Applied Mathematics and Computation 217 (2010) 1887–1895

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Derivative free two-point methods with and without memory for solving nonlinear equations

M.S. Petković [a,*], S. Ilić [b], J. Džunić [a]

[a] Faculty of Electronic Engineering, Department of Mathematics, University of Niš, 18000 Niš, Serbia
[b] Faculty of Science, Department of Mathematics, University of Niš, 18000 Niš, Serbia

ARTICLE INFO

Keywords:
Multipoint iterative methods
Nonlinear equations
Derivative free methods
Methods with memory
Computational efficiency

ABSTRACT

Two families of derivative free two-point iterative methods for solving nonlinear equations are constructed. These methods use a suitable parametric function and an arbitrary real parameter. It is proved that the first family has the convergence order four requiring only three function evaluations per iteration. In this way it is demonstrated that the proposed family without memory supports the Kung–Traub hypothesis (1974) on the upper bound 2^n of the order of multipoint methods based on $n + 1$ function evaluations. Further acceleration of the convergence rate is attained by varying a free parameter from step to step using information available from the previous step. This approach leads to a family of two-step self-accelerating methods with memory whose order of convergence is at least $2+\sqrt{5}\approx 4.236$ and even $2+\sqrt{6}\approx 4.449$ in special cases. The increase of convergence order is attained without any additional calculations so that the family of methods with memory possesses a very high computational efficiency. Numerical examples are included to demonstrate exceptional convergence speed of the proposed methods using only few function evaluations.

1. Introduction

Multipoint iterative methods for solving nonlinear equations are of great practical importance since they overcome theoretical limits of one-point methods concerning the convergence order and computational efficiency. Although this type of root-finding methods were extensively studied in Traub's book [1] and some papers and books published in the 1960s and 1970s (see, e.g., [2–7]), the interest for multipoint methods has renewed in the first decade of the 21st century.

The main goal and motivation in constructing root-solvers is to achieve as high as possible convergence order consuming as small as possible function evaluations. In the case of multipoint methods, this requirement is closely connected with results of Kung and Traub [6] who conjectured that the order of convergence of any multipoint method without memory, consuming $n + 1$ function evaluations per iteration, cannot exceed the bound 2^n (called *optimal order*). Multipoint methods with this property are usually called *optimal methods*. An extensive list of optimal methods can be found, for example, in [8]. Defining the computational efficiency of a root-finding method by the *efficiency index* $E = r^{1/\theta}$, where r is the order of convergence and θ is the number of function evaluations per iteration (see [2, p. 20]), *optimal computational efficiency* is $2^{n/(n+1)}$ for optimal methods. Therefore, the efficiency index of optimal two-point methods is $2^{2/3}\approx 1.587$.

Let α be a simple real root of a real function f: $D\subset \boldsymbol{R}\to\boldsymbol{R}$ and let x_0 be an initial approximation to α. In many practical situations it is preferable to avoid calculations of derivatives of f. For example, a classical Steffensen's method [9]

* Corresponding author.
E-mail addresses: msp@eunet.rs (M.S. Petković), sneska@pmf.ni.ac.rs (S. Ilić), jovana.dzunic@elfak.ni.ac.rs (J. Džunić).

0096-3003/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2010.06.043

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

which is obtained from Newton's method

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (k = 0, 1, \ldots)$$

by substituting the derivative $f'(x_k)$ by the ratio $\frac{f(x_k+f(x_k))-f(x_k)}{f(x_k)}$, is an example of a derivative free root-finding method.

The efficiency index of Steffensen's method is $2^{1/2} \approx 1.414$, the same as that of Newton's method. Its computational efficiency can be improved by using derivative free two-point optimal methods of the fourth order, as shown by Kung and Traub [6]. Recently, Ren et al. [10] proposed the following one-parameter family of derivative free two-point methods of the fourth order,

$$y_k = x_k - \frac{f(x_k)}{f[x_k, z_k]}, \quad x_{k+1} = y_k - \frac{f(y_k)}{f[x_k, y_k] + f[y_k, z_k] - f[x_k, z_k] + a(y_k - x_k)(y_k - z_k)}, \tag{1}$$

where $z_k = x_k + f(x_k),\ f[x, y] = \frac{f(x)-f(y)}{x-y}$ denotes a divided difference, and a is a real parameter.

In this paper we consider another derivative free two-point families of methods without and with memory. We prove that the order of convergence is four for methods without memory and asymptotically $2 + \sqrt{5} \approx 4.236$ for methods with memory and even $2 + \sqrt{6} \approx 4.449$ in a special case. From a theoretical as well as practical point of view, this means that the presented methods are competitive or even better than existing optimal two-point methods. The presented derivative free methods are useful for finding roots of a function f when the calculation of derivatives of f is complicated and expensive.

In Section 2 we construct derivative free two-point methods of optimal order four. The first step is Steffensen-like method, while a suitable approximation of the first derivative is used in the second step to reduce a number of function evaluations. In this way the order of convergence *four* is reached by only *three* function evaluations, which means that these methods are optimal in the sense of the Kung–Traub conjecture. Further improvement of convergence rate is considered in Section 3. A family of two-point methods with memory is constructed by varying a free parameter from step to step using only available information. Accelerating the convergence rate from 4 to $2 + \sqrt{5} \approx 4.236$ (asymptotically) without any new calculations, the computational efficiency of these methods is improved comparing to the methods without memory. This theoretical result is confirmed by numerical examples, two of which are presented in Section 4. A comparison with other methods of the same type, performed in Section 4, has shown that the proposed methods are competitive or even better than existing two-point optimal methods.

2. Derivative free optimal two-point methods

Let

$$\varphi_1(x) = \frac{f(x) - f(x - \beta f(x))}{\beta f(x)},$$

where β is an arbitrary real constant. Upon expanding $f(x - \beta f(x))$ into a Taylor series about x, we arrive at

$$\varphi_1(x) = f'(x) - \frac{\beta}{2} f(x) f''(x) + O(f(x)^2).$$

Hence, we obtain an approximation to the derivative $f'(x)$ in the form

$$f'(x) \approx \varphi_1(x) = \frac{f(x) - f(x - \beta f(x))}{\beta f(x)}. \tag{2}$$

Note that the approximation $f'(x) \approx \frac{f(x+\beta f(x))-f(x)}{\beta f(x)}$, used in [1, p. 178], gives the same results as the methods based on (2) and considered in the sequel.

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

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)} \\ x_{k+1} = y_k - \frac{f(y_k)}{f'(y_k)} \end{cases} \quad (k = 0, 1, \ldots). \tag{3}$$

It is well known that the above iterative scheme is inefficient and our primary aim is to improve computational efficiency. We also wish to substitute derivatives $f'(x_k)$ and $f'(y_k)$ by convenient approximations. The latter requirement can be attained at the first step using the approximation (2) to $f'(x_k)$. The derivative $f'(y_k)$ in the second step will be approximated by $\varphi_2(x) = \varphi_1(x)/h(u, v)$, where $h(u, v)$ is a differentiable function that depends on two real variables

$$u = \frac{f(y)}{f(x)}, \quad v = \frac{f(y)}{f(x - \beta f(x))}.$$

Therefore,

$$f'(x_k) \approx \varphi_1(x_k) = \frac{f(x_k) - f(x_k - \beta f(x_k))}{\beta f(x_k)}, \tag{4}$$

$$f'(y_k) \approx \varphi_2(x_k) = \frac{\varphi_1(x_k)}{h(u_k, v_k)}, \quad u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(x_k - \beta f(x_k))}. \tag{5}$$

Now we start from the iterative scheme (3) and using (4) and (5) we construct the following family of two-point methods

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{\varphi_1(x_k)} \\ x_{k+1} = y_k - h(u_k, v_k)\frac{f(y_k)}{\varphi_1(x_k)} \end{cases} \quad (k = 0, 1, \ldots). \tag{6}$$

The function h should be determined in such a way that the order of convergence of the two-point method (6) is four.

Throughout this paper $e_k = x_k - \alpha$ denotes the error of approximation calculated in the kth iteration. In our convergence analysis we will omit sometimes the iteration index k for simplicity. A new approximation x_{k+1} to the root α will be denoted with $\hat{x}$. Let us introduce the errors

$$e = x - \alpha, \quad d = y - \alpha, \quad \hat{e} = \hat{x} - \alpha.$$

We will use Taylor's expansion about the root α to express $f(x)$, $f(x - \beta f(x))$ and $f(y)$ as series in e. Then we represent d and $\hat{e}$ by Taylor's polynomials in e.

Assume that x is sufficiently close to the root α of f, then u and v are close enough to 0. Let us represent a real two-valued function h appearing in (6) by Taylor's series about (0,0) in a linearized form,

$$h(u, v) = h(0,0) + h_u(0,0)u + h_v(0,0)v, \tag{7}$$

where h_u and h_v denote partial derivatives of h with respect to u and v. It can be proved that the use of partial derivatives of higher order does not provide faster convergence (greater than four).

The expressions of Taylor's polynomials (in e) of functions involved in (6) are cumbersome and lead to tedious calculations, which needs the use of computer. If a computer is employed anyway, it is reasonable to perform all calculations necessary in finding the convergence rate using a symbolic computation, as done in this paper. Such an approach was used in [11]. Therefore, instead of the presentation of explicit expressions in a convergence analysis we use symbolic computation in the programming package *Mathematica* to find candidates for h. If necessary, intermediate expressions can be always displayed during this computation using the program given below, although their presentation is most frequently of academic interest.

We will find the coefficients $h(0,0)$, $h_u(0,0)$, $h_v(0,0)$ of the development (7) using a simple program in *Mathematica* and an interactive approach explained by the comments COM-1, COM-2 and COM-3. First, let us introduce the following abbreviations used in this program.

`ak` = $f^{(k)}(\alpha)/(k!f'(\alpha))$, `e` = $x - \alpha$, `el`= $\hat{e} = \hat{x} - \alpha$, `b` = β,
`fx` = $f(x)$, `fxi` = $f(x - \beta f(x))$, `fi` = $[f(x) - f(x - \beta f(x))]/(\beta f(x))$,
`fy` = $f(y)$, `fla` = $f'(\alpha)$, `h0` = $h(0,0)$, `hu` = $h_u(0,0)$, `hv` = $h_v(0,0)$.

Program (written in *Mathematica*)

```
fx=fla*(e+a2*e^2+a3*e^3+a4*e^4);
fxi=fla*((e-b*fx)+a2*(e-b*fx)^2+a3*(e-b*fx)^3);
fi=Series[(fx-fxi)/(b*fx),{e,0,2}];
d=e-b*fx^2*Series [1/(fx-fxi),{e,0,2}];
fy=fla*(d+a2*d^2+a3*d^3);
u=Series[fy/fx,{e,0,2}]; v=Series[fy/fxi,{e,0,2}];
el=d-(h0+hu*u+hv*v)*fy/fi//Simplify; C2=Coefficient[el,e^2]
COM-1: Out[C2] = a2(-1 + bf1a)(-1 + h0)
h0=1; C3=Coefficient[el,e^3]//Simplify
COM-2: Out[C3] = a2^2(2 + b^2f1a^2(1 - hu) - hu - hv + bf1a(-3 + 2hu + hv))
hu=1; hv=1; C4=Coefficient[el,e^4]//Simplify
COM-3: Out[C4] = -a2(-1 + bf1a)(a3(-1 + bf1a) + a2^2(5 - 5bf1a + b^2f1a^2))
```

Comment COM-1: From the expression of the error `el`= $\hat{e} = \hat{x} - \alpha$ we observe that $\hat{e}$ is of the form

$$\hat{e} = \hat{x} - \alpha = C_2e^2 + C_3e^3 + C_4e^4 + O(e^5). \tag{8}$$

The iterative two-point methods (6) will have the order of convergence equal to four if we determine the coefficients of the development appearing in (7) so that C_2 and C_3 (in (8)) vanish. We find these coefficients equalling shaded expressions in boxed formulas to 0. First, from `Out[C2]` we take $h_0 = h(0,0) = 1$ and then calculate C_3.

Comment COM-2: From `Out[C3]` we see that the term $\beta^2 f'(\alpha)$ can be eliminated only if we choose `hu` = $h_u = h_u(0,0) = 1$. With this value the coefficient C_3 will vanish taking `hv` = $h_v = h_v(0,0) = 1$.

Comment COM-3: Substituting the entries $h(0,0) = h_u(0,0) = h_v(0,0) = 1$ in the expression of `e1` (=ê), we obtain

$$\hat{e} = \hat{x} - \alpha = \Big[a_2(1-\beta f'(\alpha))\Big(a_2^2\Big(5 - 5\beta f'(\alpha) + \beta^2 f'(\alpha)^2\Big) - a_3(1-\beta f'(\alpha))\Big)\Big]e^4 + O(e^5),$$

or, using the iteration index,

$$e_{k+1} = x_{k+1} - \alpha = \Big[a_2(1-\beta f'(\alpha))\Big(a_2^2\Big(5 - 5\beta f'(\alpha) + \beta^2 f'(\alpha)^2\Big) - a_3(1-\beta f'(\alpha))\Big)\Big]e_k^4 + O(e_k^5). \tag{9}$$

Therefore, to provide the fourth order of convergence of the two-point methods (6), it is necessary to choose a two-valued function h so that its truncated developments (7) satisfies

$$h(0,0) = h_u(0,0) = h_v(0,0) = 1. \tag{10}$$

According to the above analysis we can formulate the following convergence theorem.

Theorem 1. *Let $h(u,v)$ be a differentiable two-valued function that satisfies the conditions $h(0,0) = h_u(0,0) = h_v(0,0) = 1$. If an initial approximation x_0 is sufficiently close to the root α of a function f, then the convergence order of the family of two-point methods (6) is equal to four.*

Let us stop for a while to study the choice of the function h in (6). Considering (10), we see that the simplest form of the function h is obviously

$$h(u,v) = 1 + u + v. \tag{11}$$

Let us note that any function of the form $h(u,v) = 1 + u + v + g(u,v)$, where g is a differentiable function such that $g(0,0) = g_u(0,0) = g_v(0,0) = 0$, satisfies the condition (10). For example, we can take $g(u,v) = \gamma uv$ (γ is a parameter), $g(u,v) = v(p_1 u + \cdots + p_m u^m)$ or $g(u,v) = u(q_1 v + \cdots + q_s v^s)$, and so on. However, from a practical point of view, we should take forms of as low as possible computational cost.

Another example is the function

$$h(u,v) = \frac{1+u}{1-v}. \tag{12}$$

It is easy to check that it satisfies the conditions (10). A more general (but more complicated) two-parameter function $h(u,v) = (1 + u + au^2)/(1 - v + bv^2)$ also satisfies this condition. The more convenient choices are

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

Finally, substituting

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

in (6) gives the Kung–Traub method (28) considered in Section 4.

Remark 1. The family of two-point methods (6) requires three function evaluations and has the order of convergence four. Therefore, this family is optimal in the sense of the Kung–Traub conjecture and possesses the computational efficiency $E(6) = 4^{1/3} \approx 1.587$.

Remark 2. According to (9), the asymptotic error constant of the family (6) is

$$C_4(\alpha) = \lim_{k\to\infty} \frac{x_{k+1}-\alpha}{(x_k-\alpha)^4} = a_2(1-\beta f'(\alpha))\Big(a_2^2(5 - 5\beta f'(\alpha) + \beta^2 f'(\alpha)^2) - a_3(1-\beta f'(\alpha))\Big).$$

However, this expression is valid if the Taylor series of h given by (7) is truncated to the displayed members. Obviously, it holds for the function h defined by (11). For a particular function h, the asymptotic error constant $C_4(\alpha)$ is given by a specific expression. For example, choosing $h(u,v) = (1+u)/(1-v)$ (formula (12)) we find the asymptotic error constant

$$C_4(\alpha) = a_2(1-\beta f'(\alpha))^2(a_2^2(3-\beta f'(\alpha)) - a_3). \tag{13}$$

Remark 3. Since $d = a_2(1-\beta f'(\alpha))e^2 + O(e^3)$, it is easy to show that the asymptotic error constant $C_4(\alpha)$ always contains the factor $1 - \beta f'(\alpha)$, that is,

$$C_4(\alpha) = (1-\beta f'(\alpha))\Psi(a_2, a_3, f'(\alpha)), \tag{14}$$

where Ψ is a non-zero constant depending on a_2, a_3, $f'(\alpha)$. This fact is a very important in developing the corresponding two-point methods with memory, see Section 3.

Remark 4. We speak about the family (6), since the choice of various functions h satisfying the conditions (10) and a parameter β gives a variety of two-point methods.

Remark 5. The statement `fi = Series[(fx-fxi)/(b*fx),{e,0,2}]` in the above program gives

$$\frac{f(x_k) - f(x_k - \beta f(x_k))}{\beta f(x_k)} = f'(\alpha)\left[1 + a_2(2 - \beta f'(\alpha))e_k + O(e_k^2)\right], \tag{15}$$

which will be utilized in Section 3.

3. Two-point methods with memory

Following Traub's classification [1], the two-point methods (6) belong to the class of methods without memory. Recall that a multipoint method with memory reuses old information, for example, some entries from the previous iteration. In this section we will improve the convergence rate of the family (6) using an old idea given in [1, p. 186] consisting of varying the parameter β as the iteration proceeds.

Considering the error relation (9) and having in mind the form of the asymptotic error constant (14), we observe that the order of convergence of the two-point family (6) could be increased from 4 to 5 taking $\beta = 1/f'(\alpha)$. However, in practice we have no information on the exact value $f'(\alpha)$ and this idea cannot be fully realized. Instead of that, we can vary β from step to step using information available from the previous step and exceed the order 4 without using any new function evaluations. By defining β recursively as the iteration proceeds, we obtain two-point methods *with memory* corresponding to (6). Such methods may also be called *self-accelerating* methods.

From the previous discussion, it is clear that we are forced to estimate β by an approximation of $1/f'(\alpha)$ using available data. We present two methods.

Method (I): Following (2) we estimate

$$\beta_k = \frac{\beta_{k-1} f(x_{k-1})}{f(x_{k-1}) - f(x_{k-1} - \beta_{k-1} f(x_{k-1}))} \quad (k = 1, 2, \ldots) \tag{16}$$

starting from β_0. The initial value β_0 may be chosen in various ways; we found by practical experiments that the choice of small entry (in magnitude), for example, $\beta_0 = 10^{-2}$ or less, gives satisfactory results in practice. Since

$$\frac{f(x) - f(x - \beta f(x))}{\beta f(x)} \to f'(x) \quad \text{when } \beta \to 0,$$

it may be expected that β_k, defined by (16), will quickly approach $f'(\alpha)$ in the course of iterative process, especially in latter iterations, if the initial approximation x_0 is reasonable close to the root α.

Using recursively calculated parameter β_k, we define the following two-step methods with memory

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{\varphi_1(x_k, \beta_k)} \\ x_{k+1} = y_k - h(u_k, v_k)\frac{f(y_k)}{\varphi_1(x_k, \beta_k)} \end{cases} \quad (k = 0, 1, \ldots), \tag{17}$$

where

$$\varphi_1(x_k, \beta_k) := \frac{f(x_k) - f(x_k - \beta_k f(x_k))}{\beta_k f(x_k)}$$

and β_k is calculated by (16).

To estimate the order of convergence of the two-point methods (17) with memory, we will use the relations (14)–(16). First, C_4 given by (14) (this quantity is not a constant now) is rewritten in the form

$$\widetilde{C}_4 = (1 - \beta_k f'(\alpha))\widetilde{\Psi}_k(a_2, a_3, f'(\alpha), \beta_k). \tag{18}$$

It can be observed from the convergence analysis that the parameter β_k always appears multiplied by $f'(\alpha)$. Putting $q_k = \beta_k f'(\alpha)$, we may rewrite (18) as

$$\widetilde{C}_4 = (1 - q_k)\widetilde{\Psi}_k(a_2, a_3, q_k). \tag{19}$$

From (15) and (16) we find

$$\beta_k = \frac{1}{f'(\alpha)\left(1 + a_2(2 - q_{k-1})e_{k-1} + O(e_{k-1}^2)\right)} = \frac{1}{f'(\alpha)}\left(1 - a_2(2 - q_{k-1})e_{k-1} + O(e_{k-1}^2)\right).$$

Hence

$$1 - \beta_k f'(\alpha) = 1 - q_k = a_2(2 - q_{k-1})e_{k-1} + O(e_{k-1}^2). \tag{20}$$

In a limiting process when $x_k \to \alpha$, the quantity $\widetilde{\Psi}_k(a_2, a_3, q_k)$ in (19) does not vanish and tends to a constant. Using this fact and (20) we can write the following error relation for the two-point methods (17),

$$e_{k+1} = x_{k+1} - \alpha = \widetilde{C}_4 e_k^4 + O(e_k^5) = (1 - q_k)\widetilde{\Psi}_k(a_2, a_3, q_k)e_k^4 + O(e_k^5), \tag{21}$$

giving

$$e_{k+1} = a_2(2 - q_{k-1})\widetilde{\Psi}_k(a_2, a_3, q_k)e_k^4 e_{k-1} + O(e_k^5).$$

Hence, when $x_k \to \alpha$,

$$e_{k+1} \sim e_k^4 e_{k-1}, \tag{22}$$

where the denotation $\gamma \sim \delta$ means that $\gamma = O(\delta)$.

Method (II): We may approximate the parameter β using the secant approach, that is,

$$\beta_k^* = \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \approx \frac{1}{f'(\alpha)}. \tag{23}$$

Then we can state the following family of two-point methods with memory,

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{\varphi_1(x_k, \beta_k^*)} \\ x_{k+1} = y_k - h(u_k, v_k)\frac{f(y_k)}{\varphi_1(x_k, \beta_k^*)} \end{cases} \quad (k = 0, 1, \ldots), \tag{24}$$

where

$$\varphi_1(x_k, \beta_k^*) := \frac{f(x_k) - f(x_k - \beta_k^* f(x_k))}{\beta_k^* f(x_k)}$$

and β_k is calculated by (23). In a similar way as in Section 2 for the family (6), we can derive the error relation for the family of methods (24),

$$e_{k+1} = x_{k+1} - \alpha = C_4^* e_k^4 + O(e_k^5) = (1 - \beta_k^* f'(\alpha))\Psi_k^*(a_2, a_3, q_k^*)e_k^4 + O(e_k^5), \;\; q_k^* = \beta_k^* f'(\alpha). \tag{25}$$

Using a Taylor series about α, we arrive at

$$f(x_k) = f'(\alpha)e_k + \frac{1}{2}f''(\alpha)e_k^2 + O(e_k^3)$$

and hence

$$\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = f'(\alpha) + \frac{1}{2}(e_k + e_{k-1})f''(\alpha) + O(e_{k-1}^2).$$

Then

$$1 - \beta_k^* f'(\alpha) = 1 - \frac{f'(\alpha)}{f'(\alpha) + \frac{1}{2}(e_k + e_{k-1})f''(\alpha) + O(e_{k-1}^2)} = 1 - \left(1 - \frac{(e_k + e_{k-1})f''(\alpha)}{2f'(\alpha)} + O(e_{k-1}^2)\right) = \frac{(e_k + e_{k-1})f''(\alpha)}{2f'(\alpha)} + O(e_{k-1}^2).$$

Substituting the expression of $1 - \beta_k^* f'(\alpha)$ in the error relation (25) leads again to the relation (22).

To estimate the convergence rate of the two-step methods (17) and (24), we will use the concept of the R-order of convergence introduced by Ortega and Rheiboldt [12] and the following assertion (see [13, p. 287]).

Theorem 2. *If the errors of approximations $e_j = x_j - \alpha$ obtained in an iterative root-finding process IP satisfy*

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

then the R-order of convergence of IP, denoted with $O_R(IP, \alpha)$, satisfies the inequality

$$O_R(IP, \alpha) \geqslant s^*,$$

where s^* is the unique positive solution of the equation

$$s^{n+1} - \sum_{i=0}^{n} m_i s^{n-i} = 0. \tag{26}$$

Now we can state the convergence theorem of the two-point methods (17) and (24) with memory.

Theorem 3. *If an initial approximation x_0 is sufficiently close to the root α of a function f, then the R-order of convergence of the family of two-point methods (17) and (24) is at least $2+\sqrt{5}$.*

Proof. Using Theorem 2 for $n = 1$, $m_0 = 4$, $m_1 = 1$, according to (22) and (26) we form the quadratic equation

$$s^2 - 4s - 1 = 0.$$

The lower bound of the R-order of the methods (17) and (24) is determined by the unique positive root $s^* = 2+\sqrt{5} \approx 4.236$ of this equation. □

The choice of $h(u,v) = (1+u)/(1-v)$ in (17) and (24) provides more accurate approximations compared to the function $h(u,v) = 1+u+v$ (see, for example, Tables 1 and 2). In fact, the following assertion is valid.

Theorem 4. *If an initial approximation x_0 is sufficiently close to the root α of a function f and $h(u,v) = (1+u)/(1-v)$, then the R-order of convergence of the family of two-point methods (17) and (24) is at least $2+\sqrt{6} \approx 4.449$.*

Proof. From (14) we see that the asymptotic error constant $C_4(\alpha)$ for $h(u,v) = (1+u)/(1-v)$ contains the factor $(1-\beta f'(\alpha))^2$. Since $1-\beta_k f'(\alpha) = O(e_{k-1})$ (see (20)), from a relation similar to (21) we obtain

$$e_{k+1} \sim e_k^4 e_{k-1}^2.$$

Hence, according to Theorem 2 and (26), it follows that the R-order of the families (17) and (24), with $h(u,v) = (1+u)/(1-v)$, is at least $s^* = 2+\sqrt{6} \approx 4.449$, where s^* is the positive root of the equation $s^2 - 4s - 2 = 0$. □

Remark 6. The R-order at least $2+\sqrt{6}$ can also be achieved dealing with the function $h(u,v) = (1+u)(1+v+v^2)$ and some more complicated functions.

Remark 7. The fact that the order $2+\sqrt{5}$ of the families (17) and (24) is greater than 4 (Theorem 3) does not mean that the Kung–Traub conjecture is refuted. Recall that this conjecture holds only for multipoint methods without memory, while the iterative formulas (17) and (24) define the methods *with memory*.

Table 1
$f(x) = e^x \sin 5x - 2$, $x_0 = 1.5$, $\alpha = 1.363973180263712\ldots$

Two-point methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$
(6) $h = 1+u+v$, $\beta = 0.01$	1.70(−2)	6.41(−8)	2.27(−29)	3.57(−115)
(17) $h = 1+u+v$, $\beta_0 = 0.01$	1.70(−2)	2.91(−8)	1.08(−34)	8.35(−146)
(24) $h = 1+u+v$, $\beta_0 = 0.01$	1.70(−2)	2.35(−9)	1.03(−38)	5.63(−163)
(6) $h = \frac{1+u}{1-v}$, $\beta = 0.01$	8.36(−3)	4.85(−9)	6.98(−34)	2.98(−133)
(17) $h = \frac{1+u}{1-v}$, $\beta_0 = 0.01$	8.36(−3)	1.83(−9)	4.51(−41)	3.79(−180)
(24) $h = \frac{1+u}{1-v}$, $\beta_0 = 0.01$	8.36(−3)	1.93(−10)	2.12(−44)	2.04(−195)
Ostrowski IM (27)	6.40(−3)	2.53(−9)	7.39(−35)	5.41(−137)
Kung–Traub IM (28)	1.68(−2)	1.11(−7)	3.96(−28)	6.45(−110)
Jarratt IM (29)	6.39(−3)	2.82(−9)	1.24(−34)	4.67(−136)
Maheshwari IM (30)	2.57(−2)	2.95(−7)	1.51(−26)	1.02(−103)
Ren–Wu–Bi IM (1), $a = 0, x_0 = 1.4$[a]	1.85(−2)	3.31(−4)	9.35(−12)	5.42(−42)

[a] The Ren–Wu–Bi method does not converge in 100 iterative steps for $x_0 = 1.5$.

Table 2
$f(x) = (x-2)(x^{10}+x+1)e^{-x-1}$, $x_0 = 2.1$, $\alpha = 2$.

Two-point methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$
(6) $h = 1+u+v$, $\beta = 0.01$	1.01(−3)	7.84(−11)	2.93(−39)	5.68(−153)
(17) $h = 1+u+v$, $\beta_0 = 0.01$	1.01(−3)	5.01(−11)	2.23(−42)	3.13(−175)
(24) $h = 1+u+v$, $\beta_0 = 0.01$	1.01(−3)	4.00(−11)	6.60(−43)	1.92(−177)
(6) $h = \frac{1+u}{1-v}$, $\beta = 0.01$	3.29(−4)	3.66(−13)	5.59(−49)	3.04(−192)
(17) $h = \frac{1+u}{1-v}$, $\beta_0 = 0.01$	3.29(−4)	2.00(−13)	5.20(−55)	4.69(−240)
(24) $h = \frac{1+u}{1-v}$, $\beta_0 = 0.01$	3.29(−4)	1.45(−13)	7.63(−56)	1.13(−243)
Ostrowski IM (27)	1.72(−3)	3.13(−10)	3.49(−37)	5.43(−145)
Kung–Traub IM (28)	7.56(−3)	6.80(−7)	4.88(−23)	1.29(−87)
Jarratt IM (29)	1.75(−3)	3.42(−10)	5.11(−37)	2.54(−144)
Maheshwari IM (30)	5.27(−3)	1.59(−7)	1.45(−25)	9.97(−98)
Ren–Wu–Bi IM (1), $a = 0$	2.66(−2)	2.09(−3)	1.26(−6)	2.53(−19)

Remark 8. The values of β_k approximated by (16) and (23) have interesting geometric interpretation. In both cases $-\beta_k$ approaches $-1/f'(\alpha)$. Recall that $-1/f'(\alpha)$ determines the slope of the normal to the tangent line to the curve $y = f(x)$ at the point $[\alpha, 0]$. Therefore, $-\beta_k$ tends to the tangent of the angle of this normal to the x axis.

According to Theorems 3 and 4 we conclude that iterative adjustment of a parameter β leads to the increase of convergence speed of the two-point methods considered. Since this acceleration of convergence is achieved without any new function evaluations, we conclude that the computational efficiency of the proposed methods (17) and (24) with memory is increased compared to the basic methods (6) without memory. Numerical results presented in Section 4 evidently confirm the increase of convergence speed of the family of two-point methods (17) and (24).

4. Numerical results

In this section we demonstrate the convergence behavior of the proposed families (6), (17) and (24) of two-point methods with and without memory. For comparison, in our numerical experiments we also tested the derivative free methods (1) and several two-point iterative methods (IM) reviewed below. For simplicity, we omit the iteration index and introduce the Newton correction $w(x)$ (after Ne**w**ton) and a divided difference by

$$w(x) = \frac{f(x)}{f'(x)}, \quad f[x,y] = \frac{f(x) - f(y)}{x - y}.$$

Ostrowski's method [2]

$$y = x - w(x), \quad \hat{x} = y - \frac{f(y)}{f'(x)} \cdot \frac{f(x)}{f(x) - 2f(y)}. \tag{27}$$

Kung–Traub's method [6]

$$y = x - \frac{\beta f(x)^2}{f(x + \beta f(x)) - f(x)}, \quad \hat{x} = y - \frac{f(y)f(x + \beta f(x))}{[f(x + \beta f(x)) - f(y)]f[x,y]}. \tag{28}$$

Note that the family of two-point methods (28) is a special case ($n = 2$) of derivative free multipoint family of methods of arbitrary order of convergence 2^n requiring $n + 1$ function evaluations, see [6]. The family (28) can be obtained as a special case of the family (6) taking $h(u,v) = [(1-u)(1-v)]^{-1}$. Another optimal multipoint methods of arbitrary order of convergence were considered in [14].

Jarratt's method [3]

$$\hat{x} = x - \frac{1}{2}w(x) + \frac{f(x)}{f'(x) - 3f'\left(x - \frac{2}{3}w(x)\right)}. \tag{29}$$

Maheshwari's method [15]

$$\hat{x} = x - w(x)\left\{\frac{[f(x - w(x))]^2}{f(x)^2} - \frac{f(x)}{f(x - w(x)) - f(x)}\right\}. \tag{30}$$

The listed methods, including the Ren–Wu–Bi family (1) and the proposed methods (6), (17) and (24), have the efficiency index at least $4^{1/3} \approx 1.587$. We did not insert numerical results obtained by iterative methods of lower computational efficiency since these methods are not competitive and their lower rank is predictable. Another optimal two-point methods of the fourth order with the efficiency index $4^{1/3}$ were considered in [16–22], but the cited papers do not exhaust all sources.

For demonstration, among many numerical examples we selected two examples implemented in the programming package *Mathematica* by the use of multi-precision arithmetic.

Example 1. We tested the proposed methods (6), (17) and (24) with $\beta = \beta_0 = 0.01$ and choosing two forms (11) and (12) of the function h, the family (1) for $a = 0$ and the two-point methods (27)–(30) listed above. These methods were applied to the function

$$f(x) = e^x \sin 5x - 2$$

to approximate its root $\alpha = 1.363973180263712\ldots$. The absolute values of the errors of the approximations x_k in the first four iterations are displayed in Table 1, where $A(-h)$ means $A \times 10^{-h}$. All methods used the common initial approximation $x_0 = 1.5$, except the Ren–Wu–Bi method (1) (see the remark *) below Table 1) which shows a divergent behavior in the first 100 iterations. For this reason, this method was applied for a closer approximation $x_0 = 1.4$.

Example 2. The two-point methods employed in Example 1 were also applied to the function

$$f(x) = (x-2)(x^{10} + x + 1)e^{-x-1}$$

to approximate its root $\alpha = 2$. In this test we used the common initial approximation $x_0 = 2.1$. The obtained results are presented in Table 2.

According to the results presented in Tables 1 and 2 and a number of numerical examples, we can conclude that the derivative free two-point methods (6) are competitive with existing optimal two-point methods of the fourth order. Even better, the families of methods (17) and (24) with memory, based on recursively calculated parameter β_k, are faster than the existing methods. Since all of the tested methods have the same computational cost, it is evident that the methods (17) and (24) are most efficient.

Comparing the fourth order methods (1) and (6) we notice that the main advantage of the proposed family (6) is the possibility of acceleration of convergence by varying a parameter β. The lack of a corrector (as a parameter β in (6)) in the iterative formula (1) leads to the divergence of the method (1) in Example 1 (for various values of a parameter a in a wide range) and relatively modest result in Example 2.

Our concluding remark is concerned with an important problem appearing in practical application of multipoint methods. As emphasized in [14], a fast convergence, one of the advantages of multipoint methods, can be attained only if initial approximations are sufficiently close to the sought roots; otherwise, it is not possible to realize the expected convergence speed in practice. For this reason, applying multipoint root-finding methods, a special attention should be paid to finding good initial approximations. We note that an efficient procedure for finding sufficiently good initial approximations was recently proposed by Yun [23]. For illustration, simple statements in the programming package *Mathematica*, applied to the function from Example 2 and the interval [1,5],

```
f[x_]=(x-2)(x^(10)+x + 1)*Exp[-x-1]; a = 1; b = 5; m = 5;
x0 = 0.5*(a + b+Sign[f[a]]*NIntegrate[Tanh[m*f[x]],{x, a, b}])
```

gives considerably good initial approximation $x_0 = 1.99857$ to the root $\alpha = 2$.

Acknowledgement

This work was supported by the Serbian Ministry of Science under grant 144024.

References

[1] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
[2] A.M. Ostrowski, Solution of Equations and Systems of Equations, Academic Press, New York, 1960.
[3] P. Jarratt, Some fourth order multipoint methods for solving equations, Math. Comput. 20 (1966) 434–437.
[4] P. Jarratt, Some efficient fourth-order multipoint methods for solving equations, BIT 9 (1969) 119–124.
[5] R. King, A family of fourth order methods for nonlinear equations, SIAM J. Numer. Anal. 10 (1973) 876–879.
[6] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, J. ACM 21 (1974) 643–651.
[7] B. Neta, On a family of multipoint methods for nonlinear equations, Int. J. Comput. Math. 9 (1981) 353–361.
[8] 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.
[9] J.F. Steffensen, Remark on iteration, Skand. Aktuar Tidskr. 16 (1933) 64–72.
[10] 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.
[11] R. Thukral, M.S. Petković, A family of three-point methods of optimal order for solving nonlinear equations, J. Comput. Appl. Math. 233 (2010) 2278–2284.
[12] J.M. Ortega, W.C. Rheiboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
[13] G. Alefeld, J. Herzberger, Introduction to Interval Computation, Academic Press, New York, 1983.
[14] M.S. Petković, On a general class of multipoint root-finding methods of high computational efficiency, SIAM J. Numer. Anal. 47 (2010) 4402–4414.
[15] A.K. Maheshwari, A fourth-order iterative method for solving nonlinear equations, Appl. Math. Comput. 211 (2009) 383–391.
[16] D. Basu, From third to fourth order variant of Newton's method for simple roots, Appl. Math. Comput. 202 (2008) 886–892.
[17] C. Chun, A family of composite fourth-order iterative methods for solving nonlinear equations, Appl. Math. Comput. 187 (2007) 951–956.
[18] C. Chun, Some variants of King's fourth-order family of methods for nonlinear equations, Appl. Math. Comput. 190 (2007) 57–62.
[19] C. Chun, Some fourth-order iterative methods for solving nonlinear equations, Appl. Math. Comput. 195 (2008) 454–459.
[20] C. Chun, Y. Ham, A one-parameter fourth-order family of iterative methods for nonlinear equations, Appl. Math. Comput. 189 (2007) 610–614.
[21] C. Chun, B. Neta, Certain improvements of Newton's method with fourth-order convergence, Appl. Math. Comput. 215 (2009) 821–828.
[22] M.S. Petković, L.D. Petković, A one parameter square root family of two-step root-finders, Appl. Math. Comput. 188 (2007) 339–344.
[23] B.I. Yun, A non-iterative method for solving non-linear equations, Appl. Math. Comput. 198 (2008) 691–699.