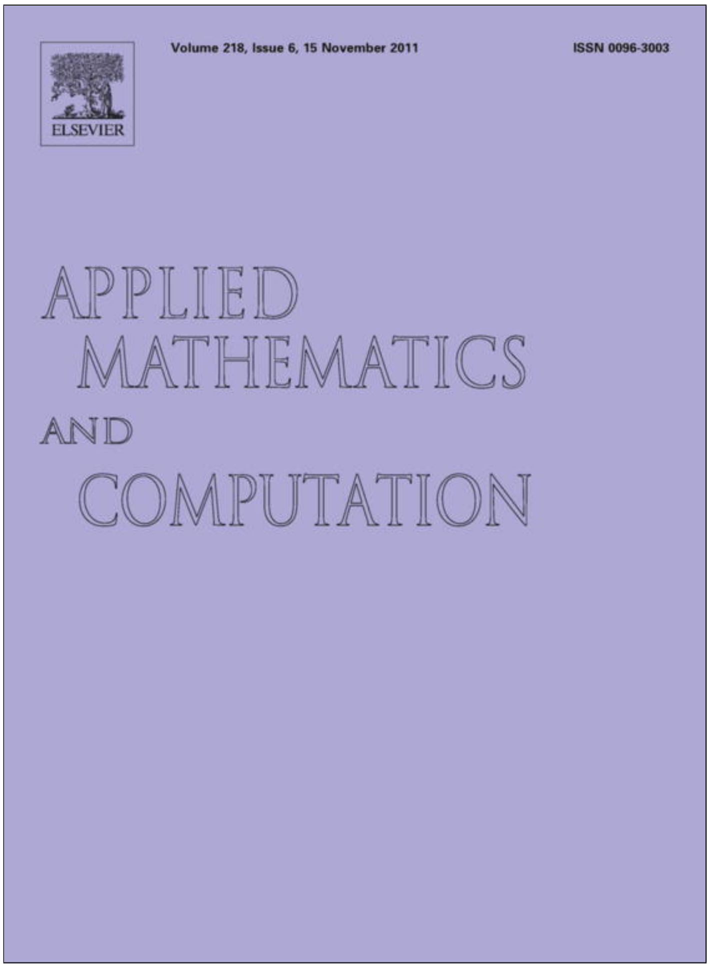
Volume 218, Issue 6, 15 November 2011
ISSN 0096-3003
ELSEVIER
APPLIED MATHEMATICS AND COMPUTATION

Applied Mathematics and Computation 218 (2011) 2533–2541

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Interpolatory multipoint methods with memory for solving nonlinear equations

Miodrag S. Petković [a,*], Jovana Džunić [a], Beny Neta [b]

[a] Faculty of Electronic Engineering, Department of Mathematics, University of Niš, 18000 Niš, Serbia
[b] Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA 93943, USA

ARTICLE INFO

Keywords:
Nonlinear equations
Multipoint methods
Inverse interpolation
Convergence
Computational efficiency

ABSTRACT

A general way to construct multipoint methods for solving nonlinear equations by using inverse interpolation is presented. The proposed methods belong to the class of multipoint methods with memory. In particular, a new two-point method with memory with the order $(5+\sqrt{17})/2 \approx 4.562$ is derived. Computational efficiency of the presented methods is analyzed and their comparison with existing methods with and without memory is performed on numerical examples. It is shown that a special choice of initial approximations provides a considerably great accuracy of root approximations obtained by the proposed interpolatory iterative methods.

1. Introduction

The main goal and motivation in constructing iterative methods for solving nonlinear equations is to attain as fast as possible order of convergence with minimal computational costs. The most efficient existing root-solvers are based on multipoint iterations, first studied in Traub's book [29] and some papers and books published in the second half of the 20th century (see, e.g., [7–11,14–17,20]). Multipoint iterative methods have again become an interesting and challenging task at the beginning of the 21st century since they overcome theoretical limits of one-point methods concerning the convergence order and computational efficiency. The highest possible computational efficiency of these methods is closely connected to the hypothesis of Kung and Traub [11] from 1974. They have conjectured that the order of convergence of any multipoint method without memory, requiring $n+1$ 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 (but not exhausting) list of optimal methods may be found, for example, in [21] and [24].

The convergence of multipoint methods can be accelerated without additional computations using information from the points at which old data are reused. Let y_j represent the $s+1$ quantities $x_j, \omega_1(x_j), \ldots, \omega_s(x_j)$ $(s \geqslant 1)$ and define an iterative process by

$$x_{k+1} = \varphi(y_k; y_{k-1}, \ldots, y_{k-m}).$$

Following Traub's terminology [29], φ is called a *multipoint iterative function with memory*. Two simple examples of this type of iterative functions were presented in Traub's book [29, pp. 185–187]. In the recent paper [22] the two-point methods of the fourth order were modified to the methods with memory which possess the increased order $2+\sqrt{5} \approx 4.236$ and $2+\sqrt{6} \approx 4.449$.

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

0096-3003/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2011.07.068

In this paper we present multipoint methods for solving nonlinear equations, constructed by inverse interpolation. These methods will be referred to as interpolatory iterative methods. The basic idea comes from one of the authors who derived very fast three-point method of the R-order $10.81\dot{5}$ at the eighties of the last century, see [16]. In Section 2 we construct a two-point method with memory of the order of convergence $(5+\sqrt{17})/2 \approx 4.561$. Multipoint methods with memory of higher order, also based on inverse interpolation, are presented in Section 3. The comparison of computational efficiency of multipoint methods with and without memory is the subject of Section 4. Numerical examples are given in Section 5 to illustrate convergence behavior of multipoint methods. It can be seen from these examples that a special choice of initial approximations provides considerably great accuracy of approximations to the roots, obtained by the proposed methods.

2. Two-point interpolatory iterative methods

Let x_0 and y_{-1} be two starting initial approximations of the sought zero α of a given real function f. We will now construct a two-point method calculating first y_k on the basis of the values of f at x_k, y_{k-1} and the value of f' at x_k. Then a new approximation x_{k+1} is calculated using the values of f at x_k, y_k and the value of f' at x_k.

We use inverse interpolation to compute y_k. Let

$$R(f(x)) = a + b(f(x) - f(x_k)) + c(f(x) - f(x_k))^2 \tag{1}$$

be a polynomial of degree two satisfying

$$x_k = R(f(x_k)), \tag{2}$$

$$\frac{1}{f'(x_k)} = R'(f(x_k)), \tag{3}$$

$$y_{k-1} = R(f(y_{k-1})). \tag{4}$$

From (2) and (3) we obtain

$$a = x_k, \quad b = \frac{1}{f'(x_k)}. \tag{5}$$

Let us introduce

$$\Phi(t) = \frac{1}{f(t) - f(x_k)}\left[\frac{t - x_k}{f(t) - f(x_k)} - \frac{1}{f'(x_k)}\right] \tag{6}$$

and let

$$N(x) = x - \frac{f(x)}{f'(x)}$$

denote Newton's iterative function. In view of (1) and (4) we obtain $c = \Phi(y_{k-1})$ so that, together with (5), it follows from (1)

$$y_k = R(0) = x_k - \frac{f(x_k)}{f'(x_k)} + f(x_k)^2\Phi(y_{k-1}) = N(x_k) + f(x_k)^2\Phi(y_{k-1}). \tag{7}$$

In the next step, to find x_{k+1} we carry out the same calculation but using y_k instead of y_{k-1}. The constant c appearing in (1) is now given by $c = \Phi(y_k)$ and we find from (1)

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} + f(x_k)^2\Phi(y_k) = N(x_k) + f(x_k)^2\Phi(y_k), \tag{8}$$

where y_k is calculated by (7).

Remark 1. To start the iterative process we need two initial approximations x_0 and y_{-1}. However, let us observe that y_{-1} may take the value $N(x_0)$ at the first iteration without any additional computational cost. Indeed, $N(x_0)$ appears anyway in (7) and (8) for $k = 0$. To avoid unnecessary evaluation at the last step of iterative process, $N(x_k)$ is calculated only if the stopping criterion is not fulfilled. In that case we calculate $N(x_k)$,increase k to $k+1$ and apply the next iteration. Practical examples show that such a choice of y_{-1} in (9) and (14) (see Section 3) considerably increases the accuracy of obtained approximations, see Tables 4–11.

The relations (7) and (8) define the two-point method with memory,

$$\text{given } x_0, \quad y_{-1} = N(x_0), \begin{cases} y_k = N(x_k) + f(x_k)^2\Phi(y_{k-1}), & (k = 0, 1, \ldots) \\ x_{k+1} = N(x_k) + f(x_k)^2\Phi(y_k), \end{cases} \tag{9}$$

where Φ is given by (6). The order of convergence of the method (9) is given in the following theorem.

Theorem 1. *The two-point method (9) has the R-order of convergence at least* $\rho(M^{(2)}) = (5+\sqrt{17})/2 \approx 4.56115$, *where* $\rho(M^{(2)})$ *is the spectral radius of the matrix*

$$M^{(2)} = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix}.$$

Proof. We shall use Herzberger's matrix method [6] on the order of a single step s-point method $x_k = G(x_{k-1}, x_{k-2}, \ldots, x_{k-s})$. A matrix $M^{(s)} = (m_{ij})$, associated to this method, has the elements

$$m_{1,j} = \text{amount of information required at point } x_{k-j}, \ (j = 1, 2, \ldots, s),$$
$$m_{i,i-1} = 1 \ (i = 2, 3, \ldots, s),$$
$$m_{i,j} = 0 \text{ otherwise.}$$

The order of an s-step method $G = G_1 \circ G_2 \circ \cdots \circ G_s$ is the spectral radius of the product of matrices $M^{(s)} = M_1 \cdot M_2 \cdots M_s$. According to the relations (7) and (8) we form the respective matrices,

$$M_1 = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}.$$

Hence

$$M^{(2)} = M_1 \cdot M_2 = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix}.$$

The characteristic polynomial of the matrix M is

$$P_2(\lambda) = \lambda^2 - 5\lambda + 2.$$

Its roots are $4.561\dot{2}$, $0.4384\dot{5}$; therefore the spectral radius of the matrix $M^{(2)}$ is $\rho(M^{(2)}) = 4.561\dot{6}$, which gives the lower bound of the R-order of the method (9). □

Remark 2. Let $y_k = x_k - f(x_k)/f'(x_k)$ be calculated in advance and let us express the condition (4) in the form $y_k = R(f(y_k))$. Finding the coefficients a, b, c from the inverse interpolation (1) and the conditions (2)–(4) we arrive at the two-point method

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)} \quad (k = 0, 1, \ldots) \\ x_{k+1} = y_k - \frac{f(x_k)^2 f(y_k)}{f'(x_k)(f(y_k) - f(x_k))^2}. \end{cases}$$

This method of optimal order four is a special case of the Kung-Traub family of arbitrary order of convergence presented in [11].

3. Multipoint interpolatory iterative methods

Now we will present in short the three-point method with memory derived by Neta [16] in 1983. This method was presented in [16] without numerical examples and comparison with existing methods and our intention is to complete numerical experiments. Neta's method requires three initial approximations x_0, y_{-1}, z_{-1} and it was constructed using inverse interpolatory polynomial

$$R(f(x)) = a + b(f(x) - f(x_k)) + c(f(x) - f(x_k))^2 + d(f(x) - f(x_k))^3$$

of degree three satisfying

$$x_k = R(f(x_k)), \tag{10}$$
$$\frac{1}{f'(x_k)} = R'(f(x_k)), \tag{11}$$
$$y_{k-1} = R(f(y_{k-1})), \tag{12}$$
$$z_{k-1} = R(f(z_{k-1})). \tag{13}$$

Let us define

$$\Psi(t) = \frac{t - x_k}{(f(t) - f(x_k))^2} - \frac{1}{(f(t) - f(x_k))f'(x_k)}.$$

Using the conditions (10), (12), (13), Neta derived the following three-point method

$$\begin{cases} y_k = N(x_k) + [f(y_{k-1})\Psi(z_{k-1}) - f(z_{k-1})\Psi(y_{k-1})]\frac{f(x_k)^2}{f(y_{k-1})-f(z_{k-1})}, \\ z_k = N(x_k) + [f(y_k)\Psi(z_{k-1}) - f(z_{k-1})\Psi(y_k)]\frac{f(x_k)^2}{f(y_k)-f(z_{k-1})}, \\ x_{k+1} = N(x_k) + [f(y_k)\Psi(z_k) - f(z_k)\Psi(y_k)]\frac{f(x_k)^2}{f(y_k)-f(z_k)} \end{cases} \tag{14}$$

for $k = 0, 1, \ldots$. It is preferable that y_{-1} takes the value $N(x_0)$ at the first iteration, see Remarks 1 and 3.

Respective matrices corresponding to the steps of the three-point method (14) are

$$M_1 = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad M_3 = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

According to this, the following theorem was proved in [16].

Theorem 2. *The three-point method (14) has the R-order of convergence at least $\rho(M^{(3)}) \approx 10.815$, where $\rho(M^{(3)})$ is the spectral radius of the matrix*

$$M^{(3)} = M_1 \cdot M_2 \cdot M_3 = \begin{bmatrix} 8 & 5 & 6 \\ 3 & 2 & 2 \\ 1 & 1 & 2 \end{bmatrix}.$$

In a similar way we could continue to construct the four-point methods using inverse interpolatory polynomial of degree four

$$R(f(x)) = a_0 + a_1(f(x) - f(x_k)) + a_2(f(x) - f(x_k))^2 + a_3(f(x) - f(x_k))^3 + a_4(f(x) - f(x_k))^4.$$

The corresponding 4×4 matrices M_1, M_2, M_3, M_4 and the resulting matrix $M^{(4)}$ are presented below:

$$M^{(4)} = M_1 \cdot M_2 \cdot M_3 \cdot M_4 = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 14 & 16 & 11 & 16 \\ 5 & 6 & 4 & 6 \\ 2 & 3 & 2 & 2 \\ 1 & 1 & 1 & 2 \end{bmatrix}.$$

The spectral radius $\rho(M^{(4)})$ of the final matrix is $\rho(M^{(4)}) \approx 22.704$ and it determines the R-order of the four-point method with memory, constructed by the inverse interpolatory polynomial of degree four. However, we regard that the convergence speed of the described method is too fast that it exceeds practical requirements and, for this reason, we will not discuss this method here.

Computational efficiency of the methods (9) and (14), constructed by inverse interpolation, and their comparison with the existing methods of order four and eight is discussed in the next section. Results of numerical experiments are given in Tables 4–11 in Section 5.

4. Comparison of computational efficiency

In this paper we consider two-point methods and three-point methods with and without memory from the computational point of view. For comparison purpose, we present Kung-Traub's n-point methods with/without memory arising from Kung-Traub's family whose order of convergence is at least 2^n ($n \geqslant 2$), see [11]. For $n = 2$ the following two-point method is generated,

$$\begin{cases} y_k = x_k - \frac{\beta_k f(x_k)^2}{f(x_k+\beta_k f(x_k)) - f(x_k)}, \\ x_{k+1} = y_k - \frac{f(y_k) f(x_k+\beta_k f(x_k))}{(f(x_k+\beta_k f(x_k)) - f(y_k)) f[x_k, y_k]}, \end{cases} \quad (k = 0, 1, \ldots), \tag{15}$$

where $f[x,y] = [f(x) - f(y)]/(x - y)$ is a divided difference and β_k is either a nonzero constant or self-accelerating variable parameter, see [29, pp. 185–187] and [22] for details.

The following three-point method is obtained as the next special case of Kung-Traub's family taking $n = 3$,

$$\begin{cases} y_k = x_k - \frac{\beta_k f(x_k)^2}{f(x_k+\beta_k f(x_k)) - f(x_k)}, \\ z_k = y_k - \frac{f(y_k) f(x_k+\beta_k f(x_k))}{(f(x_k+\beta_k f(x_k)) - f(y_k)) f[x_k, y_k]}, \quad (k = 0, 1, \ldots), \\ x_{k+1} = z_k - \frac{f(y_k) f(x_k+\beta_k f(x_k)) \left(y_k - x_k + \frac{f(x_k)}{f[x_k, z_k]}\right)}{(f(y_k) - f(z_k))(f(x_k+\beta_k f(x_k)) - f(z_k))} + \frac{f(y_k)}{f[y_k, z_k]}. \end{cases} \tag{16}$$

If the parameter β_k in (15) and (16) has a constant value during the iterative process, then the order of the two-point method (15) is four and the order of the three-point method (16) is eight. These methods belong to the class of methods *without memory*. The convergence speed of these methods can be accelerated by calculating β_k recursively as the iteration proceeds. Then we shall have the corresponding *self-accelerating* methods *with memory*.

For example, the parameter β_k may be calculated recursively during the iterative process either as

$$\beta_k = -\frac{1}{\bar{f}'(\alpha)} = -\frac{\beta_{k-1} f(x_{k-1})}{f(x_{k-1} + \beta_{k-1} f(x_{k-1})) - f(x_{k-1})} \quad (\text{method(I)}) \tag{17}$$

or

$$\beta_k = -\frac{1}{\bar{f}'(\alpha)} = -\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \quad (\text{method (II)}) \tag{18}$$

for $k = 1,2,\ldots$, where $\bar{f}'(\alpha)$ denotes an approximation to $f'(\alpha)$. Then the methods $(15)_{(17/18)}$ and $(16)_{(17/18)}$ with memory have the increased R-order $2+\sqrt{6} \approx 4.45$ and $4+2\sqrt{5} \approx 8.472$, respectively, which is the subject of the forthcoming paper [23]. Before estimating the computational efficiency of the considered methods with/without memory, we give in Table 1 a review of their R-orders and a number of required function evaluations.

From Table 1 and the corresponding iterative formulas, we see that the methods (9) and (14) are realized by different function evaluations depending on the total number of performed iterative steps necessary to fulfill a given termination criteria (e.g., the required accuracy of approximations to the roots). For this reason it is not possible to compare the methods listed in Table 1 without taking into account the total number of iterations as a parameter. It is convenient to compute the efficiency index of an iterative method (IM) by the formula

$$E_s(IM) = (r^s)^{1/(\theta_1+\cdots+\theta_s)},$$

where s is the total number of iterations, r is the R-order and θ_j is the number of function evaluations at the jth iteration. Obviously, if $\theta_1 = \cdots = \theta_s = \theta$, then the above formula reduces to the well known formula $E(IM) = r^{1/\theta}$. This is the case with the methods (15) and (16).

From Tables 4–11 we observe that the interpolatory iterative method (9) produces more accurate approximations in all presented examples in relation to the method $(15)_{(17/18)}$ and all the tested fourth-order methods. The method (14), derived by inverse interpolation of the third degree, also possesses the domination to the method $(16)_{(17/18)}$ and all the tested eighth-order methods regarding the accuracy of approximations, see Tables 8–11. However, one should say that the method (9) uses one function evaluation more and the method (14) even two function evaluations more at the first iteration. These additional calculations decrease their computational efficiency, which is evident from Table 2. It is clear that their efficiency indices approach the efficiency indices of the methods $(15)_{(17/18)}$ and $(16)_{(17/18)}$ when the number of total iterations increases since the negative effect of expensive first iterations fades away.

Remark 3. At first sight, the need for three initial approximations to start the methods (14) is a disadvantage. This would have been true if we calculated additional initial approximations y_{-1} and z_{-1} by some iterative method, spending extra function evaluations. However, as explained in Remark 1, assuming that we have found an initial approximation x_0 (necessary for any iterative method), the next initial approximation y_{-1} can be calculated as $y_{-1} = N(x_0)$ not requiring extra cost since $N(x_0)$ is anyway needed at the first iteration. A lot of practical experiments showed that another approximation z_{-1} can be taken sufficiently close to the already calculated y_{-1}, for example

$$z_{-1} = y_{-1} \pm \delta, \quad \text{with} \quad \delta \approx |f(x_0)|/10.$$

Note that the methods (9) and (14) may converge slowly at the beginning of iterative process if the initial value x_0 (and, consequently, y_{-1} and z_{-1}) is not sufficiently close to the sought root α, but this is the case with all iterative methods with local

Table 1
Characteristics of multipoint methods with memory.

Methods	Number of function evaluations	R-order	Number of initial approximations
(15), β_k fixed	3	4	1
(9)	3^{+a}	$4.5\dot{6}$	2[b]
$(15)_{(17/18)}$, β_k by (17) or (18)	3	$4.4\dot{5}$	1
(16), β_k fixed	4	8	1
(14)	4^{+a}	$10.81\dot{5}$	3 [b]
$(16)_{(17/18)}$, β_k by (17) or (18)	4	$8.47\dot{2}$	1

[a] The number of function evaluation of the methods (9) and (14) is denoted with 3^+ and 4^+ to point that the number of function evaluations is respectively 4 and 6 at the first iteration.
[b] Taking $y_{-1} = N(x_0)$ (see Remarks 1 and 3), this number is decreased by one.

Table 2
Efficiency index as a function of the total number of iterations.

Methods	E_2	E_3	E_4
(15), β_k fixed	$1.58\dot{7}$	$1.58\dot{7}$	$1.58\dot{7}$
(9)	$1.54\dot{3}$	$1.57\dot{6}$	$1.59\dot{5}$
$(15)_{(17/18)}$, β_k by (17) or (18)	$1.64\dot{5}$	$1.64\dot{5}$	$1.64\dot{5}$
(16), β_k fixed	$1.68\dot{2}$	$1.68\dot{2}$	$1.68\dot{2}$
(14)	$1.6\dot{1}$	$1.66\dot{6}$	$1.69\dot{7}$
$(16)_{(17/18)}$, β_k by (17) or (18)	$1.70\dot{6}$	$1.70\dot{6}$	$1.70\dot{6}$

convergence. This possible drawback can be solved in most "non-pathological" situations by applying an efficient procedure for finding sufficiently good initial approximations recently proposed by Yun [31] and later discussed in [32].

5. Numerical examples

In this section we compare (1) the two-point method (9) with some existing two-point methods of the fourth order and (2) the three-point method (14) with some existing three-point methods of the eight order. The Kung-Traub methods with self-accelerating parameter $(15)_{(17/18)}$ and $(16)_{(17/18)}$ were also tested. The tested functions f, together with the sought zero α and used initial approximation x_0, are listed in Table 3. The two-point methods have been applied in Examples 1–4 and the three-point methods in Examples 5–8, noting that the second and fourth function in Table 3 have been tested by both types of methods.

To save space, we will give only references in which the tested methods were presented, except the King method which appears in both cases (1) and (2).

King's family [10]:

$$\begin{cases} u(x_k) = x_k - \frac{f(x_k)}{f'(x_k)}, \\ K_f(b; x_k) = u(x_k) - \frac{f(u(x_k))}{f'(x_k)} \cdot \frac{f(x_k)+bf(u(x_k))}{f(x_k)+(b-2)f(u(x_k))} \quad (b \in \mathbf{R}). \end{cases} \tag{19}$$

The following two-point optimal methods were also tested:

- *Jarratt's method* [7].
- *Maheshwari's method* [13].
- *Ren-Wu-Bi's method* [26].
- *Kung-Traub's method* [11] without derivatives (version 1), order 4.
- *Kung-Traub's method* [11] with derivative (version 2), order 4.

For brevity, in Tables 4–11 the Kung-Traub methods, versions 1 and 2, are denoted as K-T-1 and K-T-2, respectively. Recall that the Kung-Traub families of n-point methods ($n \geqslant 2$) have the order of convergence 2^n; we dealt with $n = 2$ in Examples 1–4 and $n = 3$ in Examples 5–8.

We employed the computer algebra system *Mathematica* with multiple-precision arithmetic relying on the GNU multiple-precision package GMP developed by Granlund [5]. The errors $|x_k - \alpha|$ for the few first iterations are given in Tables 4–11, where the denotation $A(-h)$ means $A \times 10^{-h}$.

(1) ***Two-point methods: numerical examples***
We observe from Tables 4–7 that the two-point methods (9) and $(15)_{(17/18)}$ with memory produce approximations of higher accuracy compared to the two-point methods of order four. Regarding these two methods, it is evident that the new method (9) gives more accurate approximations in all tested examples. This dominance is especially stressed in

Table 3
Test functions.

Example	Function	Root α	Initial approximation x_0
1	$(x-2)(x^{10}+x+1)e^{-5x}$	2	1.7
2, 5	$e^{-x^2+x+2} - \cos(x+1) + x^3 + 1$	−1	−0.5(Ex.2), − 0.2(Ex.5)
3	$\log(x^2+x+2) - x + 1$	4.1525907367...	5
4, 6	$e^x \sin x + \log(x^2+1)$	0	0.25 (Ex. 4),0.3 (Ex. 6)
7	$e^{x^2-1}\sin x + \cos 2x - 2$	1.4477948574...	1.3
8	$(x-1)(x^{10}+x^3+1)\sin x$	1	1.1

Table 4
Results of Example 1 – two-point methods.

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$
King's IM, $b = 0$	1.39(−2)	2.14(−9)	3.45(−37)	2.35(−148)
King's IM, $b = 1$	2.92(−2)	7.46(−8)	5.12(−31)	1.14(−123)
King's IM, $b = 2$	5.55(−2)	1.77(−6)	1.61(−25)	1.12(−101)
Jarratt's IM	1.37(−2)	4.57(−10)	1.05(−39)	2.97(−158)
Maheshwari's IM	4.24(−2)	4.58(−7)	7.26(−28)	4.60(−111)
Ren-Wu-Bi's IM	1.58(−2)	4.30(−9)	6.21(−36)	2.69(−143)
K-T-1, order 4, $\beta = 0.01$	1.96(−2)	1.09(−8)	2.31(−34)	4.68(−137)
K-T-1–(17), order 4.45, $\beta_0 = 0.01$	1.96(−2)	1.07(−9)	5.17(−45)	2.51(−201)
K-T-1–(18), order 4.45, $\beta_0 = 0.01$	1.96(−2)	7.85(−11)	3.36(−49)	2.42(−220)
K-T-2, order 4	1.96(−2)	1.08(−8)	2.23(−34)	4.12(−137)
Two-point IM (9)	4.50(−3)	1.18(−11)	1.37(−50)	4.20(−228)

Table 5
Results of Example 2 – two-point methods.

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$
King's IM, $b = 0$	4.26(−4)	2.12(−15)	1.31(−60)	1.93(−241)
King's IM, $b = 1$	2.57(−3)	2.44(−12)	1.99(−48)	8.80(−193)
King's IM, $b = 2$	4.79(−3)	2.42(−11)	1.58(−44)	2.91(−177)
Jarratt's IM	2.27(−3)	2.04(−12)	1.34(−48)	2.50(−193)
Maheshwari's IM	3.68(−3)	9.35(−12)	3.90(−46)	1.18(−183)
Ren-Wu-Bi's IM	1.50(−3)	1.63(−11)	2.26(−43)	8.23(−171)
K-T-1, order 4, $\beta = 0.01$	1.68(−3)	5.39(−13)	5.73(−51)	7.28(−203)
K-T-1–(17), order 4.45, $\beta_0 = 0.01$	1.68(−3)	3.66(−14)	1.39(−62)	8.29(−278)
K-T-1–(18), order 4.45, $\beta_0 = 0.01$	1.68(−3)	9.39(−15)	3.70(−65)	2.76(−289)
K-T-2, order 4	1.30(−3)	1.73(−13)	5.37(−53)	5.02(−211)
Two-point IM (9)	1.38(−5)	6.18(−24)	1.71(−107)	1.37(−488)

Table 6
Results of Example 3 – two-point methods.

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$
King's IM, $b = 0$	1.86(−4)	7.48(−19)	1.94(−76)	8.70(−307)
King's IM, $b = 1$	2.84(−4)	6.86(−18)	2.35(−72)	3.21(−290)
King's IM, $b = 2$	3.74(−4)	2.92(−17)	1.09(−69)	2.13(−279)
Jarratt's IM	2.16(−4)	1.51(−18)	3.61(−75)	1.18(−301)
Maheshwari's IM	3.29(−4)	1.49(−17)	6.35(−71)	2.08(−284)
Ren-Wu-Bi's IM	3.00(−5)	7.97(−23)	3.93(−93)	6.18(−371)
K-T-1, order 4, $\beta = 0.01$	2.34(−4)	2.50(−18)	3.25(−74)	9.26(−298)
K-T-1–(17), order 4.45, $\beta_0 = 0.01$	2.34(−4)	1.70(−20)	1.66(−92)	6.71(−413)
K-T-1–(18), order 4.45, $\beta_0 = 0.01$	2.34(−4)	5.06(−21)	1.10(−94)	1.16(−422)
K-T-2, order 4	2.37(−4)	2.65(−18)	4.11(−74)	2.39(−297)
Two-point IM (9)	1.70(−6)	3.81(−31)	3.88(−143)	8.36(−654)

Table 7
Results of Example 4 – two-point methods.

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$
King's IM, $b = 0$	6.54(−3)	1.28(−8)	1.96(−31)	1.08(−122)
King's IM, $b = 1$	1.17(−2)	3.82(−7)	4.99(−25)	1.45(−96)
King's IM, $b = 2$	1.49(−2)	1.58(−6)	2.45(−22)	1.43(−85)
Jarratt's IM	6.47(−3)	1.21(−8)	1.59(−31)	4.66(−123)
Maheshwari's IM	1.33(−2)	8.26(−7)	1.46(−23)	1.42(−90)
Ren-Wu-Bi's IM	1.89(−2)	3.16(−6)	2.93(−21)	2.16(−81)
K-T-1, order 4, $\beta = 0.01$	9.90(−3)	1.37(−7)	5.59(−27)	1.53(−104)
K-T-1–(17), order 4.45, $\beta_0 = 0.01$	9.90(−3)	3.45(−8)	1.81(−32)	8.28(−141)
K-T-1–(18), order 4.45, $\beta_0 = 0.01$	9.90(−3)	1.56(−8)	3.42(−34)	2.03(−148)
K-T-2, order 4	9.71(−3)	1.25(−7)	3.76(−27)	3.05(−105)
Two-point IM (9)	1.63(−3)	3.82(−12)	2.37(−51)	3.94(−230)

Table 8
Results of Example 5 – three-point methods.

Methods	$\|x_1 - \alpha\|$	$\|x_2 - \alpha\|$	$\|x_3 - \alpha\|$
K-T-1, order 8, $\beta = 0.01$	2.05(−4)	1.73(−32)	4.37(−257)
K-T-1–(17), order 8.47, $\beta_0 = 0.01$	2.05(−4)	1.59(−34)	7.75(−291)
K-T-1–(18), order 8.47, $\beta_0 = 0.01$	2.05(−4)	2.88(−35)	2.80(−297)
K-T-2, order 8	1.90(−4)	7.41(−33)	3.97(−260)
Bi-Wu-Ren's IM, method 1	2.14(−4)	1.34(−32)	3.22(−258)
Bi-Wu-Ren's IM, method 2	3.14(−4)	4.08(−31)	3.28(−246)
Petković-King's IM, order 8, $b = 0$	2.84(−4)	8.01(−32)	3.22(−252)
Petković-King's IM, order 8, $b = 1$	3.44(−4)	3.03(−31)	1.09(−247)
Neta-Petković's IM	1.62(−4)	2.26(−33)	3.17(−264)
Neta's IM (14)	5.51(−8)	7.76(−77)	6.94(−775)

Table 9
Results of Example 6 – three-point methods.

Methods	$\|x_1 - \alpha\|$	$\|x_2 - \alpha\|$	$\|x_3 - \alpha\|$
K-T-1, order 8, $\beta = 0.01$	8.13(−4)	2.16(−22)	5.45(−171)
K-T-1–(17), order 8.47, $\beta_0 = 0.01$	8.13(−4)	1.97(−23)	1.02(−189)
K-T-1–(18), order 8.47, $\beta_0 = 0.01$	8.13(−4)	4.40(−24)	1.08(−195)
K-T-2, order 8	7.84(−4)	1.56(−22)	3.96(−172)
Bi-Wu-Ren's IM, method 1	6.53(−5)	1.14(−32)	9.57(−255)
Bi-Wu-Ren's IM, method 2	4.08(−4)	2.44(−25)	3.53(−195)
Petković-King's IM, order 8, $b = 0$	1.92(−4)	1.85(−28)	1.39(−220)
Petković-King's IM, order 8, $b = 1$	5.71(−4)	1.18(−23)	4.08(−181)
Neta-Petković's IM	5.54(−4)	4.66(−24)	1.17(−184)
Neta's IM (14)	1.62(−6)	1.38(−55)	3.56(−552)

Table 10
Results of Example 7 – three-point methods.

Methods	$\|x_1 - \alpha\|$	$\|x_2 - \alpha\|$	$\|x_3 - \alpha\|$
K-T-1, order 8, $\beta = 0.01$	6.23(−4)	1.45(−23)	1.22(−180)
K-T-1–(17), order 8.47, $\beta_0 = 0.01$	6.23(−4)	7.85(−24)	4.01(−199)
K-T-1–(18), order 8.47, $\beta_0 = 0.01$	6.23(−4)	1.38(−25)	1.49(−208)
K-T-2, order 8	4.67(−4)	1.04(−24)	6.59(−190)
Bi-Wu-Ren's IM, method 1	3.68(−4)	8.88(−26)	1.03(−198)
Bi-Wu-Ren's IM, method 2	0.16	1.30(−4)	1.56(−28)
Petković-King's IM, order 8, $b = 0$	2.20(−5)	2.21(−36)	2.31(−284)
Petković-King's IM, order 8, $b = 1$	1.73(−3)	6.73(−20)	3.53(−151)
Neta-Petković's IM	1.20(−4)	6.37(−30)	3.92(−232)
Neta's IM (14)	1.70(−6)	2.28(−56)	1.59(−458)

Table 11
Results of Example 8 – three-point methods.

Methods	$\|x_1 - \alpha\|$	$\|x_2 - \alpha\|$	$\|x_3 - \alpha\|$
K-T-1, order 8, $\beta = 0.01$	3.89(−4)	9.36(−23)	1.05(−171)
K-T-1–(17), order 8.47, $\beta_0 = 0.01$	3.89(−4)	1.50(−23)	4.30(−188)
K-T-1–(18), order 8.47, $\beta_0 = 0.01$	3.89(−4)	2.76(−24)	7.60(−195)
K-T-2, order 8	3.41(−4)	2.94(−23)	9.00(−176)
Bi-Wu-Ren's IM, method 1	9.21(−4)	1.17(−19)	7.76(−147)
Bi-Wu-Ren's IM, method 2	1.35(−3)	1.94(−17)	3.09(−128)
Petković-King's IM, order 8, $b = 0$	1.11(−4)	3.05(−28)	9.75(−217)
Petković-King's IM, order 8, $b = 1$	5.79(−4)	5.68(−21)	4.93(−157)
Neta-Petković's IM	1.38(−4)	4.47(−27)	5.39(−207)
Neta's IM (14)	1.26(−6)	3.08(−54)	4.04(−536)

the case of Examples 2–4. Besides, from Table 1 we note that the R-order of convergence of the new method (9) (≈ 4.56) is slightly higher than the R-order of Kung-Traub's method $(15)_{(17/18)}$ with memory (≈ 4.45). On the other hand, the method (9) requires one function evaluation more in the first iteration (compared with $(15)_{(17/18)}$ and other two-point methods of optimal order four), which decreases its computational efficiency to a certain extent, see Table 2. For these reasons, it is hard to say which of the methods (9) and $(15)_{(17/18)}$ is better. It is only clear that a negative effect of the mentioned additional function evaluation in the first iteration decreases with the growth of the total number of iterations, increasing in this way the effectiveness of the new method (9) (see Table 2).

2*) *Three-point methods: numerical examples
Beside Neta's method (14) and already mentioned the Kung-Traub methods (with order 8 in this part), we have also tested the following three-point methods:

– *Bi-Wu-Ren's method*, choosing two variants denoted by method 1 and method 2 in the same manner as in [1].
– *Petković-King's method*, [21,24]. Note that a more general method, based on the Hermite interpolatory polynomial of degree 3, can use arbitrary two-point methods of optimal order four in the first two steps. We have chosen King's method, which is stressed by the given specific name of the tested method.
– *Neta-Petković's method*, [19].
Note that several three-point methods with optimal order eight have appeared recently, e.g., [2–4,12,18,25,27,28,30]. However, these methods have a similar convergence behavior to the tested three-point methods and we omitted them.

From Tables 8–11 we notice that the method (14), constructed by inverse interpolation, produces approximations of the greatest accuracy. Also, its *R*-order (≈10.815) is higher than the *R*-order of the remaining tested methods. On the other hand, the method (14) requires two function evaluations more in the first iteration, which decreases its computational efficiency (see Table 2). Therefore, the discussion and comments given above for the two-point methods also hold for the three-point methods.

Acknowledgement

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

References

[1] W. Bi, Q. Wu, H. Ren, A new family of eight-order iterative methods for solving nonlinear equations, Appl. Math. Comput. 214 (2009) 236–245.
[2] W. Bi, H. Ren, Q. Wu, Three-step iterative methods with eight-order convergence for solving nonlinear equations, J. Comput. Appl. Math. 225 (2009) 105–112.
[3] J. Džunić, M.S. Petković, L.D. Petković, A family of optimal three-point methods for solving nonlinear equations using two parametric functions, Appl. Math. Comput. 217 (2011) 7612–7619.
[4] Y.H. Geum, Y.I. Kim, A multi-parameter family of three-step eighth-order iterative methods locating a simple root, Appl. Math. Comput. 215 (2010) 3375–3382.
[5] T. Granlund, GNU MP; The GNU Multiple Precision Arithmetic Library, edition 5.0.1, 2010.
[6] J. Herzberger, Über Matrixdarstellungen für iterationverfahren bei nichtlinearen Gleichungen, Computing 12 (1974) 215–222.
[7] P. Jarratt, Some fourth order multipoint methods for solving equations, Math. Comput. 20 (1966) 434–437.
[8] P. Jarratt, Some efficient fourth-order multipoint methods for solving equations, BIT 9 (1969) 119–124.
[9] R.F. King, A fifth order family of modified Newton methods, BIT 11 (1971) 409–412.
[10] R. King, A family of fourth order methods for nonlinear equations, SIAM J. Numer. Anal. 10 (1973) 876–879.
[11] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, J. ACM 21 (1974) 643–651.
[12] L. Liu, X. Wang, Eighth-order methods with high efficiency index for solving nonlinear equations, Appl. Math. Comput. 215 (2010) 3449–3454.
[13] A.K. Maheshwari, A fourth-order iterative method for solving nonlinear equations, Appl. Math. Comput. 211 (2009) 383–391.
[14] B. Neta, A sixth order family of methods for nonlinear equations, Int. J. Comput. Math. 7 (1979) 157–161.
[15] B. Neta, On a family of multipoint methods for nonlinear equations, Int. J. Comput. Math. 9 (1981) 353–361.
[16] B. Neta, A new family of higher order methods for solving equations, Int. J. Comput. Math. 14 (1983) 191–195.
[17] B. Neta, Several new methods for solving equations, Int. J. Comput. 23 (1988) 265–282.
[18] B. Neta, A.N. Johnson, High order nonlinear solver, J. Comput. Methods Sci. Eng. 8 (2008) 245–250.
[19] B. Neta, M.S. Petković, Construction of optimal order nonlinear solvers using inverse interpolation, Appl. Math. Comput. 217 (2010) 2448–2455.
[20] A.M. Ostrowski, Solution of Equations and Systems of Equations, Academic Press, New York, 1960.
[21] M.S. Petković, On a general class of multipoint root-finding methods of high computational efficiency, SIAM J. Numer. Anal. 47 (2010) 4402–4414.
[22] 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.
[23] M.S. Petković, B. Neta, L.D. Petković, On the Kung-Traub family of multipoint methods with memory, private communication.
[24] 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.
[25] M.S. Petković, L.D. Petković, J. Džunić, A class of three-point root-solvers of optimal order of convergence, Appl. Math. Comput. 216 (2010) 671–676.
[26] 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.
[27] J.R. Sharma, R. Sharma, A new family of modified Ostrowskis methods with accelerated eighth order convergence, Numer. Algor. 54 (2010) 445–458.
[28] R. Thukral, M.S. Petković, Family of three-point methods of optimal order for solving nonlinear equations, J. Comput. Appl. Math. 233 (2010) 2278–2284.
[29] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
[30] X. Wang, L. Liu, New eighth-order iterative methods for solving nonlinear equations, J. Comput. Appl. Math. 234 (2010) 1611–1620.
[31] B.I. Yun, A non-iterative method for solving non-linear equations, Appl. Math. Comput. 198 (2008) 691–699.
[32] B.I. Yun, M.S. Petković, Iterative methods based on the signum function approach for solving nonlinear equations, Numer. Algor. 52 (2009) 649–662.