\varepsilon_k^5).
\end{aligned}
$$

In view of (8) and (9) we find

$$
\begin{aligned}
(10)\qquad \varepsilon_{k,y} = y_k - \alpha &= \varepsilon_k - \frac{\gamma f(x_k)^2}{f(x_k+\gamma f(x_k)) - f(x_k)}\\
&= a_2(1+q)\varepsilon_k^2 + (-a_2^2(2+2q+q^2) + a_3(2+3q+q^2))\varepsilon_k^3\\
&\quad + (a_2^3(4+5q+3q^2+q^3) + a_4(3+6q+4q^2+q^3)\\
&\quad - a_2a_3(7+10q+7q^2+2q^3))\varepsilon_k^4 + O(\varepsilon_k^5).
\end{aligned}
$$

By (10) we get

$$
\begin{aligned}
(11)\qquad f(y_k) &= f'(\alpha)\Big(\varepsilon_{k,y} + a_2\varepsilon_{k,y}^2 + a_3\varepsilon_{k,y}^3 + a_4\varepsilon_{k,y}^4\Big) + O(\varepsilon_{k,y}^5)\\
&= f'(\alpha)\Big(a_2(1+q)\varepsilon_k^2 + (-a_2^2(2+2q+q^2) + a_3(2+3q+q^2))\varepsilon_k^3\\
&\quad +(a_2^3(5+7q+4q^2+q^3) + a_4(3+6q+4q^2+q^3)\\
&\quad - a_3(7+10q+7q^2+2q^3)))\varepsilon_k^4\Big) + O(\varepsilon_k^5).
\end{aligned}
$$

Using (8) and (9) we find $\varphi(x_k)$, and by (8), (9) and (11) we can express u_k and v_k given by (4). Assume that x_k is sufficiently close to the zero α, then u_k and v_k are close to 0. Hence, we can represent a two-valued function h occurring in (5) by Taylor's series about $(0,0)$ in the form

$$
\begin{aligned}
(12)\qquad h(u,v) = h(0,0) + h_u(0,0)u + h_v(0,0)v + \frac{h_{uu}(0,0)}{2}u^2\\
+ h_{uv}(0,0)uv + \frac{h_{vv}(0,0)}{2}v^2 + \cdots,
\end{aligned}
$$

where the subscript indices u and v indicate the appropriate partial derivatives. The error relation of the two-step iterative scheme (5) is

$$
(13)\qquad \varepsilon_{k+1} = x_{k+1} - \alpha = \varepsilon_{k,y} - h(u_k,v_k)\frac{f(y_k)}{\varphi(x_k)}.
$$

Using the conditions (6) and the developments (8)–(12), with the help of symbolic computation in the computer algebra system *Mathematica* we start from (13) and obtain the error relation (7).

REMARK 1. It was proved in [**14**] that the fourth order of the method (5) can be attained under the relaxed conditions

$$h(0,0) = h_u(0,0) = h_v(0,0) = 1. \tag{14}$$

The additional requirement $h_{vv}(0,0) = 2$ in Theorem 1 enables that the term $1 + \gamma f'(\alpha)$ in (5) is squared; this fact is of essential importance which will be shown later. Otherwise, the relaxed conditions (14) (assuming $h_{vv} \neq 2$) give only linear factor $1 + \gamma f'(\alpha)$ and, consequently, slower convergence, see [**14**].

We observe from (7) that the order of convergence of the family (5) is four when $\gamma \neq -1/f'(\alpha)$. If we could provide $\gamma = -1/f'(\alpha)$, the order of the family (5) would exceed four. However, the value $f'(\alpha)$ is not available in practice and such acceleration is not possible. Instead of that, we could use an approximation $\bar{f}'(\alpha) \approx f'(\alpha)$, calculated by available information. Then, setting $\gamma = -1/\bar{f}'(\alpha)$, we can achieve that the order of convergence of the modified method exceeds 4 without using any new function evaluations. Moreover, we will show in Section 3 that a special approximation of γ can produce two-point methods with memory which have the order six.

Henceforth, we will often write $w_k = x_k + \gamma_k f(x_k)$, for brevity. In this paper we will consider four methods for approximating $f'(\alpha)$:

(I) $\bar{f}'(\alpha) = \dfrac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$ (secant approach).

(II) $\bar{f}'(\alpha) = \dfrac{f(x_k) - f(y_{k-1})}{x_k - y_{k-1}}$ (improved secant approach).

(III) $\bar{f}'(\alpha) = N_2'(x_k)$ (Newton's interpolatory approach), where

$$N_2(t) = N_2(t; x_k, y_{k-1}, x_{k-1})$$

is Newton's interpolatory polynomial of second degree, set through three best available approximations (nodes) $x_k,\ y_{k-1}$ and x_{k-1}.

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

$$N_3(t) = N_3(t; x_k, y_{k-1}, x_{k-1}, w_{k-1})$$

is Newton's interpolatory polynomial of third degree, set through four best available approximations (nodes) $x_k,\ y_{k-1},\ x_{k-1}$ and w_{k-1}.

Then the parameter $\gamma = \gamma_k$ can be calculated recursively as the iteration proceeds as

$$\gamma_k = -\frac{1}{\bar{f}'(\alpha)} = -\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \quad \text{(method (I))}, \tag{15}$$

$$\gamma_k = -\frac{1}{\bar{f}'(\alpha)} = -\frac{x_k - y_{k-1}}{f(x_k) - f(y_{k-1})} \quad \text{(method (II))}, \tag{16}$$

$$\gamma_k = -\frac{1}{\bar f'(\alpha)} = -\frac{1}{N_2'(x_k)} \quad \text{(method (III))}, \tag{17}$$

$$\gamma_k = -\frac{1}{\bar f'(\alpha)} = -\frac{1}{N_3'(x_k)} \quad \text{(method (IV))}. \tag{18}$$

To calculate γ_k by (17) and (18), we need the expressions of N_2' and N_3'. Since

$$N_2(t) = f(x_k) + f[x_k, y_{k-1}](t - x_k) + f[x_k, y_{k-1}, x_{k-1}](t - x_k)(t - y_{k-1})$$

and

$$\begin{aligned} N_3(t) = f(x_k) &+ f[x_k, y_{k-1}](t - x_k) + f[x_k, y_{k-1}, x_{k-1}](t - x_k)(t - y_{k-1}) \\ &+ f[x_k, y_{k-1}, x_{k-1}, w_{k-1}](t - x_k)(t - y_{k-1})(t - x_{k-1}), \end{aligned}$$

we find

$$N_2'(x_k) = \Big[\frac{\mathrm{d}}{\mathrm{d}t} N_2(t)\Big]_{t=x_k} = f[x_k, y_{k-1}] + f[x_k, y_{k-1}, x_{k-1}](x_k - y_{k-1}) \tag{19}$$

and

$$\begin{aligned} N_3'(x_k) = \Big[\frac{\mathrm{d}}{\mathrm{d}t} N_3(t)\Big]_{t=x_k} &= f[x_k, y_{k-1}] + f[x_k, y_{k-1}, x_{k-1}](x_k - y_{k-1}) \\ &+ f[x_k, y_{k-1}, x_{k-1}, w_{k-1}](x_k - y_{k-1})(t - x_{k-1}). \end{aligned} \tag{20}$$

It is preferable to calculate divided differences of higher order by a recursive procedure using divided differences of lower order.

Remark 2. The secant methods (I) and (II) are, in fact, the derivatives $N_1'(x_k)$ of Newton's interpolatory polynomials of first order at the nodes x_k, x_{k-1} and x_k, y_{k-1}, respectively.

Remark 3. The accelerating method (15), actually Traub's method [**16**] from 1964, was used in [**14**] to increase the order from 4 to $2 + \sqrt{5} \approx 4.236$ under the conditions (14). The accelerating methods (16), (17) and (18), together with the additional condition $h_{vv}(0, 0) = 2$, are new, simple and very useful, providing considerable improvement of convergence rate without any additional function evaluations.

By defining γ recursively as the iteration proceeds using (15), (16), (17) or (18), we obtain a new derivative free two-point method *with memory* corresponding to (5),

$$\begin{cases} y_k = x_k - \dfrac{\gamma_k f(x_k)^2}{f(x_k + \gamma_k f(x_k)) - f(x_k)}, \\[2ex] x_{k+1} = y_k - h(u_k, v_k) \dfrac{\gamma_k f(x_k) f(y_k)}{f(x_k + \gamma_k f(x_k)) - f(x_k)}, \end{cases} \quad (k = 0, 1, \ldots). \tag{21}$$

We use the term *method with memory* following Traub's classification [**16**, p. 8] and the fact that the evaluation of the parameter γ_k depends on data available from the current and the previous iterative step. Accelerating methods obtained by recursively calculated free parameter may also be called *self-accelerating methods.* The initial value γ_0 should be chosen before starting the iterative process, for example, using one of ways proposed in [**16**, p. 186].

Note that the iterative scheme (21) defines a family of two-step methods. We can apply different two-valued weight functions h that satisfy the conditions (6). For convenience, recall the list of functions of simple form presented in [**14**]:

1) $h(u,v) = \dfrac{1+u}{1-v}$;
2) $h(u,v) = \dfrac{1}{(1-u)(1-v)}$;
3) $h(u,v) = 1+u+v+v^2$;
4) $h(u,v) = 1+u+v+(u+v)^2$;
5) $h(u,v) = u + \dfrac{1}{1-v}$.

Using simple rearrangement, it is easy to show that the choice $h(u,v) = \dfrac{1}{(1-u)(1-v)}$ gives the KUNG-TRAUB method (2) with memory as a special case,

$$\begin{cases} y_k = x_k - \dfrac{\gamma_k f(x_k)^2}{f(x_k+\gamma_k f(x_k)) - f(x_k)}, \\ x_{k+1} = y_k - \dfrac{f(y_k) f(x_k+\gamma_k f(x_k))}{\big(f(x_k+\gamma_k f(x_k)) - f(y_k)\big) f[x_k,y_k]}, \end{cases} \quad (k=0,1,\ldots). \tag{22}$$

3. CONVERGENCE THEOREMS

To estimate the convergence rate of the family of two-point methods (21), we will use the concept of the R-order of convergence introduced by ORTEGA and RHEINBOLDT [**10**]. The R-order of convergence of an iterative method (IM) that converges to the zero α, will be denoted with $O_R((\mathrm{IM}),\alpha)$.

To avoid higher order terms in some relations, which make only "parasite" parts of Taylor's expansions and do not influence the convergence order, we employ the notation used in TRAUB's book [**16**]: If $\{f_k\}$ and $\{g_k\}$ are null sequences and

$$\frac{f_k}{g_k} \to C,$$

where C is a nonzero constant, we shall write $f_k = O(g_k)$ or $f_k \sim C g_k$.

In our convergence analysis we will use the Bachman-Landau *a little o-notation*: For the sequences $\{\varphi_k\}$ and $\{\psi_k\}$ which tend to 0 when $k \to \infty$ we write $\varphi_k = o(\psi_k)$ if $\lim\limits_{k\to\infty} \varphi_k/\psi_k = 0$; in other words, φ is dominated by ψ asymptotically. Some auxiliary estimations necessary in this analysis are given in the following lemma.

Lemma 1. *Let N_m be the Newton interpolation polynomial of degree m that interpolates a function f at $m+1$ distinct interpolation nodes $t_0, t_1, \ldots, t_m$ contained in an interval I and the derivative $f^{(m+1)}$ is continuous in I. Assume that*

1) *all errors $e_j = t_j - \alpha$ are sufficiently small, that is, all nodes $t_0, t_1, \ldots, t_m$ are sufficiently close to the zero α;*

2) *the condition $e_0 = o\big(e_1 \cdot e_2 \cdots e_m\big)$ holds.*

Then

$$N_m'(t_0) \sim f'(\alpha)\bigg(1 + (-1)^{m+1} a_{m+1} \prod_{j=1}^{m} e_j\bigg). \tag{23}$$

Proof. The error of the Newton interpolation is given by the well known formula

$$f(t) - N_m(t) = \frac{f^{(m+1)}(d)}{(m+1)!} \prod_{j=0}^{m} (t - t_j), \quad (d \in I). \tag{24}$$

Differentiating (24) yields at the point $t = t_0$

$$N_m'(t_0) = f'(t_0) - \frac{f^{(m+1)}(d)}{(m+1)!} \prod_{j=1}^{m} (t_0 - t_j). \tag{25}$$

In the neighborhood of the zero α, the function f and its derivatives may be developed into Taylor series (for $t = t_0$),

$$\begin{aligned} f(t_0) &= f'(\alpha)\big(e_0 + a_2 e_0^2 + a_3 e_0^3 + \ldots\big), \\ f'(t_0) &= f'(\alpha)\big(1 + 2a_2 e_0 + 3a_3 e_0^2 + \ldots\big), \end{aligned} \tag{26}$$

$$f^{(m+1)}(d) = f'(\alpha)\Big((m+1)!a_{m+1} + \frac{(m+2)!}{1!} a_{m+2} e_d + \cdots\Big), \tag{27}$$

where $e_d = d - \alpha$. Substituting (26) and (27) into (25) and taking into account the conditions of Lemma 1, after short arrangement we arrive at the relation (23).

Remark 4. The condition 2) of Lemma 1 is typical for multipoint methods with memory. If $\{e_{k,j}\}_{j=0,1,\ldots,m}$ define iterative null sequences with orders $r_0, r_1, \ldots, r_m$, this condition means that $r_0 > r_1 + \cdots + r_m$.

First we state the convergence theorem of the family of two-point methods (21) with memory which uses the calculation of γ_k by (15).

Theorem 2. *Let the varying parameter γ_k in* (21) *be recursively calculated by* (15). *If an initial approximation x_0 is sufficiently close to a zero α of f, then the R-order of convergence of the two-point methods* (21) *is at least $2 + \sqrt{6} \approx 4.45$.*

Proof. Let r be the R-order of the two-point methods (21) with memory, then we may write

$$\varepsilon_{k+1} \sim D_{k,r}\varepsilon_k^r, \tag{28}$$

where $D_{k,r}$ tends to the asymptotic error constant of (21) when $k \to \infty$. From some convenient applications, we rewrite the error relation (7) in the form

$$\varepsilon_{k+1} \sim C_k(1+\gamma_k f'(\alpha))^2\varepsilon_k^4, \tag{29}$$

where

$$C_k = -a_2\Big[a_3 + a_2^2(-4 + h_{uu}(0,0)/2 + h_{uv}(0,0) + (h_{uu}(0,0)/2 - 1)\gamma_k f'(\alpha)\Big]$$

is now a varying quantity due to variable γ_k.

From Lemma 1 for $m = 1$ (see Remark 2) we have $e_0 = \varepsilon_k,\ e_1 = \varepsilon_{k-1}$ so that

$$N_1'(x_k) \sim f'(\alpha)\big(1 + a_2\varepsilon_{k-1}\big).$$

Hence

$$\gamma_k = -\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} = -\frac{1}{N_1'(x_k)} \sim \frac{1 - a_2\varepsilon_{k-1}}{f'(\alpha)}$$

so that

$$1 + \gamma_k f'(\alpha) \sim a_2\varepsilon_{k-1}. \tag{30}$$

Substituting (30) in (29) yields $\varepsilon_{k+1} \sim a_2^2 C_k\varepsilon_k^4\varepsilon_{k-1}^2$. Since $\varepsilon_{k+1} \sim D_{k,r}\varepsilon_k^r \sim D_{k,r}D_{k-1,r}^r\varepsilon_{k-1}^{r^2}$ and $\varepsilon_k^4\varepsilon_{k-1}^2 \sim D_{k-1,r}^4\varepsilon_{k-1}^{4r}\varepsilon_{k-1}^2$, equating exponents of ε_{k-1} in the last two relations we obtain the quadratic equation $r^2 - 4r - 2 = 0$. The positive root $r = 2 + \sqrt{6}$ of this equation determines the lower bound of the R-order of convergence of the method (21)–(15). □

To attain faster convergence, we now apply the improved secant approach (16) where a better approximation y_{k-1} is used instead of x_{k-1}. The R-order of the corresponding two-point family with memory in this case is the subject of the following theorem.

Theorem 3. *Let the varying parameter γ_k in* (21) *be recursively calculated by* (16)*. If an initial approximation x_0 is sufficiently close to a zero α of f, then the R-order of convergence of the two-point methods* (21) *is at least five.*

Proof. Calculating γ_k by (16) and using Lemma 1 for $m = 2,\ e_0 = \varepsilon_k,\ e_1 = \varepsilon_{k,y}$, we arrive at the following relation

$$1 + \gamma_k f'(\alpha) \sim a_2\varepsilon_{k-1,y}. \tag{31}$$

Assume that the R-order of convergence of the sequence of errors $\{\varepsilon_{k,y}\}$ is p, then we may write

$$\varepsilon_{k,y} \sim D_{k,p}\varepsilon_k^p \quad (2 \le p \le 3). \tag{32}$$

Hence, by (28) and (32), we obtain

$$\varepsilon_{k,y} \sim D_{k,p}\Big(D_{k-1,r}\varepsilon_{k-1}^{r}\Big)^{p} = D_{k,p}D_{k-1,r}^{p}\varepsilon_{k-1}^{rp}. \tag{33}$$

On the other hand, by combining (10), (28), (31) and (32) we find

$$\varepsilon_{k,y} \sim a_2(1+\gamma_k f'(\alpha))\varepsilon_k^2 \sim a_2^2\varepsilon_{k-1,y}\varepsilon_k^2 \sim a_2^2\Big(D_{k-1,p}\varepsilon_{k-1}^{p}\Big)\Big(D_{k-1,r}\varepsilon_{k-1}^{r}\Big)^2,$$

whence

$$\varepsilon_{k,y} \sim a_2^2 D_{k-1,p}D_{k-1,r}^2\varepsilon_{k-1}^{2r+p}. \tag{34}$$

From (28)–(32) we obtain

$$\begin{aligned}\varepsilon_{k+1} &\sim C_k\big(a_2\varepsilon_{k-1,y}\big)^2\varepsilon_k^4 \sim a_2^2 C_k\Big(D_{k-1,p}\varepsilon_{k-1}^{p}\Big)^2\Big(D_{k-1,r}\varepsilon_{k-1}^{r}\Big)^4\\ &= a_2^2 C_k D_{k-1,p}^2 D_{k-1,r}^4\varepsilon_{k-1}^{4r+2p}.\end{aligned} \tag{35}$$

On the other hand, we obtain from (28)

$$\varepsilon_{k+1} \sim D_{k,r}\Big(D_{k-1,r}\varepsilon_{k-1}^{r}\Big)^{r} = D_{k,r}D_{k-1,r}^{r}\varepsilon_{k-1}^{r^2}. \tag{36}$$

By comparing exponents of ε_{k-1} on the right-hand side of (33) and (34), and then on the right-hand side of (35) and (36), we form the following system of equations

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

with non-trivial solution $p=5/2$ and $r=5$. Therefore, the R-order of (21) is at least five. □

Finally, we wish to show that the R-order of the family (21) is even greater than 5 if the Newton interpolation polynomial of higher order is applied in the calculation of γ_k by (17) and (18). It is assumed that only available information are used.

Theorem 4. *Let the varying parameter γ_k in the iterative scheme* (21) *be recursively calculated by* (17). *If an initial approximation x_0 is sufficiently close to a zero α of f, then the R-order of convergence of the two-point methods* (21)–(17) *with memory is at least* $\frac{1}{2}\big(5+\sqrt{33}\big)\approx 5.372$.

Proof. In regard to Lemma 1, taking $m=2,\ e_0=\varepsilon_k,\ e_1=\varepsilon_{k-1,y},\ e_2=\varepsilon_{k-1}$ in (23) we obtain

$$N_2'(x_k) \sim f'(\alpha)\big(1-a_3\varepsilon_{k-1}\varepsilon_{k-1,y}\big).$$

From the last relation and (17) we find

$$1+\gamma_k f'(\alpha) \sim a_3\varepsilon_{k-1}\varepsilon_{k-1,y}. \tag{37}$$

Using (10), (28), (32) and (37), we obtain the following error relation

$$\begin{aligned}\varepsilon_{k,y} &\sim a_2(1+\gamma_k f'(\alpha))\varepsilon_k^2 \sim a_2a_3\varepsilon_{k-1}\varepsilon_{k-1,y}\varepsilon_k^2\\ &\sim a_2a_3\varepsilon_{k-1}(D_{k-1,p}\varepsilon_{k-1}^p)(D_{k-1,r}\varepsilon_{k-1}^r)^2\\ &= a_2a_3D_{k-1,p}D_{k-1,r}^2\varepsilon_{k-1}^{2r+p+1}.\end{aligned} \tag{38}$$

In the similar manner, in regard to (28), (29), (32) and (37), we find

$$\begin{aligned}\varepsilon_{k+1} &\sim C_k(1+\gamma_k f'(\alpha))^2\varepsilon_k^4 \sim C_k(a_3\varepsilon_{k-1}\varepsilon_{k-1,y})^2\varepsilon_k^4\\ &\sim C_ka_3^2\varepsilon_{k-1}^2(D_{k-1,p}\varepsilon_{k-1}^p)^2(D_{k-1,r}\varepsilon_{k-1}^r)^4\\ &= C_ka_3^2D_{k-1,p}^2D_{k-1,r}^4\varepsilon_{k-1}^{4r+2p+2}.\end{aligned} \tag{39}$$

Comparing the error exponents of ε_{k-1} in pairs of relations (33),(38) and (36),(39), we form the system of equations in p and r

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

Positive solutions of this system are $p=\frac{1}{4}(5+\sqrt{33}\,)$ and $r=\frac{1}{2}(5+\sqrt{33}\,)$, and we conclude that the lower bound of the R-order of the methods with memory (21)–(17) is at least $r=\frac{1}{2}(5+\sqrt{33}\,)\approx 5.372$. □

Even faster convergence can be obtained if γ_k is calculated by (18), without any additional computational cost, which is the subject of the following theorem.

Theorem 5. *Let the varying parameter γ_k in the iterative scheme* (21) *be recursively calculated by* (18). *If an initial approximation x_0 is sufficiently close to a zero α of f, then the R-order of convergence of the two-point methods* (21)–(18) *with memory is at least six.*

Proof. Let the errors appearing in the kth iteration be denoted with $\varepsilon_{k,y}=y_k-\alpha$, $\varepsilon_{k,w}=w_k-\alpha$, $\varepsilon_k=x_k-\alpha$. Take $e_0=\varepsilon_k,\ e_1=\varepsilon_{k-1,y},\ e_2=\varepsilon_{k-1},\ e_3=\varepsilon_{k-1,w}$ in (23), according to Lemma 1 we have $N_3'(x_k)\sim f'(\alpha)\big(1+a_4\varepsilon_{k-1}\varepsilon_{k-1,y}\varepsilon_{k-1,w}\big)$. Hence

$$1+\gamma_k f'(\alpha)\sim a_4\varepsilon_{k-1}\varepsilon_{k-1,y}\varepsilon_{k-1,w}. \tag{40}$$

As in the previous analysis, following (33) and (36) we may write

$$\varepsilon_{k+1}\sim D_{k,r}\varepsilon_k^r\sim D_{k,r}D_{k-1,r}^r\varepsilon_{k-1}^{r^2}, \tag{41}$$

$$\varepsilon_{k,y}\sim D_{k,q}\varepsilon_k^q\sim D_{k,q}D_{k-1,r}^r\varepsilon_{k-1}^{rq}, \tag{42}$$

$$\varepsilon_{k,w}\sim D_{k,p}\varepsilon_k^p\sim D_{k,p}D_{k-1,r}^r\varepsilon_{k-1}^{rp}. \tag{43}$$

Furthermore, from (8), (10) and (29) we have

$$\varepsilon_{k+1}\sim C_k(1+\gamma_k f'(\alpha))^2\varepsilon_k^4, \tag{44}$$

$$\varepsilon_{k,y} \sim a_2(1+\gamma_k f'(\alpha))\varepsilon_k^2, \tag{45}$$

$$\varepsilon_{k,w} \sim (1+\gamma_k f'(\alpha))\varepsilon_k. \tag{46}$$

By combining the above expressions (40)–(46) we derive the following error relations

$$\begin{aligned}\varepsilon_{k+1} &\sim C_k(1+\gamma_k f'(\alpha))^2\varepsilon_k^4 \\ &\sim C_k(a_4\varepsilon_{k-1}\varepsilon_{k-1,y}\varepsilon_{k-1,w})^2\varepsilon_k^4 \sim C_k a_4^2 D_{k-1,r}^4 D_{k-1,p}^2 D_{k-1,q}^2 \varepsilon_{k-1}^{4r+2p+2q+2},\end{aligned} \tag{47}$$

$$\begin{aligned}\varepsilon_{k,y} &\sim a_2(1+\gamma_k f'(\alpha))\varepsilon_k^2 \\ &\sim a_2(a_4\varepsilon_{k-1}\varepsilon_{k-1,y}\varepsilon_{k-1,w})\varepsilon_k^2 \sim a_2 a_4 D_{k-1,r}^2 D_{k-1,p} D_{k-1,q}\varepsilon_{k-1}^{2r+p+q+1},\end{aligned} \tag{48}$$

$$\begin{aligned}\varepsilon_{k,w} &\sim (1+\gamma_k f'(\alpha))\varepsilon_k \\ &\sim (a_4\varepsilon_{k-1}\varepsilon_{k-1,y}\varepsilon_{k-1,w})\varepsilon_k \sim a_4 D_{k-1,r} D_{k-1,p} D_{k-1,q}\varepsilon_{k-1}^{r+p+q+1}.\end{aligned} \tag{49}$$

Equating appropriate exponents of ε_{k-1} in pairs of relations (41),(47), then (42),(48) and (43),(49), we arrive at the following system of equations in p, q and r,

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

with the positive solution $p=2, q=3$ and $r=6$. Hence we conclude that the lower bound of the R-order of the methods with memory (21)–(18) is at least six. □

Theorems 2, 3, 4 and 5 give the lower bound of the R-order of convergence of the family (21) in the case of the accelerating approaches (I)–(IV). We observe that the methods (21) with memory are considerably accelerated (even up to 50%) related to the corresponding methods (5) without memory. The main advantage of the presented methods with memory is their very high computational efficiency, significantly higher than the efficiency of the existing two-point methods and even higher than the efficiency of three-point methods of optimal order eight.

4. NUMERICAL EXAMPLES

We compared the family of two-point methods (21) with memory with several optimal two-point iterative methods (IM) of the fourth order which also require three function evaluations. First, we give a list of these methods, where the abbreviation $\nu(x)=f(x)/f'(x)$ is used.

King*'s family* [**4**]:

$$\begin{aligned} x_{k+1} = K_f(\beta; x_k) :& \\ = x_k - \nu(x_k) - \frac{f\big(x_k-\nu(x_k)\big)}{f'(x_k)} &\cdot \frac{f(x_k)+\beta f\big(x_k-\nu(x_k)\big)}{f(x_k)+(\beta-2)f\big(x_k-\nu(x_k)\big)},\end{aligned} \tag{50}$$

where β is a parameter. Let us note that King's family gives the following special cases:

Ostrowski*'s method* [**11**], $\beta = 0$:

$$K(0; x_k) = x_k - \nu(x_k) - \frac{\nu(x_k) f(x_k - \nu(x_k))}{f(x_k) - 2f(x_k - \nu(x_k))};$$

Kou-Li-Wang*'s method* [**5**], $\beta = 1$:

$$K(1; x_k) = x_k - \frac{f(x_k)^2 + [f(x_k - \nu(x_k))]^2}{f'(x_k)[f(x_k) - f(x_k - \nu(x_k))]};$$

Chun*'s method* [**1**], $\beta = 2$:

$$K(2; x_k) = x_k - \nu(x_k) \left\{ 1 + \frac{f(x_k - \nu(x_k))}{f(x_k)} + \frac{2\big[f(x_k - \nu(x_k))\big]^2}{f(x_k)^2} \right\}.$$

Jarratt*'s method* [**2**]:

$$x_{k+1} = x_k - \frac{1}{2}\nu(x_k) + \frac{f(x_k)}{f'(x_k) - 3f'\Big(x_k - \frac{2}{3}\nu(x_k)\Big)}. \tag{51}$$

Maheshwari*'s method* [**7**]:

$$x_{k+1} = x_k - \nu(x_k) \left\{ \frac{\big[f\big(x_k - \nu(x_k)\big)\big]^2}{f(x_k)^2} - \frac{f(x_k)}{f\big(x_k - \nu(x_k)\big) - f(x_k)} \right\}. \tag{52}$$

Ren-Wu-Bi *method* [**15**]:

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f[x_k, z_k]}, \quad z_k = x_k + f(x_k), \\[2ex] x_{k+1} = y_k - \dfrac{f(y_k)}{f[x_k, y_k] + f[y_k, z_k] - f[x_k, z_k] + a(y_k - x_k)(y_k - z_k)}. \end{cases} \tag{53}$$

Kung-Traub*'s two-point method with derivative* [**6**]:

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f'(x_k)}, \\[2ex] x_{k+1} = y_k - \dfrac{f(x_k)^2 f(y_k)}{f'(x_k)\big(f(y_k) - f(x_k)\big)^2}. \end{cases} \tag{54}$$

We applied the methods (21) and (50)–(54) to the following functions:

Example	Function	Root α	Initial guess x_0
1	$e^{-x^2+x+2} - \cos(x+1) + x^3 + 1$	-1	-0.5
2	$e^x \sin 5x - 2$	$1.3639731802\ldots$	1.2
3	$\log(x^2 + x + 2) - x + 1$	$4.1525907367\ldots$	3.2
4	$e^x \sin x + \log(x^2 + 1)$	0	0.3

We employed the computer algebra system *Mathematica* with multiple-precision arithmetic. The errors $|x_k - \alpha|$ for the first four iterations are given in Tables 1–4, where the denotation $A(-h)$ means $A \times 10^{-h}$ and K-T $(\cdot)$ is the abbreviation for Kung-Traub's methods (22) and (54).

It is evident that approximations to the roots given in Tables 1-4 possess great accuracy. Results of the fourth iteration are given only for demonstration of convergence speed of the tested methods and they are not required for practical problems at present. Initial value $\gamma_0 = 0.01$ was used.

Methods	$\|x_1 - \alpha\|$	$\|x_2 - \alpha\|$	$\|x_3 - \alpha\|$	$\|x_4 - \alpha\|$
King's IM (50) $\beta = 0$	4.26(−4)	2.12(−15)	1.31(−60)	1.93(−241)
King's IM (50) $\beta = 1$	2.57(−3)	2.44(−12)	1.99(−48)	8.80(−193)
King's IM (50) $\beta = 2$	4.79(−3)	2.42(−11)	1.58(−44)	2.91(−177)
Jarratt's IM (51)	2.27(−3)	2.04(−12)	1.34(−48)	2.50(−193)
Maheshwari's IM (52)	3.68(−3)	9.35(−12)	3.90(−46)	1.18(−183)
Ren-Wu-Bi's IM (53)	1.50(−3)	1.63(−11)	2.26(−43)	8.23(−171)
K-T (22), $\gamma_k = 0.01$	1.68(−3)	5.39(−13)	5.73(−51)	7.28(−203)
K-T (22), γ_k by (15)	1.68(−3)	9.36(−15)	3.70(−65)	2.76(−289)
K-T (22), γ_k by (16)	1.68(−3)	1.17(−16)	1.68(−83)	1.03(−417)
K-T (22), γ_k by (17)	1.68(−3)	6.27(−17)	1.33(−89)	7.75(−480)
K-T (22), γ_k by (18)	1.68(−3)	1.81(−17)	4.71(−103)	1.48(−616)
K-T (54)	1.30(−3)	1.73(−13)	5.37(−53)	5.02(−211)
(21)–(15) $h = (1+u)/(1-v)$	3.18(−3)	1.20(−13)	3.51(−60)	3.63(−267)
(21)–(16) $h = (1+u)/(1-v)$	3.18(−3)	1.51(−15)	6.05(−78)	6.26(−390)
(21)–(17) $h = (1+u)/(1-v)$	3.18(−3)	7.31(−16)	1.02(−83)	2.86(−448)
(21)–(18) $h = (1+u)/(1-v)$	3.18(−3)	2.02(−16)	9.31(−97)	8.80(−579)
(21)–(15) $h = 1+u+v+v^2$	4.51(−3)	4.48(−13)	1.29(−57)	8.68(−256)
(21)–(16) $h = 1+u+v+v^2$	4.51(−3)	5.64(−15)	4.52(−75)	1.46(−375)
(21)–(17) $h = 1+u+v+v^2$	4.51(−3)	2.51(−15)	9.85(−81)	2.82(−432)
(21)–(18) $h = 1+u+v+v^2$	4.51(−3)	6.71(−16)	1.23(−93)	4.93(−560)
(21)–(15) $h = 1+u+v+(u+v)^2$	1.31(−3)	3.72(−15)	5.90(−67)	3.01(−297)
(21)–(16) $h = 1+u+v+(u+v)^2$	1.31(−3)	4.61(−17)	1.61(−85)	8.39(−428)
(21)–(17) $h = 1+u+v+(u+v)^2$	1.31(−3)	2.54(−17)	8.83(−92)	1.68(−490)
(21)–(18) $h = 1+u+v+(u+v)^2$	1.31(−3)	7.40(−18)	2.12(−105)	1.23(−630)
(21)–(15) $h = u+1/(1-v)$	4.37(−3)	3.95(−13)	7.30(−58)	6.83(−257)
(21)–(16) $h = u+1/(1-v)$	4.37(−3)	4.99(−15)	2.43(−75)	6.48(−377)
(21)–(17) $h = u+1/(1-v)$	4.37(−3)	2.22(−15)	5.02(−81)	7.54(−433)
(21)–(18) $h = u+1/(1-v)$	5.92(−16)	6.09(−94)	7.11(−81)	7.54(−562)

Table 1. $f_1(x) = e^{-x^2+x+2} - \cos(x+1) + x^3 + 1, \quad \alpha = -1, \quad x_0 = -0.5$

According to the results presented in Tables 1–4 and a number of numerical examples, we can conclude that the convergence behavior of the two-point methods (21) with memory (including the modified Kung-Traub method (22)), based on the self-correcting parameter γ_k recursively calculated by (15)–(18), is considerably

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$
King's IM (50) $\beta=0$	3.53(−3)	3.22(−10)	1.95(−38)	2.62(−151)
King's IM (50) $\beta=1$	5.34(−3)	1.82(−9)	2.10(−35)	3.74(−139)
King's IM (50) $\beta=2$	7.80(−3)	9.59(−9)	1.72(−32)	1.78(−127)
Jarratt's IM (51)	3.25(−3)	2.37(−10)	6.24(−39)	2.98(−153)
Maheshwari's IM (52)	6.57(−3)	4.48(−9)	7.98(−34)	8.04(−133)
Ren-Wu-Bi's IM (53)	diverges	−	−	−
Ren-Wu-Bi's IM (53) [a] $x_0=1.4$	1.85(−2)	3.31(−4)	9.35(−12)	5.42(−42)
K-T (22), $\gamma_k=0.01$	7.28(−3)	9.15(−9)	1.85(−32)	3.08(−127)
K-T (22), γ_k by (15)	7.28(−3)	1.33(−11)	5.41(−49)	4.13(−216)
K-T (22), γ_k by (16)	7.28(−3)	4.17(−12)	2.89(−59)	4.29(−295)
K-T (22), γ_k by (17)	7.28(−3)	5.34(−12)	7.66(−61)	2.31(−323)
K-T (22), γ_k by (18)	7.28(−3)	2.62(−13)	2.75(−76)	3.74(−454)
K-T (54)	4.31(−3)	7.23(−10)	5.11(−37)	1.27(−145)
(21)–(15) $h=(1+u)/(1-v)$	9.78(−3)	4.12(−11)	9.44(−47)	3.68(−206)
(21)–(16) $h=(1+u)/(1-v)$	9.78(−3)	1.45(−11)	1.53(−56)	1.81(−281)
(21)–(17) $h=(1+u)/(1-v)$	9.78(−3)	1.78(−11)	5.85(−58)	6.68(−308)
(21)–(18) $h=(1+u)/(1-v)$	9.78(−3)	1.22(−12)	2.76(−72)	3.86(−430)
(21)–(15) $h=1+u+v+v^2$	1.20(−2)	9.26(−11)	3.91(−45)	5.65(−199)
(21)–(16) $h=1+u+v+v^2$	1.20(−2)	3.65(−11)	1.54(−54)	1.84(−271)
(21)–(17) $h=1+u+v+v^2$	1.20(−2)	4.31(−11)	7.34(−56)	1.22(−296)
(21)–(18) $h=1+u+v+v^2$	1.20(−2)	2.94(−12)	3.07(−69)	7.37(−412)
(21)–(15) $h=1+u+v+(u+v)^2$	1.26(−3)	1.29(−14)	1.20(−62)	9.17(−277)
(21)–(16) $h=1+u+v+(u+v)^2$	1.26(−3)	3.05(−15)	5.69(−75)	1.27(−373)
(21)–(17) $h=1+u+v+(u+v)^2$	1.26(−3)	4.44(−15)	8.55(−78)	2.77(−414)
(21)–(18) $h=1+u+v+(u+v)^2$	1.26(−3)	8.69(−17)	3.77(−97)	2.49(−579)
(21)–(15) $h=u+1/(1-v)$	1.25(−2)	1.10(−10)	8.52(−45)	1.78(−197)
(21)–(16) $h=u+1/(1-v)$	1.25(−2)	4.43(−11)	4.10(−54)	2.49(−269)
(21)–(17) $h=u+1/(1-v)$	1.25(−2)	5.19(−11)	2.06(−55)	3.07(−294)
(21)–(18) $h=u+1/(1-v)$	1.25(−2)	5.00(−12)	1.28 − 68)	3.77(−408)

Table 2. $f_2(x)=e^x\sin 5x-2,\quad \alpha=1.36397318026\ldots,\quad x_0=1.2$

better than the existing methods (2) and (50)–(54) for most examples. It is obvious from these tables that recursive calculation by the Newton interpolation (18) gives the best results. Having in mind that all of the tested methods have the same computational cost, we can conclude that the family of methods (21) with memory

[a] The method (53) is divergent for $x_0=1.2$, but it converges for the closer approximation $x_0=1.4$.

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	$\lvert x_4-\alpha\rvert$
King's IM (50) $\beta=0$	1.01(−3)	6.13(−16)	9.81(−65)	5.73(−260)
King's IM (50) $\beta=1$	2.12(−3)	2.14(−14)	2.25(−58)	2.69(−234)
King's IM (50) $\beta=2$	3.44(−3)	2.09(−13)	2.85(−54)	9.87(−218)
Jarratt's IM (51)	1.08(−3)	9.57(−16)	5.82(−64)	7.94(−257)
Maheshwari's IM (52)	2.78(−3)	7.62(−14)	4.32(−56)	4.46(−225)
Ren-Wu-Bi's IM (53)	1.57(−4)	5.98(−20)	1.25(−81)	2.37(−328)
K-T (22), $\gamma_k=0.01$	1.50(−3)	4.17(−15)	2.50(−61)	3.21(−246)
K-T (22), γ_k by (15)	1.50(−3)	1.98(−17)	1.05(−78)	1.44(−351)
K-T (22), γ_k by (16)	1.50(−3)	9.12(−20)	8.36(−101)	5.41(−506)
K-T (22), γ_k by (17)	1.50(−3)	8.05(−22)	1.20(−118)	2.60(−639)
K-T (22), γ_k by (18)	1.50(−3)	8.45(−23)	3.63(−138)	2.30(−830)
K-T (54)	1.52(−3)	4.46(−15)	3.30(−61)	9.95(−246)
(21)–(15) $h=(1+u)/(1-v)$	2.09(−3)	7.36(−17)	3.91(−76)	3.86(−340)
(21)–(16) $h=(1+u)/(1-v)$	2.09(−3)	3.46(−19)	6.56(−98)	1.61(−491)
(21)–(17) $h=(1+u)/(1-v)$	2.09(−3)	2.52(−21)	7.08(−116)	1.80(−624)
(21)–(18) $h=(1+u)/(1-v)$	2.09(−3)	5.00(−22)	1.56(−133)	1.47(−802)
(21)–(15) $h=1+u+v+v^2$	2.68(−3)	2.52(−16)	1.11(−73)	3.76(−328)
(21)–(16) $h=1+u+v+v^2$	2.68(−3)	1.19(−18)	3.11(−95)	3.89(−478)
(21)–(17) $h=1+u+v+v^2$	2.68(−3)	6.33(−21)	1.28(−113)	2.15(−612)
(21)–(18) $h=1+u+v+v^2$	2.68(−3)	2.84(−21)	5.57(−129)	3.20(−775)
(21)–(15) $h=1+u+v+(u+v)^2$	5.69(−4)	3.04(−19)	6.26(−87)	6.18(−371)
(21)–(16) $h=1+u+v+(u+v)^2$	5.69(−4)	1.41(−21)	7.29(−110)	2.73(−551)
(21)–(17) $h=1+u+v+(u+v)^2$	5.69(−4)	1.58(−23)	5.13(−128)	1.39(−689)
(21)–(18) $h=1+u+v+(u+v)^2$	5.69(−4)	5.49(−25)	2.78(−151)	4.59(−909)
(21)–(15) $h=u+1/(1-v)$	2.73(−3)	2.74(−16)	1.64(−73)	2.07(−328)
(21)–(16) $h=u+1/(1-v)$	2.73(−3)	1.28(−18)	4.63(−95)	2.82(−477)
(21)–(17) $h=u+1/(1-v)$	2.73(−3)	7.44(−21)	2.71(−113)	1.29(−610)
(21)–(18) $h=u+1/(1-v)$	2.73(−3)	2.76(−21)	4.58(−129)	9.58(−776)

Table 3. $f_3(x)=\log(x^2+x+2)-x+1, \quad \alpha=4.1525907367\ldots, \quad x_0=3.2$

is the most efficient. More precisely, calculating the computational efficiency of an iterative method (IM) of the order r, requiring θ function evaluations, by Ostrowski-Traub's formula $E(IM)=r^{1/\theta}$ (see [**11**, p. 20], [**16**, Appendix C]), we find

$$E(2)=E(50)=E(51)=E(52)=E(53)=E(54)=4^{1/3}\approx 1.587$$

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$
KING's IM (50) $\beta=0$	1.06(−2)	8.74(−8)	4.28(−28)	2.46(−109)
KING's IM (50) $\beta=1$	1.80(−2)	2.00(−6)	3.78(−22)	4.78(−85)
KING's IM (50) $\beta=2$	2.24(−2)	7.37(−6)	1.16(−19)	7.15(−75)
JARRATT's IM (51)	1.05(−2)	8.32(−8)	3.49(−28)	1.08(−109)
MAHESHWARI's IM (52)	2.02(−2)	4.06(−6)	8.52(−21)	1.65(−79)
REN-WU-BI's IM (53)	2.92(−2)	1.67(−5)	2.29(−18)	8.12(−70)
K-T (22), $\gamma_k=0.01$	1.55(−2)	7.91(−7)	6.11(−24)	2.18(−92)
K-T (22), γ_k by (15)	1.55(−2)	1.14(−7)	2.32(−30)	2.33(−131)
K-T (22), γ_k by (16)	1.55(−2)	1.66(−8)	2.08(−38)	6.25(−188)
K-T (22), γ_k by (17)	1.55(−2)	6.04(−10)	8.01(−51)	5.33(−272)
K-T (22), γ_k by (18)	1.55(−2)	6.13(−10)	3.03(−54)	4.46(−320)
K-T (54)	1.52(−2)	7.20(−7)	4.12(−24)	4.43(−93)
(21)–(15) $h=(1+u)/(1-v)$	1.84(−2)	2.57(−7)	8.49(−29)	2.09(−124)
(21)–(16) $h=(1+u)/(1-v)$	1.84(−2)	3.41(−8)	6.99(−37)	2.68(−180)
(21)–(17) $h=(1+u)/(1-v)$	1.84(−2)	1.69(−9)	1.87(−48)	2.89(−257)
(21)–(18) $h=(1+u)/(1-v)$	1.84(−2)	1.71(−9)	1.43(−51)	4.97(−304)
(21)–(15) $h=1+u+v+v^2$	2.10(−2)	6.20(−7)	5.72(−27)	3.82(−116)
(21)–(16) $h=1+u+v+v^2$	2.10(−2)	9.06(−8)	8.97(−35)	9.28(−170)
(21)–(17) $h=1+u+v+v^2$	2.10(−2)	7.67(−9)	3.08(−45)	7.49(−240)
(21)–(18) $h=1+u+v+v^2$	2.10(−2)	7.72(−9)	2.58(−47)	3.57(−278)
(21)–(15) $h=1+u+v+(u+v)^2$	1.57(−2)	6.92(−8)	1.56(−31)	8.34(−137)
(21)–(16) $h=1+u+v+(u+v)^2$	1.57(−2)	1.11(−8)	2.03(−39)	5.57(−193)
(21)–(17) $h=1+u+v+(u+v)^2$	1.57(−2)	7.01(−10)	7.22(−51)	4.40(−279)
(21)–(18) $h=1+u+v+(u+v)^2$	1.57(−2)	7.09(−10)	7.43(−54)	9.83(−318)
(21)–(15) $h=u+1/(1-v)$	2.06(−2)	5.56(−7)	3.55(−27)	4.56(−117)
(21)–(16) $h=u+1/(1-v)$	2.06(−2)	7.67(−8)	4.25(−35)	2.21(−171)
(21)–(17) $h=u+1/(1-v)$	2.06(−2)	4.96(−9)	4.91(−46)	3.13(−243)
(21)–(18) $h=u+1/(1-v)$	2.06(−2)	5.00(−9)	1.39(−48)	6.37(−286)

TABLE 4. $f_4(x)=e^x\sin x+\log(x^2+1),\quad \alpha=0,\quad x_0=0.3$

and

$$E((21)_{(15)})=\big(2+\sqrt{6}\,\big)^{1/3}\approx 1.645,\quad E((21)_{(16)})=5^{1/3}\approx 1.71,$$
$$E((21)_{(17)})=\Big(\frac{1}{2}\big(5+\sqrt{33}\,\big)\Big)^{1/3}\approx 1.75,\quad E((21)_{(18)})=6^{1/3}\approx 1.817.$$

Note that the last four entries do not refute the KUNG-TRAUB conjecture because they are related to the methods *with memory*, which are not the subject of the

Kung-Traub conjecture. Also, note that the efficiency indices $E((21)_{(16)}) \approx 1.71$, $E((21)_{(17)}) \approx 1.75$ and $E((21)_{(18)}) \approx 1.817$ are even higher than the efficiency index ($8^{1/4} \approx 1.68$) of optimal three-point methods (without memory) of order eight.

We end this paper with the comment on the importance of the choice of initial approximations. If they are chosen sufficiently close to the sought roots, then the expected (theoretical) convergence speed will be reached in practice; otherwise, all multipoint methods (and, in general, all iterative root-finding methods) show slower convergence, especially at the beginning of the iterative process. For this reason, a special attention should be paid to finding good initial approximations. We note that an efficient method for the determination of initial approximations of great accuracy was recently proposed in the excellent paper [**17**]. The well known fact that the accuracy of obtained approximations strongly depends on initial approximations and the structure and form of tested functions can be observed in the case of $f_4(x) = e^x \sin x + \log(x^2+1)$; all obtained approximations (Table 4) are considerably worse compared to those produced in the remaining examples.

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

REFERENCES

1. C. Chun: *Some fourth-order iterative methods for solving nonlinear equations.* Appl. Math. Comput., **195** (2008), 454–459.
2. P. Jarratt: *Some fourth order multipoint methods for solving equations.* Math. Comp., **20** (1966), 434–437.
3. P. Jarratt: *Some efficient fourth-order multipoint methods for solving equations.* BIT, **9** (1969), 119–124.
4. R. King: *A family of fourth order methods for nonlinear equations.* SIAM J. Numer. Anal., **10** (1973), 876–879.
5. J. Kou, Y. Li, X. Wang: *A composite fourth-order iterative method.* Appl. Math. Comput., **184** (2007), 471–475.
6. H. T. Kung, J. F. Traub: *Optimal order of one-point and multipoint iteration.* J. ACM, **21** (1974), 643–651.
7. A. K. Maheshwari: *A fourth-order iterative method for solving nonlinear equations.* Appl. Math. Comput., **211** (2009), 383–391.
8. B. Neta: *On a family of multipoint methods for nonlinear equations.* Int. J. Comput. Math., **9** (1981), 353–361.
9. B. Neta: *A new family of higher order methods for solving nonlinear equations.* Int. J. Comput. Math., **14** (1983), 191–195.
10. J. M. Ortega, W. C. Rheiboldt: *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, 1970.
11. A. M. Ostrowski: *Solution of Equations and Systems of Equations.* Academic Press, New York, 1960.
12. M. S. Petković: *On a general class of multipoint root-finding methods of high computational efficiency.* SIAM. J. Numer. Anal., **47** (2010), 4402–4414.

13. 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.

14. 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.

15. H. Ren, Q. Wu, W. Bi: *A class of two-step Steffensen type methods with fourth-order convergence.* Appl. Math. Comput., **209** (2009), 206–210.

16. J. F. Traub: *Iterative Methods for the Solution of Equations.* Prentice-Hall, New Jersey, 1964.

17. B. I. Yun: *A non-iterative method for solving non-linear equations.* Appl. Math. Comput., **198** (2008), 691–699.

Department of Mathematics,
Faculty of Electronic Engineering,
University of Niš, 18000 Niš,
Serbia
E-mails: msp@junis.ni.ac.rs,
jovana.dzunic@elfak.ni.ac.rs

(Received August 15, 2011)
(Revised September 1, 2011)

Department of Mathematics,
Faculty of Mechanical Engineering,
University of Niš, 18000 Niš,
Serbia
E-mail: ljiljana@masfak.ni.ac.rs