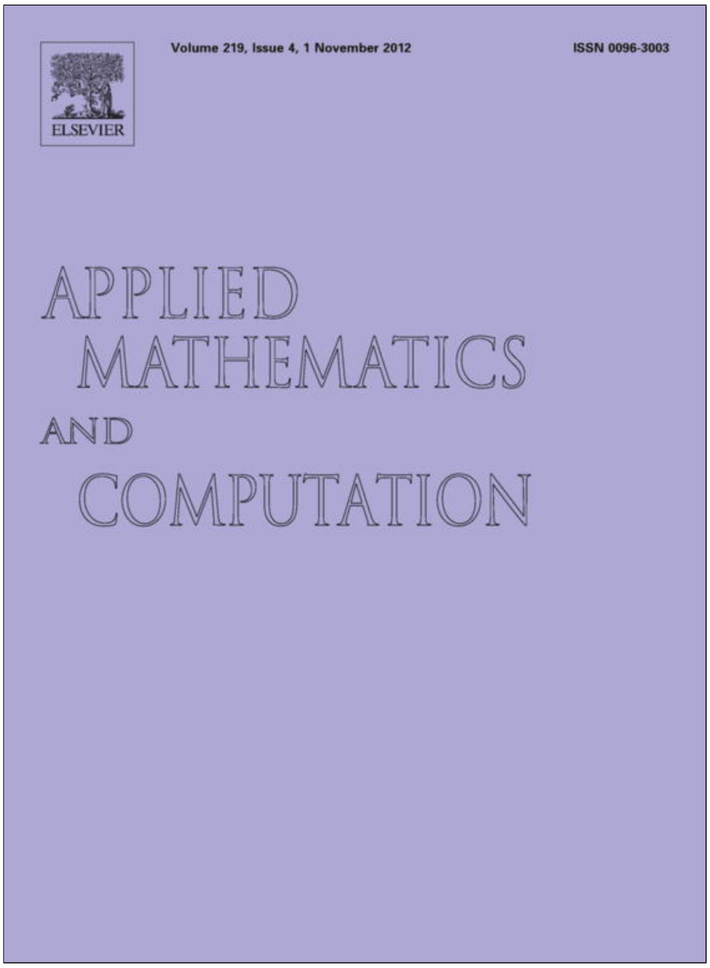
Volume 219, Issue 4, 1 November 2012
ISSN 0096-3003
ELSEVIER
APPLIED MATHEMATICS AND COMPUTATION

Applied Mathematics and Computation 219 (2012) 1749–1760

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

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

On an efficient family of derivative free three-point methods for solving nonlinear equations

Xiaofeng Wang [a,b], Jovana Džunić [c,*], Tie Zhang [a]

[a] College of Sciences, Northeastern University, Shenyang, 110819 Liaoning, China
[b] Department of Mathematics, Bohai University, Jinzhou, 121013 Liaoning, China
[c] Faculty of Electronic Engineering, Department of Mathematics, University of Niš, 18000 Niš, Serbia

ARTICLE INFO

Keywords:
Nonlinear equations
Multipoint methods
Methods with memory
Acceleration of convergence
Order of convergence
Computational efficiency

ABSTRACT

New three-step derivative free families of three-point methods for solving nonlinear equations are presented. First, a new family without memory of optimal order eight, consuming four function evaluations per iteration, is proposed by using two weight functions. The improvement of the convergence rate of this basic family, even up to 50%, is obtained without any additional function evaluation using a self-accelerating parameter. This varying parameter is calculated in each iterative step employing only information from the current and the previous iteration, defining in this way a family with memory. The self-accelerating parameter is calculated applying Newton's interpolating polynomials of degree scaling from 1 to 4. The corresponding *R*-orders of convergence are increased from 8 to 10, 11, $6+4\sqrt{2}\approx 11.66$ and 12, providing very high computational efficiency of the proposed methods with memory. Another convenient fact is that these methods do not use derivatives. Numerical examples and comparison with the existing three-point methods are included to confirm theoretical results.

1. Introduction

The most efficient existing root-solvers are based on multipoint iterations. This class of methods overcomes theoretical limits of one-point methods related to the convergence order and computational efficiency. With these important advantages multipoint methods have attracted considerable attention at the beginning of the 21st century. The development of symbolic computation and multi-precision arithmetics have additionally contributed to the rapid development of multipoint methods.

Derivative free n-point methods with optimal order 2^n, where $n+1$ is a fixed number of function evaluations per iterations, are of special interest since they provide the construction of accelerated multipoint methods with memory of great computational efficiency. This kind of methods, called *optimal* multipoint methods, is the subject of this paper. First, we derive a family of three-point methods without memory with order eight (Section 2) and give some special cases used in numerical tests in Section 5. Bearing in mind that higher-order multipoint methods without memory were already derived in the literature, see, e.g., [1–19], the proposed family of three-point methods without memory is competitive with the existing three-point optimal methods. The convergence rate of this family is significantly increased in Section 3 using an old idea by Traub [16], recently extended in [12]. The key idea that provides the order acceleration lies in a special form of the error relation and a convenient choice of a free parameter. We define a self-accelerating parameter, which is calculated during the

* Corresponding author.
E-mail addresses: w200888w@163.com (X. Wang), jovana.dzunic@elfak.ni.ac.rs (J. Džunić), ztmath@163.com (T. Zhang).

0096-3003/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2012.05.069

iterative process using Newton's interpolating polynomial. Accelerating technique relies on information from the current and the previous iterative step, defining in this way three-point methods with memory. The significant increase of convergence speed is achieved without additional function evaluations, which is the main advantage of these methods compared to the existing multi-point methods.

Section 4 is devoted to theoretical results connected to the R-order of the methods with memory. We show that, depending on the accelerating technique (the degree of the interpolating polynomial), the convergence rate can be improved up to 50%. Numerical results given in Section 5 confirm theoretical results and demonstrate very fast convergence and high computational efficiency of the proposed methods.

2. Derivative free three-point methods

Let α be a simple real zero of a real function $f : D \subset \boldsymbol{R} \to \boldsymbol{R}$ and let x_0 be an initial approximation to α. We take the tripled Newton method as the base for constructing a new three-step scheme

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - \frac{f(y_k)}{f'(y_k)}, \\ x_{k+1} = z_k - \frac{f(z_k)}{f'(z_k)}, \end{cases} \tag{1}$$

where $k = 0, 1, \ldots$ is the iteration index. The order of convergence of scheme (1) is eight but its computational efficiency is low. To improve this disadvantage, we substitute derivatives in all three steps by suitable approximations that use available data. In the first step, we approximate

$$f'(x_k) \approx f[x_k, w_k], \text{ where } w_k = x_k + \gamma f(x_k), \quad \gamma \in \boldsymbol{R}/\{0\}$$

and $f[x,y] = \frac{f(x)-f(y)}{x-y}$ denotes a divided difference. Similarly, the other two derivatives can be approximated by secants with additional adjustments carried out by weight functions with one and two variables. We introduce approximations

$$f'(y_k) \approx \frac{f[y_k, w_k]}{H(t_k)}, \quad t_k = \frac{f(y_k)}{f(x_k)},$$
$$f'(z_k) \approx \frac{f[z_k, w_k]}{G(t_k, s_k)}, \quad s_k = \frac{f(z_k)}{f(y_k)},$$

used in the second and the third step of (1), where G and H are weight functions. The following iterative family of three-point methods is obtained:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f[x_k,w_k]}, & w_k = x_k + \gamma f(x_k), \\ z_k = y_k - \frac{f(y_k)}{f[y_k,w_k]} H(t_k), & t_k = \frac{f(y_k)}{f(x_k)}, \\ x_{k+1} = z_k - \frac{f(z_k)}{f[z_k,w_k]} G(t_k, s_k), & s_k = \frac{f(z_k)}{f(y_k)}. \end{cases} \tag{2}$$

The weight functions H and G should be chosen so that the family (2) is optimal, that is, its order should be at least eight. For simplicity we drop the subscript on approximations. The new approximation to the sought root will be denoted with the symbol $\hat{}$. To determine conditions under which the iterative scheme (2) is optimal, we will use Taylor's representation of functions appearing in (2) when x is in the neighborhood of α. When $x \to \alpha$, than $y \to \alpha$ obviously, and we wish $z, \ \hat{x} \to \alpha$ even faster.

Introduce the abbreviations

$$\varepsilon = x - \alpha, \quad \varepsilon_y = y - \alpha, \quad \varepsilon_z = z - \alpha, \quad \varepsilon_w = w - \alpha,$$
$$c_j = \frac{f^{(j)}(\alpha)}{j! f'(\alpha)} \quad (j = 2, 3, \ldots), \quad H_j = H^{(j)}(0),$$
$$G_{i,j} = \left[\frac{\partial^{i+j}}{\partial^i t \partial^j s} G(t,s)\right]_{(t,s)=(0,0)} \qquad (i,j = 0, 1, 2, \ldots).$$

Thus,

$$H(t) = H_0 + H_1 t + \frac{1}{2} H_2 t^2 + \frac{1}{6} H_3 t^3 + \ldots, \tag{3}$$

$$G(t,s) = G_{0,0} + G_{1,0} t + G_{0,1} s + \frac{1}{2}\left(G_{2,0} t^2 + 2G_{1,1} ts + G_{0,2} s^2\right) + \frac{1}{6}\left(G_{3,0} t^3 + 3G_{2,1} t^2 s + 3G_{1,2} ts^2 + G_{0,3} s^3\right) + \ldots \tag{4}$$

To deal with rather cumbersome expressions that appear in a standard convergence analysis of iterative methods, we will employ symbolic computation in the computational software package *Mathematica*. The results of the convergence analysis will be explained later by the comments C1–C7. The following abbreviations were used:

$$\texttt{e} = \varepsilon = x - \alpha, \quad \texttt{ew} = \varepsilon_w = w - \alpha = \varepsilon + \gamma f(x),$$
$$\texttt{ey} = \varepsilon_y = y - \alpha, \quad \texttt{ez} = \varepsilon_z = z - \alpha, \quad \texttt{e1} = \hat{\varepsilon} = \hat{x} - \alpha,$$
$$\texttt{cj} = c_j, \quad \texttt{Hj} = H_j, \quad \texttt{Gij} = G_{i,j}, \quad \texttt{g} = \gamma,$$
$$\texttt{fx} = f(x), \quad \texttt{fy} = f(y), \quad \texttt{fz} = f(z), \quad \texttt{f1a} = f'(\alpha), \quad \texttt{fxw} = \frac{f(x) - f(w)}{x - w}.$$

Program written in *Mathematica*

```
fx=f1a*(e+c2*e^2+c3*e^3+c4*e^4+c5*e^5+c6*e^6+c7*e^7+c8*e^8);
ew=e+g*fx; fw=fx/.e→ew; fxw=Series[(fx-fw)/(e-ew),{e,0,8}]//Simplify;
ey=Series[e-fx/fxw,{e,0,8}]; fy=f1a*(ey+c2*ey^2+c3*ey^3+c4*ey^4);
fyw=Series[(fy-fw)/(ey-ew),{e,0,8}]//Simplify; t=Series[fy/fx,{e,0,8}];
H=H0+H1*t+1/2*H2*t^2+1/6*H3*t^3+1/24*H4*t^4;
ez=Series[ey-H*fy/fyw,{e,0,8}];
b2=Coefficient[ez,e,2]//FullSimplify
C1: Out[b2]=-c2*(1+g*f1a)*(H0-1)
H0=1;b3=Coefficient[ez,e,3]//FullSimplify
C2: Out[b3]=-c2^2*(1+g*f1a)^2*(H1-1)
H1=1;b4=Coefficient[ez,e,4]//FullSimplify
Out[b4]=-1/2*c2*(1+g*f1a)^2*(2*c3+c2^2*(H2-6+g*f1a*(H2-2)))
fz=f1a*(ez+c2*ez^2); s=Series[fz/fy,{e,0,8}]; fzw=Series[(fz-fw)/(ez-ew),{e,0,8}];
G=G00+G10*t+G01*s+1/2*(G20*t^2+2*G11*t*s+G02*s^2)
  +1/6*(G30*t^3+3*G21*t^2*s+3*G12*t*s^2+G03*s^3)
  +1/24*(G40*t^4+4*G31*t^3*s+6*G22*t^2*s^2+4*G13*t*s^3+G04*s^4);
e1=Series[ez-G*fz/fzw,{e,0,8}];
a4=Coefficient[e1,e,4]//FullSimplify
C3: Out[a4]=1/2*c2*(1+g*f1a)^2*(G00-1)*(2*c3+c2^2*(g*f1a*(H2-2)+H2-6))
G00=1; a5=Coefficient[e1,e,5]//FullSimplify
C4: Out[a5]=1/2*c2^2*(1+g*f1a)^3*(G10-1)*(2*c3+c2^2*(g*f1a*(H2-2)+H2-6))
G10=1; a6=Coefficient[e1,e,6]//FullSimplify
C5: Out[a6]=-1/4*c2*(1+g*f1a)^3*(2*c3+c2^2*(g*f1a*(H2-2)+H2-6))(2*c3
          *(G01-1)+c2^2*(g*f1a*(G01*(H2-2)-G20+2)
          +G01*(H2-6)-G20+6))
G01=1; G20=H2; a7=Coefficient[e1,e,7]//FullSimplify
C6: Out[a7]=-1/12*c2^2*(1+g*f1a)^4*(2*c3+c2^2*(g*f1a*(H2-2)+H2-6))*(6*c3
          *(G11-2)+c2^2*(g*f1a*(3*G11*(H2-2)-3*H2+H3-G30+6)
          +3*G11*(H2-6)-3*H2+H3-G30+24))
G11=2; G30=3*H2+H3-6/(1+g*fxw)-6;
a8=Coefficient[e1,e,8]//FullSimplify
C7: Out[a8]=1/48*c2*(1+g*f1a)^4*(2*c3+c2^2*(g*f1a*(H2-2)+H2-6))
          *(-24*c2*c4+12*c3^2*(G02-2)+12*c2^2*c3*(22-G21+G02*(H2-6)
          -H2-g*f1a*(-6+G21-G02*(H2-2)+H2))
          +c2^4*(-312+G40-6*G21*(H2-6)+3*G02*(H2-6)^2+60*H2-4*H3
          +(g*f1a)^2*(-24+G40-6*G21*(H2-2)+3*G02*(H2-2)^2+24*H2-4*H3
          -H4)+2*g*f1a*(-84+G40-6*G21*(H2-4)+3*G02*(H2-6)*(H2-2)
          +42*H2-4*H3-H4)-H4))
```

Comments C1 and **C2:** Since one of our goals is to provide $\varepsilon_z = O(\varepsilon^4)$, meaning that the first two steps of the scheme (2) represent an optimal family of two-step methods ($\varepsilon_z = o(\varepsilon_y)$ is necessary for $s \to 0$ when $\alpha \to 0$), than the coefficients (b2 and b3) with the terms ε^2 and ε^3 in ε_z representation have to be annulated. We obtain this by annulation of the expressions in shadowed boxes, which leads to the choice H_0 = H0 = 1 and H_1 = H1 = 1.

Comment C3: In the same manner, to provide $\widehat{\varepsilon} = O(\varepsilon^8)$, the representation of $\widehat{\varepsilon}$ must not have terms in ε of less degree than eight. Thus, the coefficient a4 vanishes if the shaded expression is 0. Then the choice $G_{0,0}$ = G00 = 1 comes as obvious.

We use the same deduction in **C4–C6** to determine $G_{1,0} = 1,\ G_{0,1} = 1,\ G_{2,0} = H_2$ requiring that the coefficients a4, a5, a6 with $\varepsilon^4,\ \varepsilon^5$ and ε^6 are 0.

Comment C7: After determining a7, the coefficient with ε^7, the first obvious choice is to eliminate 6*c3*(G11 − 2). It is easily done by setting G11 = 2. The reminding part of the coefficient a7 that we wish to annulate is than

3 * H2 + H3 − 12 − G30 + g * f1a * (3 * H2 + H3 − 6 − G30)

which leads to the choice G30 = 3 * H2 + H3 − 6 − $\frac{6}{1+g*f1a}$. However, the value f1a = $f'(\alpha)$ is not available, therefore we use an available value fxw = $f[x,w]$ to approximate $f'(\alpha)$, and come to the necessary conditions for the weight functions H and G

$$\begin{aligned} &H_0 = 1, \quad H_1 = 1, \quad |H_2|,\ |H_3| < \infty, \\ &G_{0,0} = 1, \quad G_{1,0} = 1, \quad G_{0,1} = 1, \quad G_{2,0} = H_2, \quad G_{1,1} = 2, \\ &G_{3,0} = 3H_2 + H_3 - \frac{6}{1+\gamma f[x,w]} - 6. \end{aligned} \tag{5}$$

Therefore, functions H and G should have Taylor's representation in the neighborhood of 0 (for H) and (0,0) (for G) of the form

$$\begin{aligned} &H(t) = 1 + t + \frac{1}{2}at^2 + \frac{1}{6}bt^3 + \cdots, \\ &G(t,s) = 1 + t + s + \frac{1}{2}at^2 + 2ts + \frac{1}{2}cs^2 + \left(\frac{1}{2}a + \frac{1}{6}b - \frac{1}{1+\gamma f[x,w]} - 1\right)t^3 + \ldots \end{aligned} \tag{6}$$

Summarizing the above results, we can state the convergence theorem for the family (2).

Theorem 1. *If an initial approximation x_0 is sufficiently close to a simple zero α of f and the weight functions H and G satisfy the conditions (5), then the convergence order of the family of three-point methods (2) is equal to eight.*

The following error relations, obtained with the help of the given program are very important in later discussions

$$\varepsilon_{k,w} = w_k - \alpha = (1 + \gamma f'(\alpha))\varepsilon_k + O(\varepsilon_k^2), \tag{7}$$

$$\varepsilon_{k,y} = y_k - \alpha = c_2(1 + \gamma f'(\alpha))\varepsilon_k^2 + O(\varepsilon_k^3), \tag{8}$$

$$\varepsilon_{k,z} = z_k - \alpha = a_{k,4}(1 + \gamma f'(\alpha))^2\varepsilon_k^4 + O(\varepsilon_k^5), \tag{9}$$

$$\varepsilon_{k+1} = x_{k+1} - \alpha = a_{k,8}(1 + \gamma f'(\alpha))^4\varepsilon_k^8 + O(\varepsilon_k^9), \tag{10}$$

where $a_{k,4}$ and $a_{k,8}$ are determined for the specific choice of the weight functions H and G.

We will consider now some particular methods following from the family (2). We give examples for the weight functions H and G of simple form that satisfy the conditions (5).

Example 1.

From Taylor's expansion (6), we conclude that the choice $a = c = 2$ gives a sub-family of (2)

$$\begin{cases} y = x - \frac{f(x)}{f[x,w]}, & w = x + \gamma f(x), \\ z = y - \frac{f(y)}{f[y,w]}H(t), & t = \frac{f(y)}{f(x)}, \\ \hat{x} = z - \frac{f(z)}{f[z,w]}K(v), & v = t + \frac{f(z)}{f(y)}, \end{cases}$$

where functions H and K satisfy the conditions

$$\begin{aligned} &H(0) = 1, \quad H'(0) = 1, \quad H''(0) = 2, \quad H'''(0) < \infty, \\ &K(0) = 1, \quad K'(0) = 1, \quad K''(0) = 2, \\ &K'''(0) = 3H''(0) + H'''(0) - \frac{6}{1+\gamma f[x,w]} - 6. \end{aligned}$$

Particular methods arising from this sub-family are included in examples below.

Example 2. Choosing $H_1(t) = 1 + t$ allows the following choices for G

$$\begin{aligned} &G_1(t,s) = 1 + t + s + 2ts - (1+\lambda)t^3, \quad \lambda = \frac{1}{1+\gamma f[x,w]}, \\ &G_2(t,s) = \frac{1+\lambda t^2}{1-(t+s)+(1+\lambda)t^2}. \end{aligned}$$

Example 3. Another particular choice can be $H_2(t) = \frac{1}{1-t}$ with G given by

$$G_3(t,s) = \frac{1+(t+s)+\lambda(t+s)^2}{1+(\lambda-1)(t+s)^2}, \quad G_4(t,s) = \frac{1+\lambda t^2}{1-(t+s)+\lambda t^2}.$$

Example 4. Functions $H_3(t) = 1+t+t^2$, and $H_3^* = \left(\frac{1+t+1/8t^2}{1+1/2t}\right)^2$ can be both combined with

$$G_5(t,s) = \frac{1+\lambda(t+s)^2-(t+s)^3}{1-(t+s)+\lambda(t+s)^2}, \quad G_6(t,s) = 1+(t+s)+(t+s)^2-\lambda(t+s)^3.$$

Example 5. Another choice $H_4(t) = 1+t+t^3$ can be combined with

$$G_7(t,s) = \frac{1-s+t+\lambda t^2}{1-2s+\lambda t^2},$$

$$G_8(t,s) = \frac{1+\gamma f[x,w](1-t^2)}{(1+\gamma f[x,w])(1-t-s)+t^2}.$$

Example 6. A square-root example $H_5(t) = \sqrt{\frac{3+4t+5t^2}{3-2t}}$ allows the choices

$$G_9(t,s) = \frac{1+(1+\lambda)t^2}{1-t-s+(1+\lambda)t^2},$$

$$G_{10}(t,s) = \frac{1+(1+\lambda)(t+s)(1+t+s)}{1+\lambda(t+s)}.$$

3. New families of three-point methods with memory

We observe from (10) that the order of convergence of the family (2) is eight when $\gamma \neq -1/f'(\alpha)$. With the choice $\gamma = -1/f'(\alpha)$, it can be proved that the order of the family (2) can reach 12. Since the value $f'(\alpha)$ is unknown in practice, instead of that, we could use an approximation $\bar{f}'(\alpha) \approx f'(\alpha)$, based on available information. Then, setting $\gamma = -1/\bar{f}'(\alpha)$ in (2), we can achieve that the order of convergence of the modified methods exceeds eight without the use of any new function evaluations.

In this paper we consider Newton's interpolation as the method for approximating $f'(\alpha)$ in the following four cases:

(I) $\bar{f}'(\alpha) = N_1'(x_k)$ (N1 approach), where $N_1(t) = N_1(t; x_k, z_{k-1})$ is Newton's interpolating polynomial of first degree, set through two best available approximations (nodes) x_k and z_{k-1}.

(II) $\bar{f}'(\alpha) = N_2'(x_k)$ (N2 approach), where $N_2(t) = N_2(t; x_k, z_{k-1}, y_{k-1})$ is Newton's interpolating polynomial of second degree, set through three best available approximations (nodes) $x_k,\ z_{k-1}$ and y_{k-1}.

(III) $\bar{f}'(\alpha) = N_3'(x_k)$ (N3 approach), where $N_3(t) = N_3(t; x_k, z_{k-1}, y_{k-1}, w_{k-1})$ is Newton's interpolating polynomial of third degree, set through four best available approximations (nodes) $x_k,\ z_{k-1},\ y_{k-1}$ and w_{k-1}.

(IV) $\bar{f}'(\alpha) = N_4'(x_k)$ (N4 approach), where $N_4(t) = N_4(t; x_k, z_{k-1}, y_{k-1}, x_{k-1}, w_{k-1})$ is Newton's interpolating polynomial of fourth degree, set through five best available approximations (nodes) $x_k,\ z_{k-1},\ y_{k-1},\ x_{k-1}$ and w_{k-1}.

The main idea in constructing methods with memory consists of the calculation of the parameter $\gamma = \gamma_k$ as the iteration proceeds by the formula $\gamma_k = -1/\bar{f}'(\alpha)$ for $k = 1, 2, \ldots$ It is assumed that the initial estimate γ_0 should be chosen before starting the iterative process, for example, using one of the ways proposed in [16, p. 186]. Regarding the above methods (I)–(IV), we present the following four formulae for the calculation of γ_k:

$$\gamma_k = -\frac{1}{N_j'(x_k)} \quad (j = 1, 2, 3, 4), \tag{11}$$

where,

$$N_1'(x_k) = \left[\frac{d}{dt}N_1(t)\right]_{t=x_k} = \left[\frac{d}{dt}(f(x_k) + f[x_k, z_{k-1}](t - x_k))\right]_{t=x_k} = f[x_k, z_{k-1}], \tag{12}$$

$$\begin{aligned} N_2'(x_k) &= \left[\frac{d}{dt}N_2(t)\right]_{t=x_k} = \left[\frac{d}{dt}(f(x_k)+f[x_k,z_{k-1}](t-x_k)+f[x_k,z_{k-1},y_{k-1}](t-x_k)(t-z_{k-1}))\right]_{t=x_k} \\ &= f[x_k,z_{k-1}]+f[x_k,z_{k-1},y_{k-1}](x_k-z_{k-1}) = f[x_k,z_{k-1}]+f[x_k,y_{k-1}]-f[z_{k-1},y_{k-1}], \end{aligned} \tag{13}$$

$$\begin{aligned} N_3'(x_k) &= \left[\frac{d}{dt}N_3(t)\right]_{t=x_k} \\ &= \left[\frac{d}{dt}(f(x_k)+f[x_k,z_{k-1}](t-x_k)+f[x_k,z_{k-1},y_{k-1}](t-x_k)(t-z_{k-1})+f[x_k,z_{k-1},y_{k-1},w_{k-1}](t-x_k)(t-z_{k-1})(t-y_{k-1}))\right]_{t=x_k} \\ &= f[x_k,z_{k-1}]+f[x_k,z_{k-1},y_{k-1}](x_k-z_{k-1})+f[x_k,z_{k-1},y_{k-1},w_{k-1}](x_k-z_{k-1})(x_k-y_{k-1}), \end{aligned} \tag{14}$$

$$\begin{aligned} N_4'(x_k) &= \left[\frac{d}{dt}N_4(t)\right]_{t=x_k} \\ &= \left[\frac{d}{dt}(f(x_k)+f[x_k,z_{k-1}](t-x_k)+f[x_k,z_{k-1},y_{k-1}](t-x_k)(t-z_{k-1})+f[x_k,z_{k-1},y_{k-1},x_{k-1}](t-x_k)(t-z_{k-1})(t-y_{k-1})\right. \\ &\quad \left.+f[x_k,z_{k-1},y_{k-1},x_{k-1},w_{k-1}](t-x_k)(t-z_{k-1})(t-y_{k-1})(t-x_{k-1}))\right]_{t=x_k} \\ &= f[x_k,z_{k-1}]+f[x_k,z_{k-1},y_{k-1}](x_k-z_{k-1})+f[x_k,z_{k-1},y_{k-1},x_{k-1}](x_k-z_{k-1})(x_k-y_{k-1}) \\ &\quad +f[x_k,z_{k-1},y_{k-1},x_{k-1},w_{k-1}](x_k-z_{k-1})(x_k-y_{k-1})(x_k-x_{k-1}). \end{aligned} \tag{15}$$

Divided differences of higher order are defined recursively. A divided difference of order m, denoted by $f[t_0,t_1,\ldots,t_m]$, is defined as

$$f[t_0,t_1,\ldots,t_m] = \frac{f[t_1,\ldots,t_m]-f[t_0,t_1,\ldots,t_{m-1}]}{t_m-t_0}, \quad m \geqslant 2.$$

Since γ_k is calculated as the iteration proceeds using (12)–(15) in (11), the family of three-point methods with memory corresponding to the family (2) follows

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f[x_k,w_k]}, & w_k = x_k+\gamma_k f(x_k), \\ z_k = y_k - \frac{f(y_k)}{f[y_k,w_k]}H(t_k), & t_k = \frac{f(y_k)}{f(x_k)}, \\ x_{k+1} = z_k - \frac{f(z_k)}{f[z_k,w_k]}G(t_k,s_k), & s_k = \frac{f(z_k)}{f(y_k)}, \end{cases} \tag{16}$$

where H and G are one and two-valued weight functions that satisfy (6). 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 the data available from the current and the previous iterative step.

4. Convergence theorem

To estimate the convergence rate of the family of three-point methods with memory (16), where γ_k is calculated using one of the formulae given by (11) and $N_j'(x_k)$ by (12)–(15), we will use the concept of the R-order of convergence introduced by Ortega and Rheinboldt [9]. Now we state the convergence theorem for the family (16) of three-point methods with memory.

Theorem 2. *Let the varying parameter γ_k in the iterative scheme (16) be calculated by (11) $\wedge$ {(12)–(15)}. If an initial approximation x_0 is sufficiently close to a simple zero α of f, then the R-order of convergence of the three-point methods (16)–(N1), (16)–(N2), (16)–(N3) and (16)–(N4) with memory is at least $10, 11, 6+4\sqrt{2} \approx 11.66$ and 12, respectively.*

Proof. We will use Herzberger's matrix method [6] to determine the R-order of convergence for each case (N1)–(N4).The lower bound of order of a single step s-point method $x_k = G(x_{k-1},x_{k-2},\ldots,x_{k-s})$ is the spectral radius of a matrix $M^{(s)} = (m_{ij})$, associated to this method, with elements

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

The lower bound of 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 $M = M_1 \cdot M_2 \cdots M_s$.

We can express each approximation $x_{k+1},\ z_k,\ y_k$, and w_k as a function of available information $f(z_k),\ f(y_k),\ f(w_k),\ f(x_k)$ from the k-th iteration and $f(z_{k-1}),\ f(y_{k-1}),\ f(w_{k-1}),\ f(x_{k-1})$ from the previous iteration, depending on the accelerating technique. According to the relations (16) and (N1)–(N4) we form the respective matrices. More details and illustrations about the construction of Herzberger's M-matrices can be found in [11].

Now we determine the R-order of convergence of the family (16) for all approaches (N1)–(N4) applied for the calculation of γ_k.

Method (N1), γ_k is calculated using (12):

We use the following matrices to express informational dependence

$$x_{k+1} = \varphi_1(z_k, y_k, w_k, x_k, z_{k-1}) \to M_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$z_k = \varphi_2(y_k, w_k, x_k, z_{k-1}, y_{k-1}) \to M_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$y_k = \varphi_3(w_k, x_k, z_{k-1}, y_{k-1}, w_{k-1}) \to M_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$w_k = \varphi_4(x_k, z_{k-1}, y_{k-1}, w_{k-1}, x_{k-1}) \to M_4 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The matrix $M^{(N_1)}$ corresponding to the multi-point method (16)–(N1) is

$$M^{(N_1)} = M_1M_2M_3M_4 = \begin{bmatrix} 8 & 4 & 0 & 0 & 0 \\ 4 & 2 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and its eigenvalues are $\{10, 0, 0, 0, 0\}$. Since the spectral radius of the matrix $M^{(N_1)}$ is $r = 10$, we conclude that the R-order of the methods with memory (16)–(N1) is at least *ten*.

Method (N2), γ_k is calculated using (13):

Similarly, matrices $M_1,\ M_2$ and M_3 again express informational dependence for $x_{k+1},\ z_k$ and y_k, while for w_k we have

$$M_4 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The matrix $M^{(N_2)}$ corresponding to the multi-point method (16)–(N2) is now

$$M^{(N_2)} = M_1M_2M_3M_4 = \begin{bmatrix} 8 & 4 & 4 & 0 & 0 \\ 4 & 2 & 2 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

with the eigenvalues $\{11, 0, 0, 0, 0\}$. Thus, we conclude that the R-order of the methods with memory (16)–(N2) is at least *eleven*.

Method (N3), γ_k is calculated using (14):

The matrices $M_1,\ M_2$ and M_3 remain the same, and the last informational matrix is

$$M_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The matrix $M^{(N_3)}$ corresponding to the multi-point method (16)–(N3) is

$$M^{(N_3)} = M_1 M_2 M_3 M_4 = \begin{bmatrix} 8 & 4 & 4 & 4 & 0 \\ 4 & 2 & 2 & 2 & 0 \\ 2 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

with the eigenvalues $\{6 + 4\sqrt{2}, 6 - 4\sqrt{2}, 0, 0, 0\}$. Therefore, the R-order of the methods with memory (16)–(N3) is not less than $6 + 4\sqrt{2} \approx 11.66$.

Method (N4), γ_k is calculated using (15):

Again, only the last informational matrix has a different form

$$M_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The matrix $M^{(N_4)}$ corresponding to the multi-point method (16)–(N4) is

$$M^{(N_4)} = M_1 M_2 M_3 M_4 = \begin{bmatrix} 8 & 4 & 4 & 4 & 4 \\ 4 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

with the eigenvalues $\{12, 0, 0, 0, 0\}$. Hence, the R-order of the methods with memory (16)–(N4) is at least *twelve*.

In this way we have completed the analysis of all accelerating methods (N1)–(N4) so that the proof of Theorem 2 is finished. □

5. Numerical examples

We have tested the proposed families of three-point methods (2) and (16) using the programming package *Mathematica* with multiple-precision arithmetic. Apart from these families, several three-point iterative methods (IM) of optimal order eight, which also require four function evaluations, have been tested. For demonstration, we have selected five methods displayed below.

Three-point methods of Bi et al. [2]:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - g(u_k)\frac{f(y_k)}{f'(x_k)}, \\ x_{k+1} = z_k - \frac{f(x_k)+\beta f(z_k)}{f(x_k)+(\beta-2)f(z_k)} \cdot \frac{f(z_k)}{f[z_k,y_k]+f[z_k,x_k,x_k](z_k-y_k)}, \end{cases} \tag{17}$$

where $\beta \in \mathbf{R}$, $u_k = f(y_k)/f(x_k)$ and $g(u)$ is a real-valued function satisfying

$$g(0) = 1, \quad g'(0) = 2, \quad g''(0) = 10, \quad |g'''(0)| < \infty.$$

Derivative free Kung-Traub's family [7]:

$$\begin{cases} y_k = x_k - \frac{\gamma f(x_k)^2}{f(x_k+\gamma f(x_k))-f(x_k)}, \\ z_k = y_k - \frac{f(y_k)f(x_k+\gamma f(x_k))}{[f(x_k+\gamma f(x_k))-f(y_k)]f[x_k,y_k]}, \quad \gamma \in \boldsymbol{R}. \\ x_{k+1} = z_k - \frac{f(y_k)f(x_k+\gamma f(x_k))\left(y_k-x_k+\frac{f(x_k)}{f[x_k,z_k]}\right)}{[f(y_k)-f(z_k)][f(x_k+\gamma f(x_k))-f(z_k)]} + \frac{f(y_k)}{f[y_k,z_k]}. \end{cases} \tag{18}$$

Kung-Traub's family with first derivative [7]:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - \frac{f(x_k)f(y_k)}{[f(x_k)-f(y_k)]^2}\frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = z_k - \frac{f(x_k)f(y_k)f(z_k)\{f(x_k)^2+f(y_k)[f(y_k)-f(z_k)]\}}{[f(x_k)-f(y_k)]^2[f(x_k)-f(z_k)]^2[f(y_k)-f(z_k)]}\frac{f(x_k)}{f'(x_k)}. \end{cases} \tag{19}$$

Sharma-Sharma's method [14]:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - \frac{f(y_k)}{f'(x_k)} \cdot \frac{f(x_k)}{f(x_k)-2f(y_k)}, \\ x_{k+1} = z_k - \left(1+\frac{f(z_k)}{f(x_k)}\right)\frac{f(z_k)f[x_k,y_k]}{f[x_k,z_k]f[y_k,z_k]}. \end{cases} \tag{20}$$

Džunić–Petković–Petković's method [4]:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f[x_k,w_k]}, \\ z_k = y_k - h(u_k,v_k)\frac{f(y_k)}{f[x_k,w_k]}, \quad u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \\ x_{k+1} = z_k - \frac{f(z_k)}{f[z_k,y_k]+f[z_k,y_k,x_k](z_k-y_k)+f[z_k,y_k,x_k,w_k](z_k-y_k)(z_k-x_k)}. \end{cases} \tag{21}$$

The errors $|x_k - \alpha|$ of approximations to the zeros, produced by (2), (17)–(21), are given in Tables 1 and 2, where $A(-h)$ denotes $A \times 10^{-h}$. These tables include the values of the computational order of convergence r_c calculated by the formula [10]

$$r_c = \frac{\log|f(x_k)/f(x_{k-1})|}{\log|f(x_{k-1})/f(x_{k-2})|}, \tag{22}$$

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

$$f(x) = e^{x^2+x\cos x-1}\sin \pi x + x\log(x\sin x+1), \quad \alpha = 0,\ x_0 = 0.6,$$
$$f(x) = \log(x^2-2x+2) + e^{x^2-5x+4}\sin(x-1), \quad \alpha = 1,\ x_0 = 1.35.$$

From Tables 1 and 2 and many tested examples we can conclude that all implemented methods converge very fast and generate results of approximately same accuracy. From the last column of Tables 1 and 2 we notice that the computational order of convergence r_c, calculated by (22), matches very well the theoretical order.

Applying the family (16) to the same functions as above, we observe considerable increase of the accuracy of approximations produced by the methods with memory. The quality of the approaches in calculating γ_k by (12)–(15) can be noticed

Table 1
Three-point methods without memory $f(x) = e^{x^2+x\cos x-1}\sin \pi x + x\log(x\sin x+1),\ \alpha = 0,\ x_0 = 0.6.$

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	r_c (22)
(17) $g(u) = 1+\frac{4u}{2-5u}$	0.166(−2)	0.221(−21)	0.221(−172)	7.999
(18) $\gamma = 0.01$	0.126(−2)	0.370(−23)	0.198(−187)	8.000
(19)	0.114(−2)	0.152(−23)	0.154(−190)	8.000
(20)	0.136(−2)	0.279(−23)	0.876(−189)	7.999
(21) $h(u,v) = u+1/(1-v),\ \gamma = -0.1$	0.645(−4)	0.127(−32)	0.284(−262)	8.000
(2) $H_1 - G_1,\ \gamma = -1$	0.212(−3)	0.953(−33)	0.161(−267)	8.000
(2) $H_1 - G_2$	0.318(−3)	0.180(−31)	0.195(−257)	8.000
(2) $H_2 - G_3$	0.271(−2)	0.804(−23)	0.438(−187)	8.001
(2) $H_2 - G_4$	0.257(−3)	0.875(−32)	0.160(−259)	8.000
(2) $H_3 - G_5$	0.609(−3)	0.370(−28)	0.680(−230)	8.000
(2) $H_3 - G_6$	0.155(−2)	0.748(−25)	0.211(−203)	8.000
(2) $H_4 - G_7$	0.636(−3)	0.177(−28)	0.654(−233)	8.000
(2) $H_4 - G_8$	0.276(−3)	0.958(−32)	0.206(−259)	8.000
(2) $H_5 - G_9$	0.313(−3)	0.288(−31)	0.151(−255)	8.000
(2) $H_5 - G_{10}$	0.186(−3)	0.200(−32)	0.366(−264)	8.000

Table 2
Three-point methods without memory $f(x) = \log(x^2 - 2x + 2) + e^{x^2-5x+4}\sin(x-1),\ \alpha = 1,\ x_0 = 1.35$.

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	r_c (22)
(17), $g(u) = 1 + \frac{4u}{2-5u}$	0.570(−4)	0.898(−31)	0.341(−245)	7.999
(18) $\gamma = 0.01$	0.877(−4)	0.218(−30)	0.314(−243)	7.999
(19)	0.845(−4)	0.169(−30)	0.426(−244)	7.999
(20)	0.782(−4)	0.832(−31)	0.136(−246)	7.999
(21) $h(u,v) = u + 1/(1-v),\ \gamma = -0.1$	0.499(−5)	0.291(−39)	0.385(−313)	8.000
(2) $H_1 - G_1,\ \gamma = -1.2$	0.960(−6)	0.768(−49)	0.128(−393)	8.000
(2) $H_1 - G_2$	0.113(−5)	0.241(−48)	0.101(−389)	8.000
(2) $H_2 - G_3$	0.231(−5)	0.103(−45)	0.162(−368)	8.000
(2) $H_2 - G_4$	0.138(−5)	0.109(−47)	0.163(−384)	8.000
(2) $H_3 - G_5$	0.240(−5)	0.442(−46)	0.594(−372)	8.000
(2) $H_3 - G_6$	0.282(−5)	0.385(−45)	0.461(−364)	8.000
(2) $H_4 - G_7$	0.132(−5)	0.915(−48)	0.488(−385)	8.000
(2) $H_4 - G_8$	0.109(−5)	0.158(−48)	0.315(−391)	8.000
(2) $H_5 - G_9$	0.144(−5)	0.181(−47)	0.114(−382)	8.000
(2) $H_5 - G_{10}$	0.511(−5)	0.187(−43)	0.618(−351)	8.000

Table 3
Families of three-point methods (16) with memory $f(x) = e^{x^2+x\cos x-1}\sin\pi x + x\log(x\sin x + 1),\ \alpha = 0,\ x_0 = 0.6,\ \gamma_0 = -1$.

Methods		$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	r_c (22)
$H_1 - G_1$	N_1	0.212(−3)	0.399(−36)	0.251(−363)	10.000
	N_2	0.212(−3)	0.486(−40)	0.256(−443)	11.007
	N_3	0.212(−3)	0.125(−42)	0.182(−499)	11.645
	N_4	0.212(−3)	0.111(−41)	0.344(−501)	12.003
$H_1 - G_2$	N_1	0.318(−3)	0.545(−35)	0.110(−351)	9.970
	N_2	0.318(−3)	0.549(−39)	0.343(−431)	10.967
	N_3	0.318(−3)	0.432(−41)	0.672(−481)	11.615
	N_4	0.318(−3)	0.846(−41)	0.464(−490)	11.956
$H_2 - G_3$	N_1	0.271(−2)	0.210(−26)	0.123(−266)	9.963
	N_2	0.271(−2)	0.624(−29)	0.274(−322)	11.012
	N_3	0.271(−2)	0.422(−30)	0.249(−355)	11.695
	N_4	0.271(−2)	0.208(−29)	0.154(−354)	11.990
$H_2 - G_4$	N_1	0.257(−3)	0.181(−35)	0.217(−356)	9.981
	N_2	0.257(−3)	0.246(−39)	0.509(−435)	10.985
	N_3	0.257(−3)	0.706(−42)	0.555(−490)	11.621
	N_4	0.257(−3)	0.571(−41)	0.422(−492)	11.981
$H_3 - G_5$	N_1	0.609(−3)	0.154(−31)	0.496(−318)	10.018
	N_2	0.609(−3)	0.625(−35)	0.278(−388)	11.046
	N_3	0.609(−3)	0.110(−38)	0.995(−456)	11.668
	N_4	0.609(−3)	0.629(−36)	0.888(−433)	12.031
$H_3 - G_6$	N_1	0.155(−2)	0.265(−28)	0.118(−285)	9.987
	N_2	0.155(−2)	0.287(−31)	0.534(−348)	11.023
	N_3	0.155(−2)	0.357(−33)	0.317(−391)	11.686
	N_4	0.155(−2)	0.588(−32)	0.388(−385)	12.004
$H_4 - G_7$	N_1	0.636(−3)	0.347(−32)	0.436(−324)	9.975
	N_2	0.636(−3)	0.169(−36)	0.970(−405)	10.968
	N_3	0.636(−3)	0.208(−37)	0.259(−439)	11.654
	N_4	0.636(−3)	0.293(−39)	0.141(−475)	12.008
$H_4 - G_8$	N_1	0.276(−3)	0.144(−35)	0.205(−357)	9.970
	N_2	0.276(−3)	0.163(−39)	0.548(−437)	10.971
	N_3	0.276(−3)	0.925(−42)	0.109(−488)	11.616
	N_4	0.276(−3)	0.286(−41)	0.105(−495)	11.964
$H_5 - G_9$	N_1	0.313(−3)	0.103(−34)	0.655(−349)	9.980
	N_2	0.313(−3)	0.121(−38)	0.211(−427)	10.978
	N_3	0.313(−3)	0.541(−41)	0.105(−479)	11.618
	N_4	0.313(−3)	0.245(−40)	0.159(−484)	11.971
$H_5 - G_{10}$	N_1	0.186(−3)	0.756(−36)	0.636(−361)	10.036
	N_2	0.186(−3)	0.183(−39)	0.886(−438)	11.063
	N_3	0.186(−3)	0.112(−45)	0.484(−537)	11.638
	N_4	0.186(−3)	0.107(−40)	0.708(−490)	12.063

Table 4
Families of three-point methods (16) with memory $f(x) = \log(x^2 - 2x + 2) + e^{x^2-5x+4}\sin(x-1),\ \alpha = 1,\ x_0 = 1.35,\ \gamma_0 = -1.2$.

Methods		$\|x_1 - \alpha\|$	$\|x_2 - \alpha\|$	$\|x_3 - \alpha\|$	r_c (22)
$H_1 - G_1$	N_1	0.960(−6)	0.143(−58)	0.573(−586)	9.983
	N_2	0.960(−6)	0.700(−64)	0.313(−702)	10.980
	N_3	0.960(−6)	0.141(−67)	0.541(−788)	11.651
	N_4	0.960(−6)	0.215(−68)	0.633(−820)	11.996
$H_1 - G_2$	N_1	0.113(−5)	0.241(−58)	0.290(−581)	9.976
	N_2	0.113(−5)	0.210(−63)	0.955(−697)	10.970
	N_3	0.113(−5)	0.567(−67)	0.114(−780)	11.643
	N_4	0.113(−5)	0.105(−67)	0.125(−811)	11.992
$H_2 - G_3$	N_1	0.231(−5)	0.217(−55)	0.587(−554)	9.966
	N_2	0.231(−5)	0.134(−60)	0.953(−666)	10.955
	N_3	0.231(−5)	0.103(−63)	0.149(−742)	11.634
	N_4	0.231(−5)	0.320(−64)	0.255(−773)	12.048
$H_2 - G_4$	N_1	0.138(−5)	0.436(−57)	0.514(−571)	9.979
	N_2	0.138(−5)	0.223(−62)	0.184(−685)	10.971
	N_3	0.138(−5)	0.612(−66)	0.129(−768)	11.643
	N_4	0.138(−5)	0.114(−66)	0.354(−799)	11.992
$H_3 - G_5$	N_1	0.240(−5)	0.307(−55)	0.185(−552)	9.966
	N_2	0.240(−5)	0.192(−60)	0.506(−664)	10.955
	N_3	0.240(−5)	0.156(−63)	0.194(−740)	11.634
	N_4	0.240(−5)	0.498(−64)	0.510(−771)	12.048
$H_3 - G_6$	N_1	0.282(−5)	0.114(−54)	0.949(−547)	9.963
	N_2	0.282(−5)	0.773(−60)	0.231(−657)	10.951
	N_3	0.282(−5)	0.907(−63)	0.146(−731)	11.633
	N_4	0.282(−5)	0.331(−63)	0.381(−761)	12.048
$H_4 - G_7$	N_1	0.132(−5)	0.211(−57)	0.230(−574)	9.981
	N_2	0.132(−5)	0.114(−62)	0.431(−689)	10.977
	N_3	0.132(−5)	0.422(−66)	0.429(−771)	11.654
	N_4	0.132(−5)	0.933(−67)	0.267(−800)	11.996
$H_4 - G_8$	N_1	0.109(−5)	0.283(−58)	0.685(−583)	9.976
	N_2	0.109(−5)	0.144(−63)	0.145(−698)	10.971
	N_3	0.109(−5)	0.371(−67)	0.810(−783)	11.643
	N_4	0.109(−5)	0.666(−68)	0.549(−814)	11.992
$H_5 - G_9$	N_1	0.144(−5)	0.625(−57)	0.190(−569)	9.979
	N_2	0.144(−5)	0.322(−62)	0.106(−683)	10.971
	N_3	0.144(−5)	0.931(−66)	0.169(−766)	11.642
	N_4	0.144(−5)	0.179(−66)	0.786(−797)	11.992
$H_5 - G_{10}$	N_1	0.511(−5)	0.857(−53)	0.827(−528)	9.943
	N_2	0.511(−5)	0.878(−58)	0.221(−634)	10.928
	N_3	0.511(−5)	0.409(−60)	0.338(−700)	11.617
	N_4	0.511(−5)	0.220(−60)	0.158(−723)	11.977

from Tables 3 and 4. The computational order of convergence, given in the last column of Tables 3 and 4, is not so close to the theoretical value of order as in the case of methods without memory (see Tables 1 and 2), but it is still quite acceptable as a measure of convergence speed having in mind that methods with memory have more complex structure and the varying parameter, related to the methods without memory.

The R-order of convergence of the family (16) with memory is increased from 8 up to 12, in accordance with the quality of the applied accelerating method given by (12)–(15). The increase of convergence order is attained without any additional function evaluations, which points to a very high computational efficiency of the proposed methods with memory.

Our concluding remark is concerned with an important problem appearing in practical application of multipoint methods. As emphasized in [10], 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 [20] and Yun and Petković [21].

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Nos. 11101051 and 11071033) and the Serbian Ministry of Science under Grant 174022.

References

[1] 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.
[2] 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.
[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] J. Džunić, M.S. Petković, L.D. Petković, Three-point methods with and without memory for solving nonlinear equations, Appl. Math. Comput. 218 (2012) 4917–4927.
[5] 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.
[6] J. Herzberger, Über Matrixdarstellungen für Iterationverfahren bei nichtlinearen Gleichungen, Computing 12 (1974) 215–222.
[7] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, J. ACM 21 (1974) 643–651.
[8] B. Neta, M.S. Petković, Construction of optimal order nonlinear solvers using inverse interpolation, Appl. Math. Comput. 217 (2010) 2448–2455.
[9] J.M. Ortega, W.C. Rheiboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
[10] M.S. Petković, Remarks on "On a general class of multipoint root-finding methods of high computational efficiency", SIAM J. Numer. Math. 49 (2011) 1317–1319.
[11] M.S. Petković, J. Džunić, B. Neta, Interpolatory multipoint methods with memory for solving nonlinear equations, Appl. Math. Comput. 218 (2011) 2533–2541.
[12] M.S. Petković, J. Džunić, L.D. Petković, A family of two-point methods with memory for solving nonlinear equations, Appl. Anal. Discrete Math. 5 (2011) 298–317.
[13] 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.
[14] J.R. Sharma, R. Sharma, A new family of modified Ostrowski's methods with accelerated eighth order convergence, Numer. Algorithms 54 (2010) 445–458.
[15] 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.
[16] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
[17] X. Wang, L. Liu, New eighth-order iterative methods for solving nonlinear equations, J. Comput. Appl. Math. 234 (2010) 1611–1620.
[18] X. Wang, L. Liu, Modified Ostrowski's method with eighth-order convergence and high efficiency index, Appl. Math. Lett. 23 (2010) 549–554.
[19] Q. Zheng, J. Li, F. Huang, Optimal Steffensen-type families for solving nonlinear equations, Appl. Math. Comput. 217 (2011) 9592–9597.
[20] B.I. Yun, A non-iterative method for solving non-linear equations, Appl. Math. Comput. 198 (2008) 691–699.
[21] B.I. Yun, M.S. Petković, Iterative methods based on the signum function approach for solving nonlinear equations, Numer. Algorithms 52 (2009) 649–662.