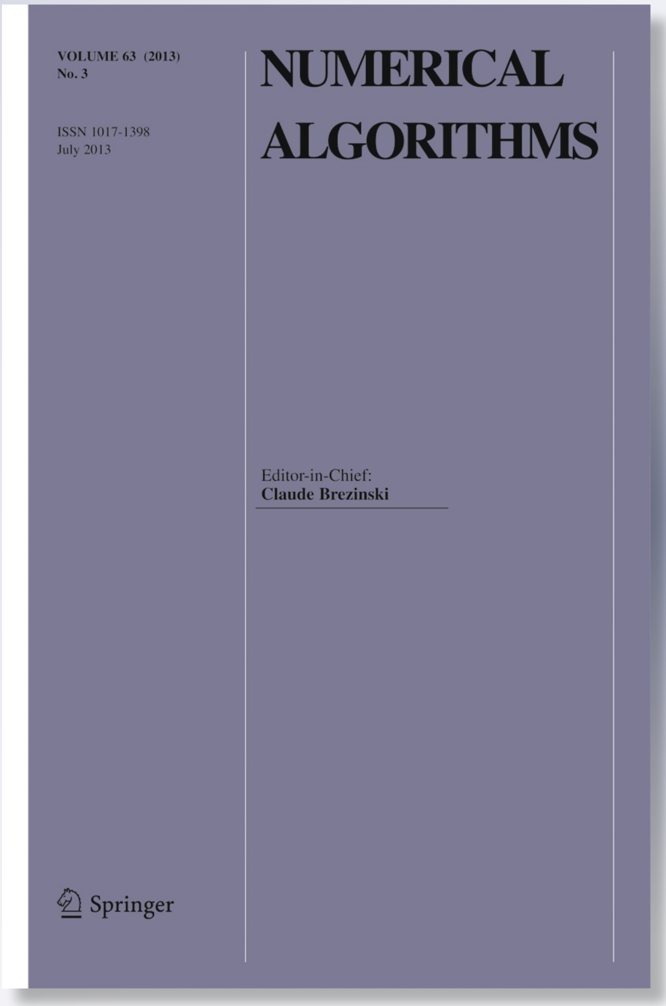
On efficient two-parameter methods for solving nonlinear equations

Jovana Džunić

Numerical Algorithms

ISSN 1017-1398
Volume 63
Number 3

Numer Algor (2013) 63:549-569
DOI 10.1007/s11075-012-9641-3



Springer

Numer Algor (2013) 63:549–569
DOI 10.1007/s11075-012-9641-3

ORIGINAL PAPER

On efficient two-parameter methods for solving nonlinear equations

Jovana Džunić

Received: 16 May 2012 / Accepted: 24 August 2012 / Published online: 7 September 2012

Abstract Derivative free methods for solving nonlinear equations of Steffensen's type are presented. Using two self-correcting parameters, calculated by Newton's interpolatory polynomials of second and third degree, the order of convergence is increased from 2 to 3.56. This method is used as a corrector for a family of biparametric two-step derivative free methods with and without memory with the accelerated convergence rate up to order 7. Significant acceleration of convergence is attained without any additional function calculations, which provides very high computational efficiency of the proposed methods. Another advantage is a convenient fact that the proposed methods do not use derivatives. Numerical examples are given to demonstrate excellent convergence behavior of the proposed methods and good coincidence with theoretical results.

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

Mathematics Subject Classification (2010) 65H05

This research was supported by the Serbian Ministry of Education and Science under grant number 174022.

J. Džunić (✉)
Faculty of Electronic Engineering, Department of Mathematics,
University of Niš, 18000 Niš, Serbia
e-mail: jovana.dzunic@elfak.ni.ac.rs

1 Introduction

In this paper we present two new iterative methods with memory of Steffensen's type for solving nonlinear equations. Our motivation for constructing these methods is directly connected with the basic concept of Numerical analysis that any numerical algorithm should give as good as possible output results with minimal computational cost. In other words, it is necessary to search for algorithms of great computational efficiency. The main goal of this paper is to present new, simple methods of considerably increased computational efficiency, higher than that of the existing methods in the considered class of methods.

A very high computational efficiency of these methods is obtained by applying a new accelerating procedure based on varying two parameters calculated by Newton's interpolating polynomials in each iteration. Considerable increase of convergence is achieved without additional function evaluations, making the proposed root solvers very efficient. Biparametric acceleration technique by self-correcting parameters was not applied in the literature at present. Numerical examples confirm excellent convergence properties of the presented methods.

In our convergence analysis of the proposed methods we employ the O- and o-notation: If $\{g_k\}$ and $\{h_k\}$ are null sequences and $g_k/h_k \to C$, where C is a nonzero constant, we write $g_k = O(h_k)$ or $g_k \sim Ch_k$. If $g_k/h_k \to 0$, we write $g_k = o(h_k)$.

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. In his book [13] Traub considered the iterative function of order two

$$\Phi(x,\gamma) = x - \frac{\gamma f(x)^2}{f(x+\gamma f(x)) - f(x)} = x - \frac{f(x)}{f\left[x, x+\gamma f(x)\right]}, \tag{1}$$

where $\gamma \neq 0$ is a real constant and $f[x,y] = \dfrac{f(x)-f(y)}{x-y}$ denotes a divided difference. Note that the choice $\gamma = 1$ reduces (1) to the well-known Steffensen iterative method [12].

Introducing the abbreviations

$$u(x) = \frac{f(x)}{f'(x)}, \quad C_2(x) = \frac{f''(x)}{2f'(x)},$$

Traub [13] derived the error relation of method (1) in the form

$$\Phi(x,\gamma) - \alpha = \left(1+\gamma f'(x)\right) C_2(x) u(x)^2 + O\left(u(x)^3\right), \tag{2}$$

and showed that the Steffensen-like method (1) can somewhat be improved by the reuse of information from the previous iteration. Approximating $f'(x)$ by the secant

$$\widetilde{f}'(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = f\left[x_k, x_{k-1}\right],$$

Traub constructed the following method with memory

$$\begin{cases} \gamma_0 \text{ is given}, \quad \gamma_k = -\dfrac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \text{ for } k \geq 1, \\ x_{k+1} = x_k - \dfrac{\gamma_k f(x_k)^2}{f(x_k + \gamma_k f(x_k)) - f(x_k)}, \end{cases} \quad (k = 0, 1, \ldots) \tag{3}$$

with the order of convergence at least $1 + \sqrt{2} \approx 2.414$.

Similar approaches for accelerating derivative free multipoint methods by varying parameters were applied in [2, 3, 9] in a considerably more efficient way. Following Traub's classification [13, pp. 8–9], methods that use information from the current and previous iteration are called *methods with memory*. This kind of methods is the subject of this paper.

In this paper we show that the iterative method (1) can be additionally accelerated without increasing computational cost, which directly improves computational efficiency of the modified method. The main idea in constructing a higher order method consists of the introduction of another parameter p and the improvement of accelerating technique for the parameter γ. The error relation of the new method, constructed in this way, gives a clear idea and motivation for further acceleration with the reuse of available information. The same idea is applied to a family of two-point Steffensen-like methods to achieve considerable improvement of computational efficiency.

2 Choice of initial approximations

Although the choice of good initial approximations is of great importance in the application of iterative methods, including multipoint methods, this task is very seldom considered in the literature. Recall that Newton-like and Steffensen-like methods of the second order have been most frequently used as predictors in the first step of multipoint methods. Both classes of these methods are of tangent type and, therefore, they are locally convergent, which means that a reasonably close initial approximation to the sought zero α should be found. Otherwise, if the chosen initial approximation is too far from the sought zero (say, if it is chosen randomly), then the applied methods, either the ones proposed in this paper or some others with local convergence developed during the last two centuries, will probably find some other (often unwanted) zero or they will diverge. Therefore, the determination of a reasonably good approximation x_0 that guarantees the convergence of the sequence of approximations $\{x_k\}_{k \in \boldsymbol{N}}$ to the zero of f is a significant task. It is interesting to note that initial approximations, chosen randomly in a suitable way, give acceptable results when simultaneous methods for finding all roots of polynomial equations are applied, e.g., employing Aberth's approach [1].

There are many methods (mainly of non-iterative nature) and strategies for finding sufficiently good initial approximations. The well-known bisection method and its modifications belong to the simplest but not always sufficiently efficient techniques. There is a vast literature on this subject so that we omit details here. We only note that complete root-finding algorithms often consist of two parts: (i) slowly convergent search algorithm to isolate distinct real or complex interval containing single root and (ii) rapidly convergent iterative method for finding sufficiently close approximation of the isolated root to the required accuracy. In this paper we are concentrating on the part (ii). Applying computer algebra systems, a typical statement for solving nonlinear equations reads `FindRoot`[*equation*, $\{x,\ x_0\}$], see, e.g., Wolfram's computational software package *Mathematica*, that is, an initial approximation x_0 is required.

In finding good initial approximations, a great advance was recently achieved by developing an efficient non-iterative method of significant practical importance, originally proposed by Yun [15] and latter discussed in [10, 16, 17]. Yun's method is based on numerical integration briefly referred to as NIM, where tanh, arctan and signum functions are involved. The NIM requires neither any knowledge of the derivative $f'(x)$ nor any iterative process. Handling non-pathological cases it is not necessary to have a close approximation to the zero; instead, a real interval (not necessarily tight) that contains the root (so-called inclusion interval) is sufficient. For illustration, to find an initial approximation x_0 of the zero $\alpha = \pi$ of the function

$$f(x) = e^{-x^2}\frac{\sin x}{x^2-1} + x^2\log(1 + x - \pi)$$

(tested in Section 6), isolated in the interval $[2.2, 10]$, we employed Yun's algorithm with the statement

```
x0=0.5*(a+b+Sign[f[a]]*NIntegrate[Tanh[m*f[x]],{x,a,b}])
```

taking $m = 3$, $a = 2.2$, $b = 10$, and found very good approximation $x_0 = 3.14147$ with $|x_0 - \alpha| \approx 1.2 \times 10^{-4}$.

Remark 1 Solving real-life problems in engineering disciplines, computer sciences, physics, biology, economics, etc., an approximate location of the wanted solution of a given nonlinear equation is most frequently known to the user. This means that there is no need for an extensive zero-searching method; the mentioned Yun's method [15] applied to the isolated inclusion interval (not necessarly tight) gives satisfactory results. Having in mind this fact and recently developed Yun's improved NIM global method [17] that finds all (simple or multiple) roots within a given interval, it could be said that numerical testing locally convergent root-finding methods with randomly selected initial approximations are rather of academic interest.

3 Improved Steffensen-like method

For $k \geq 1$ introduce the abbreviations

$$\varepsilon_k = x_k - \alpha, \quad \varepsilon_{k+1} = x_{k+1} - \alpha, \quad w_k = x_k + \gamma f(x_k), \quad \varepsilon_{k,w} = w_k - \alpha,$$

and

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

Let us consider the following modification of method (1) with an additional parameter p,

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k] + pf(w_k)} \quad (k = 0, 1, \ldots). \tag{4}$$

Using Taylor's series about a simple zero α of f, we obtain

$$f(x_k) = f'(\alpha)\Big(\varepsilon_k + c_2\varepsilon_k^2 + c_3\varepsilon_k^3 + O\left(\varepsilon_k^4\right)\Big), \tag{5}$$

$$\varepsilon_{k,w} = \left(1 + \gamma f'(\alpha)\right)\varepsilon_k + \gamma \frac{f''(\alpha)}{2}\varepsilon_k^2 + O\left(\varepsilon_k^3\right), \tag{6}$$

$$f(w_k) = f'(\alpha)\Big(\varepsilon_{k,w} + c_2\varepsilon_{k,w}^2 + c_3\varepsilon_{k,w}^3 + O\left(\varepsilon_{k,w}^4\right)\Big). \tag{7}$$

In view of (4)–(7) we find

$$\begin{aligned} f[x_k, w_k] + pf(w_k) = f'(\alpha)\Big(&1 + c_2\varepsilon_k + (c_2 + p)\,\varepsilon_{k,w} + c_3\left(\varepsilon_k^2 + \varepsilon_k\varepsilon_{k,w}\right) \\ &+ (c_3 + pc_2)\,\varepsilon_{k,w}^2 + O\left(\varepsilon_k^3\right)\Big), \end{aligned} \tag{8}$$

$$\frac{f(x_k)}{f[x_k, w_k] + pf(w_k)} = \varepsilon_k - (c_2 + p)\,\varepsilon_k\varepsilon_{k,w} + O\left(\varepsilon_k^2\varepsilon_{k,w}\right), \tag{9}$$

$$\begin{aligned} \varepsilon_{k+1} = x_{k+1} - \alpha &= \varepsilon_k - \frac{f(x_k)}{f[x_k, w_k] + pf(w_k)} \\ &= (c_2 + p)\,\varepsilon_k\varepsilon_{k,w} + O\left(\varepsilon_k^2\varepsilon_{k,w}\right) \end{aligned} \tag{10}$$

and hence

$$\varepsilon_{k+1} \sim \left(1 + \gamma f'(\alpha)\right)(c_2 + p)\varepsilon_k^2. \tag{11}$$

From relation (11) we conclude that the biparametric Steffensen-like method without memory (4) has order of convergence two, the same one as method (1).

Error relation (11) plays the key role in our study of the convergence acceleration. Method (4) serves as the basis for constructing a new biparametric derivative free family of two-point methods of the form

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f[x_k, w_k] + pf(w_k)}, & w_k = x_k + \gamma f(x_k), \\ x_{k+1} = y_k - g(t_k)\dfrac{f(y_k)}{f\left[y_k, w_k\right] + pf(w_k)}, & t_k = \dfrac{f(y_k)}{f(x_k)}. \end{cases} \tag{12}$$

A weight function g should be determined in such way that it provides the optimal order four for the two-point iterative scheme (12).

Let $\varepsilon_{k,y} = y_k - \alpha$. Taylor's expansion (5) and relations (8)–(10) lead to

$$\varepsilon_{k,y} = (c_2 + p)\varepsilon_k\varepsilon_{k,w} + (c_3 - c_2(c_2+p))\varepsilon_k^2\varepsilon_{k,w} + \left(c_3 + pc_2 - (c_2+p)^2\right)\varepsilon_k\varepsilon_{k,w}^2 + O\left(\varepsilon_k^3\varepsilon_{k,w}\right), \tag{13}$$

$$f(y_k) = f'(\alpha)\left(\varepsilon_{k,y} + c_2\varepsilon_{k,y}^2 + O\left(\varepsilon_{k,y}^3\right)\right), \tag{14}$$

$$t_k = \frac{f(y_k)}{f(x_k)} = \frac{\varepsilon_{k,y}}{\varepsilon_k}\frac{1 + c_2\varepsilon_{k,y} + O\left(\varepsilon_{k,y}^2\right)}{1 + c_2\varepsilon_k + O\left(\varepsilon_k^2\right)} = (c_2+p)\varepsilon_{k,w} + \left(c_3 - 2c_2(c_2+p)\right)\varepsilon_k\varepsilon_{k,w} + \left(c_3 + pc_2 - (c_2+p)^2\right)\varepsilon_{k,w}^2 + O(\varepsilon_k^2\varepsilon_{k,w}) \tag{15}$$

and

$$f[y_k, w_k] + pf(w_k) = f'(\alpha)\Big(1 + (c_2+p)\varepsilon_{k,w} + (c_3+pc_2)\varepsilon_{k,w}^2 + c_2\varepsilon_{k,y} + O\left(\varepsilon_{k,w}^3\right)\Big), \tag{16}$$

$$\frac{f(y_k)}{f[y_k, w_k] + pf(w_k)} = \varepsilon_{k,y} - (c_2+p)\varepsilon_{k,w}\varepsilon_{k,y} + \left((c_2+p)^2 - (c_3+pc_2)\right)\varepsilon_{k,w}^2\varepsilon_{k,y} + O\left(\varepsilon_{k,w}^3\varepsilon_{k,y}\right). \tag{17}$$

Since $t_k \to 0$ when $k \to \infty$ (see (15)), let g be represented by its Taylor's expansion about 0,

$$g(t) = g(0) + g'(0)t + \tfrac{1}{2}g''(0)t^2 + O\left(t^3\right). \tag{18}$$

After some elementary calculations, having in mind (18), we arrive at the error estimate for the new approximation x_{k+1},

$$\begin{aligned}
\varepsilon_{k+1} = x_{k+1} - \alpha &= \varepsilon_{k,y} - \left(g(0) + g'(0)t_k + \tfrac{1}{2}g''(0)t_k^2 + O\left(t_k^3\right)\right)\Big(\varepsilon_{k,y} \\
&\quad - (c_2+p)\varepsilon_{k,w}\varepsilon_{k,y} + \left((c_2+p)^2 - (c_3+pc_2)\right)\varepsilon_{k,w}^2\varepsilon_{k,y} + O\left(\varepsilon_{k,w}^3\varepsilon_{k,y}\right)\Big) \\
&= \varepsilon_{k,y}\Big(1 - g(0) + \left(g(0) - g'(0)\right)(c_2+p)\varepsilon_{k,w} - g'(0)\left(c_3 - 2c_2(c_2+p)\right)\varepsilon_k\varepsilon_{k,w} \\
&\quad + \Big((g(0) - g'(0))(c_3+pc_2) - \left(g(0) - 2g'(0) + \tfrac{1}{2}g''(0)\right)(c_2+p)^2\Big)\varepsilon_{k,w}^2\Big) \\
&\quad + O\left(\varepsilon_k^2\varepsilon_{k,w}\varepsilon_{k,y}\right).
\end{aligned} \tag{19}$$

From (19) we conclude that the choice $g(0) = g'(0) = 1$ gives the highest order to the family of two-point methods (12), with the error relation

$$\begin{aligned}\varepsilon_{k+1} &= \Big(\big(2c_2(c_2+p)-c_3\big)\varepsilon_k\varepsilon_{k,w} + \big(1-\tfrac{1}{2}g''(0)\big)(c_2+p)^2\varepsilon_{k,w}^2\Big)\varepsilon_{k,y}\\ &\quad + O\left(\varepsilon_k^2\varepsilon_{k,w}\varepsilon_{k,y}\right)\\ &= \Big(2c_2(c_2+p)-c_3+\big(1-\tfrac{1}{2}g''(0)\big)(1+\gamma f'(\alpha))(c_2+p)^2\Big)\varepsilon_k\varepsilon_{k,w}\varepsilon_{k,y}\\ &\quad + O\left(\varepsilon_k^2\varepsilon_{k,w}\varepsilon_{k,y}\right). \end{aligned} \tag{20}$$

After substituting relations (6) and (13) into (20), the error relation takes the form

$$\begin{aligned}\varepsilon_{k+1} &\sim (c_2+p)\left(1+\gamma f'(\alpha)\right)^2\\ &\quad\times\Big(2c_2(c_2+p)-c_3+\big(1-\tfrac{1}{2}g''(0)\big)(1+\gamma f'(\alpha))(c_2+p)^2\Big)\varepsilon_k^4. \end{aligned} \tag{21}$$

The above results can be summarized in the following theorem:

Theorem 1 *For a sufficiently good initial approximation x_0 of a simple zero α of the function f, the family of two-point methods* (12) *obtains the order at least four if the weight function g satisfies conditions*

$$g(0)=1, \quad g'(0)=1, \quad \left|g''(0)\right| < \infty. \tag{22}$$

Then the error relation for the family (12) *is given by* (21).

We give some examples for the weight function g with a simple form satisfying conditions (22). The first example is

$$g(t) = \left(1+\frac{t}{s}\right)^s, \quad s \neq 0.$$

In particular, for $s = 1$ we obtain $g(t) = 1 + t$. Also, the choices $s = -1$ with $g(t) = 1/(1-t)$ (giving the Kung–Traub two-point method [6] for $p = 0$) and $s = 1/2$ with $g(t) = \sqrt{1+t/2}$ are convenient for practical applications. Some other simple examples are

$$g(t) = \frac{1+\beta_1 t}{1+(\beta_1-1)t}, \quad g(t) = \frac{1+\beta_2 t^2}{1-t},$$

$$g(t) = \frac{t^2+(\beta_3-1)t-1}{\beta_3 t-1}, \quad g(t) = \frac{1}{1-t+\beta_4 t^2},$$

where $\beta_1, \beta_2, \beta_3, \beta_4 \in \boldsymbol{R}$. All of these four weight functions are only special cases of the generalized function

$$g(t) = \frac{1+at+bt^2}{1+(a-1)t+ct^2}$$

for $a, b, c \in \boldsymbol{R}$. Another general class of weight functions is given by

$$g(t) = \frac{\left(1 + at + bt^2\right)^{s_1}}{\left(1 + \dfrac{as_1 - 1}{s_2}t + ct^2\right)^{s_2}},$$

for $a, b, c \in \boldsymbol{R}$ and $s_1, s_2 \neq 0$.

4 Acceleration of the one-point method

The main idea in constructing methods with memory consists of the calculation of parameters $\gamma = \gamma_k$ and $p = p_k$ as the iteration proceeds by the formulas $\gamma_k = -1/\widetilde{f'}(\alpha)$ and $p_k = -\widetilde{c}_2$ for $k = 1, 2, \ldots$, where $\widetilde{f'}(\alpha)$ and $\widetilde{c}_2$ are approximations to $f'(\alpha)$ and c_2, respectively. In essence, in this way we minimize the factors $1 + \gamma_k f'(\alpha)$ and $c_2 + p_k$ that appear in (21). It is preferable to find approximations $\widetilde{f'}(\alpha)$ and $\widetilde{c}_2$ of a good quality, for example, $1 + \gamma_k f'(\alpha) = O(\varepsilon_{k-1}^q)$ and $c_2 + p_k = O(\varepsilon_{k-1}^q)$ for $q > 1$. It is assumed that initial estimates γ_0 and p_0 should be chosen before starting the iterative process, for example, to take γ_0 using one of the ways proposed in [13, p. 186] and set $p_0 = 0$.

Our model for the calculation of the self-accelerating parameters γ_k and p_k in each iteration is based on the approximations

$$\gamma_k = -\frac{1}{N_2'(x_k)} = -\frac{1}{\widetilde{f'}(\alpha)} \approx -\frac{1}{f'(\alpha)}, \tag{23}$$

$$p_k = -\frac{N_3''(w_k)}{2N_3'(w_k)} = -\widetilde{c}_2 \approx -c_2, \tag{24}$$

where

$$N_2(\tau) = N_2\left(\tau; x_k, w_{k-1}, x_{k-1}\right) \ \text{ and } \ N_3(\tau) = N_3\left(\tau; w_k, x_k, w_{k-1}, x_{k-1}\right)$$

are Newton's interpolating polynomials set through 3 and 4 available approximations (nodes) from the current and previous iteration. Obviously, if fewer nodes are used for the interpolating polynomials, slower acceleration is achieved.

Combining (4) with (23) and (24), we construct the following derivative free method with memory of Steffensen's type

$$\begin{cases} \gamma_0,\ p_0 \text{ are given}, \quad w_k = x_k + \gamma_k f(x_k), \\ \gamma_k = -\dfrac{1}{N_2'(x_k)}, \quad p_k = -\dfrac{N_3''(w_k)}{2N_3'(w_k)} \ \text{ for } k \geq 1, \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f[x_k, w_k] + p_k f(w_k)}, \quad (k = 0, 1, \ldots). \end{cases} \tag{25}$$

Let $\{x_k\}$ be a sequence of approximations generated by an iterative method (IM). If this sequence converges to the zero α of f with the order at least r, we will write

$$\varepsilon_{k+1} \sim D_{k,r}\varepsilon_k^r, \tag{26}$$

where $D_{k,r}$ tends to the asymptotic error constant D_r of the iterative method (IM) when $k \to \infty$. Similarly to (26), we write

$$\varepsilon_{k,w} \sim D_{k,r_1}\varepsilon_k^{r_1}. \tag{27}$$

Formally, we use the order of a sequence of approximations as the subscribe index to distinguish asymptotic error constants.

Lemma 1 *The estimates*

$$1 + \gamma_k f'(\alpha) \sim -c_3\varepsilon_{k-1,w}\varepsilon_{k-1} \;\; and \;\; c_2 + p_k \sim \frac{c_4}{c_2}\varepsilon_{k-1,w}\varepsilon_{k-1}$$

hold.

Proof We use the well known formula for the error of Newton's interpolation. If Newton's interpolating polynomial of degree $s \in \boldsymbol{N}$ is set through the nodes $t_0, t_1, \ldots, t_s$ from the interval $I = [\min\{t_0, t_1, \ldots, t_s\}, \max\{t_0, t_1, \ldots, t_s\}]$, then for some $\zeta \in I$ we have

$$f(t) - N_s(t) = \frac{f^{(s+1)}(\zeta)}{(s+1)!}\prod_{j=0}^{s}\left(t - t_j\right). \tag{28}$$

After differentiating (28) once at the point x_k (for $s = 2$) and twice at w_k (for $s = 3$), we find

$$\begin{aligned} N_2'(x_k) &= f'(x_k) - \frac{f'''(\zeta_2)}{3!}(x_k - w_{k-1})(x_k - x_{k-1}) \\ &\sim f'(\alpha)\big(1 + 2c_2\varepsilon_k - c_3\varepsilon_{k-1,w}\varepsilon_{k-1}\big), \end{aligned} \tag{29}$$

$$\begin{aligned} N_3'(w_k) &= f'(w_k) - \frac{f^{(4)}(\zeta_3)}{4!}(w_k - x_k)(w_k - w_{k-1})(w_k - x_{k-1}) \\ &\sim f'(\alpha)\big(1 + 2c_2\varepsilon_{k,w} + c_4\varepsilon_k\varepsilon_{k-1,w}\varepsilon_{k-1}\big), \end{aligned} \tag{30}$$

$$\begin{aligned} N_3''(w_k) &= f''(w_k) - \frac{2f^{(4)}(\zeta_3)}{4!}\Big[(w_k - x_k)(w_k - w_{k-1}) + (w_k - x_k)(w_k - x_{k-1}) \\ &\qquad + (w_k - w_{k-1})(w_k - x_{k-1})\Big] \\ &\sim f''(\alpha)\left(1 + \frac{3c_3}{2c_2}\varepsilon_{k,w} - \frac{c_4}{c_2}\varepsilon_{k-1,w}\varepsilon_{k-1}\right). \end{aligned} \tag{31}$$

Using (30) and (31) we obtain

$$\frac{N_3''(w_k)}{2N_3'(w_k)} \sim \frac{1}{2}\,\frac{f''(\alpha)\Big(1+\frac{3c_3}{2c_2}\varepsilon_{k,w}-\frac{c_4}{c_2}\varepsilon_{k-1,w}\varepsilon_{k-1}\Big)}{f'(\alpha)\big(1+2c_2\varepsilon_{k,w}+c_4\varepsilon_k\varepsilon_{k-1,w}\varepsilon_{k-1}\big)} \tag{32}$$

$$\sim \frac{f''(\alpha)}{2f'(\alpha)}\big(1-\frac{c_4}{c_2}\varepsilon_{k-1,w}\varepsilon_{k-1}\big), \tag{33}$$

that is,

$$c_2+p_k=\frac{f''(\alpha)}{2f'(\alpha)}-\frac{N_3''(w_k)}{2N_3'(w_k)} \sim \frac{c_4}{c_2}\varepsilon_{k-1,w}\varepsilon_{k-1}. \tag{34}$$

From (29) it follows that

$$1+\gamma_k f'(\alpha) \sim 1-\frac{1}{1-c_3\varepsilon_{k-1,w}\varepsilon_{k-1}+2c_2\varepsilon_k} \sim -c_3\varepsilon_{k-1,w}\varepsilon_{k-1}, \tag{35}$$

since $\varepsilon_k \sim (c_2+p_{k-1})\varepsilon_{k-1,w}\varepsilon_{k-1}=o(\varepsilon_{k-1,w}\varepsilon_{k-1})$, based on (10) and (34). □

Now we state the following convergence theorem:

Theorem 2 *If an initial approximation x_0 is sufficiently close to a simple zero α of f, then the order of convergence of the Steffensen-like method with memory (25) is at least $\frac{1}{2}(3+\sqrt{17}) \approx 3.56$.*

Proof By virtue of Lemma 1, from (26), (27), (35) and (34) we have

$$1+\gamma_k f'(x_k) \sim -c_3 D_{k-1,r_1}\varepsilon_{k-1}^{1+r_1}, \tag{36}$$

$$c_2+p_k \sim \frac{c_4}{c_2} D_{k-1,r_1}\varepsilon_{k-1}^{1+r_1}. \tag{37}$$

Combining (10), (26), (27), (36) and (37) yields

$$\varepsilon_{k+1} \sim (c_2+p_k)\varepsilon_{k,w}\varepsilon_k \sim \frac{c_4}{c_2} D_{k-1,r_1} D_{k,r_1}\varepsilon_{k-1}^{1+r_1}\varepsilon_k^{1+r_1} \sim A_{k+1}\varepsilon_{k-1}^{(1+r_1)(1+r)}, \tag{38}$$

where $A_{k+1}=\frac{c_4}{c_2}D_{k-1,r_1}D_{k,r_1}D_{k-1,r}^{1+r_1}$. Similarly, by (6), (27), (36) and (37),

$$\varepsilon_{k,w} \sim \big(1+\gamma_k f'(\alpha)\big)\,\varepsilon_k \sim -c_3 D_{k-1,r_1}\varepsilon_{k-1}^{1+r_1}\varepsilon_k \sim B_k\varepsilon_{k-1}^{1+r_1+r}, \tag{39}$$

where $B_k=-c_3 D_{k-1,r_1}D_{k-1,r}$. From (26) and (27) we obtain another pair of estimates

$$\varepsilon_{k+1} \sim D_{k,r}D_{k-1,r}^{r}\varepsilon_{k-1}^{r^2}, \tag{40}$$

$$\varepsilon_{k,w} \sim D_{k,r_1}D_{k-1,r}^{r_1}\varepsilon_{k-1}^{rr_1}. \tag{41}$$

Equating the exponents of the error ε_{k-1} in pairs of relations (38) and (40), and (39) and (41), we arrive at the system of equations

$$r^2 - (1+r)(1+r_1) = 0,$$

$$rr_1 - r - r_1 - 1 = 0.$$

Positive solution is given by $r = \frac{1}{2}(3+\sqrt{17}) \approx 3.56$, and it defines the order of convergence of the Steffensen-like method with memory (25). □

Remark 2 From (32) we note that an interpolating polynomial of lower degree can be used for approximating $f'(\alpha)$ in the expression for $\widetilde{c}_2$ without decreasing the convergence order, as long as it is set through the nodes w_k and x_k. In this manner, the simplest choice for p_k, which would still give maximal convergence order, is $p_k = -\widetilde{c}_2 = -N_3''(w_k)/(2f[w_k, x_k])$. However, although the described simplification is theoretically founded, numerical results have shown that this approximation gives slight drop in the accuracy of the obtained approximations for the sought root.

5 Acceleration of the family of two-point methods

An accelerating approach, similar to that used in the previous section, will be now applied for constructing two-point methods with memory. Calculation of the parameters $\gamma = \gamma_k$ and $p = p_k$ becomes more complex since more information are available per iteration. Formulas for the calculation of $\gamma = \gamma_k$ and $p = p_k$ are given by

$$\gamma_k = -\frac{1}{N_3'(x_k)} = -\frac{1}{\widetilde{f}'(\alpha)} \approx -\frac{1}{f'(\alpha)}, \tag{42}$$

$$p_k = -\frac{N_4''(w_k)}{2N_4'(w_k)} = -\widetilde{c}_2 \approx -c_2, \tag{43}$$

where

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

and

$$N_4(\tau) = N_4\,(\tau; w_k, x_k, y_{k-1}, w_{k-1}, x_{k-1})$$

are Newton's interpolating polynomials set through 4 and 5 available approximations (nodes) from the current and previous iteration.

Combining (12) with (42) and (43), we construct the following derivative free family of two-point methods with memory of Steffensen's type

$$\begin{cases} \gamma_0,\ p_0 \text{ are given}, \quad w_k = x_k + \gamma_k f(x_k), \\ \gamma_k = -\dfrac{1}{N_3'(x_k)}, \quad p_k = -\dfrac{N_4''(w_k)}{2N_4'(w_k)} \ \text{for } k \geq 1, \\ y_k = x_k - \dfrac{f(x_k)}{f[x_k, w_k] + p_k f(w_k)} \\ x_{k+1} = y_k - g(t_k)\dfrac{f(y_k)}{f[y_k, w_k] + pf(w_k)}, \quad t_k = \dfrac{f(y_k)}{f(x_k)}, \end{cases} \quad (k = 0, 1, \ldots), \tag{44}$$

where the weight function g satisfies conditions (22).

Similarly to the proof of Lemma 1, with regard to (20) the following statement can be proved:

Lemma 2 *The estimates*

$$1 + \gamma_k f'(\alpha) \sim c_4 \varepsilon_{k-1,y}\varepsilon_{k-1,w}\varepsilon_{k-1} - 2c_2\varepsilon_k \sim G_k \varepsilon_{k-1,y}\varepsilon_{k-1,w}\varepsilon_{k-1} \tag{45}$$

and

$$c_2 + p_k \sim -\frac{c_5}{c_2}\varepsilon_{k-1,y}\varepsilon_{k-1,w}\varepsilon_{k-1} \tag{46}$$

hold, where $G_k = c_4 - 2c_2 B_{k-1}$ *and*

$$B_k = 2c_2\,(c_2 + p_k) - c_3 + \left(1 - \tfrac{1}{2}g''(0)\right)\left(1 + \gamma_k f'(\alpha)\right)(c_2 + p_k)^2. \tag{47}$$

Now we state the convergence theorem for the family of two-point methods with memory (44).

Theorem 3 *If an initial approximation* x_0 *is sufficiently close to a simple zero* α *of* f, *then the order of convergence of the family of two-point methods with memory* (44) *is at least seven.*

Proof Similar to (26) and (27), let us assume that for the iterative sequence $\{y_k\}$ the following relations hold

$$\varepsilon_{k,y} \sim D_{k,r_2}\varepsilon_k^{r_2} \tag{48}$$

and

$$\varepsilon_{k,y} \sim D_{k,r_2} D_{k-1,r}^{r_2} \varepsilon_{k-1}^{rr_2}. \tag{49}$$

From (45)–(48), (26) and (27) we have

$$1 + \gamma_k f'(\alpha) \sim G_k D_{k-1,r_1} D_{k-1,r_2} \varepsilon_{k-1}^{1+r_1+r_2}, \tag{50}$$

$$c_2 + p_k \sim -\frac{c_5}{c_2} D_{k-1,r_1} D_{k-1,r_2} \varepsilon_{k-1}^{1+r_1+r_2}. \tag{51}$$

Combining (13), (20), (27) and (45)–(50), we find

$$\varepsilon_{k+1} \sim B_k \varepsilon_k \varepsilon_{k,w} \varepsilon_{k,y} \sim B_k \varepsilon_k D_{k,r_1} \varepsilon_k^{r_1} D_{k,r_2} \varepsilon_k^{r_2} \sim A_{k+1} \varepsilon_k^{1+r_1+r_2}, \tag{52}$$

$$\varepsilon_{k,y} \sim (c_2 + p_k)\varepsilon_k \varepsilon_{k,w} \sim -\frac{c_5}{c_2} D_{k-1,r_1} D_{k-1,r_2} D_{k,r_1} D_{k-1,r}^{1+r_1} \varepsilon_{k-1}^{(1+r_1)(1+r)+r_2}, \tag{53}$$

$$\varepsilon_{k,w} \sim (1 + \gamma_k f'(\alpha))\varepsilon_k \sim G_k D_{k-1,r_1} D_{k-1,r_2} D_{k-1,r} \varepsilon_{k-1}^{1+r_1+r_2+r}, \tag{54}$$

where B_k is given in (47) and $A_{k+1} = B_k D_{k,r_1} D_{k,r_2}$.

Equating the exponents of the error ε_k in pair of relations (52) and (26) and the error ε_{k-1} in pairs of relations (54) and (41), and (53) and (49), we arrive at the system of equations

$$\begin{aligned} &r - r_1 - r_2 - 1 = 0, \\ &rr_1 - r - r_1 - r_2 - 1 = 0, \\ &rr_2 - (1 + r_1)(1 + r) + r_2 = 0. \end{aligned}$$

Positive solution is given by $r = 7$ so that convergence order of the family (44) of two-point methods with memory is seven, which concludes the proof. □

Remark 3 The simplest choice for p_k that still gives order 7 for the family with memory (44) is $p_k = -N_4''(w_k)/(2f[w_k, x_k])$, see Remark 2.

Remark 4 The other choice of nodes (of worse quality) gives approximations for γ_k and p_k of somewhat less accuracy, so that the corresponding families of the form (44) have the convergence order greater than 4 but less than 7. In this paper we are concentrating on the choice of as good as possible available nodes to obtain the maximal order 7.

6 Numerical results

The presentation of numerical results in this section serves to point to very high computational efficiency and, of less importance, to demonstrate fast convergence of the proposed methods. It is clear that a very fast convergence, a property of many existing methods, is not of particular advantage if the considered method is too expensive from a computational point of view. This simple fact has been often neglected in many papers. In this section the convergence rate is expressed by the computational convergence order r_c calculated by (57) and given in the last column of displayed tables. Computational efficiency can be successfully measured by the *efficiency index*. For an iterative method (IM) with convergence order r that requires θ function evaluations, the efficiency index (also named computational efficiency) is calculated by Ostrowski–Traub's formula

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

(see [7, p. 20], [13, Appendix C]). According to the last formula we find

$$E(25) = \left(\frac{1}{2}(3+\sqrt{17})\right)^{1/2} \approx 1.887, \text{ and } E(44) = 7^{1/3} \approx 1.913,$$

whereas computational efficiency of the most efficient three-point schemes (IM) is only $E(IM) = 8^{1/4} \approx 1.682$. Obviously, the obtained efficiency indices of the proposed methods are remarkably high.

As mentioned in Section 2, applying any root solver with local convergence, a special attention must be paid to the choice of initial approximations. If initial values are sufficiently close to the sought roots, then the expected (theoretical) convergence speed is obtainable in practice; otherwise, all iterative root-finding methods work with great difficulties and show slower convergence, especially at the beginning of the iterative process. In our numerical examples we have chosen initial approximations to real zeros of real functions using the mentioned Yun's global algorithm [17]. We have also experimented with crude approximations (see Tables 2 (part I), 5 (part I), 7 (part I) and 10 (part I)) in order to study convergence behavior of the proposed methods under modest initial conditions, and randomly chosen approximations (Tables 4, 9 and 12). In some examples different choice of initial approximations caused the convergence to different zeros, see Tables 6 and 11.

We have tested the methods (4), (12), (25) and (44) for the functions given in Table 1.

Tables 2, 3, 4, 5 and 6 contain results of iterative methods (1) and its accelerated variant (3), as well. The results for

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad \text{(Newton's method)} \tag{55}$$

of the second order and

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k) + \dfrac{f(x_k)\,f''(x_k)}{2\,f'(x_k)}} \quad \text{(Halley's method)} \tag{56}$$

with order of convergence three, are also included, for better observation of convergence acceleration obtained and an insight of impact on quality of the produced results.

The selected test-functions are not of simple form and they are certainly more complicated than most of often used Rice's test functions [11], Jenkins–Traub's test polynomials [5] and test functions used in many papers concerning nonlinear equations. For example, the function f_1 displayed in Fig. 1 shows nontrivial behavior since its graph is of ⊓-form with two relatively close zeros and a singularity close to the sought zero. The test function f_2 is a polynomial of Wilkinson's type with real zeros $1, 2, \ldots, 20$. It is well-known that this class of polynomials is ill-conditioned (a part of its "uphill-downhill" graph with very large amplitudes is displayed in Fig. 2); small perturbations of polynomial coefficients cause drastic variations of zeros. Notice that many iterative

Table 1 List of test functions

Test function		α
f_1	$(x-1)(x^6+x^{-6}+4)\sin(x^2)$	1
f_2	$\prod_{i=1}^{20}(x-i)$	2, 8, 16
f_3	$e^{-x^2}\dfrac{\sin x}{x^2-1}+x^2\log(1+x-\pi)$	π
f_4	$x+\sin x+\dfrac{1}{x}-1+2i$	$0.28860\ldots-i1.24220\ldots$
f_5	$e^{x^2-2x+3}+x+\dfrac{4}{x-1}-2+i\sqrt{2}$	$1+i\sqrt{2},\ 0.50195\ldots+i0.05818\ldots$

Table 2 One-step methods without and with memory applied to f_1

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$	r_c (57)
$\alpha=1$ $x_0=-1.5$ $\gamma_0=-0.1$ $p_0=-0.01$					
(1)	1.91 (−3)	2.31 (−6)	3.39 (−12)	7.30 (−24)	2.00
(3)	1.91 (−3)	2.71 (−6)	2.29 (−14)	2.35 (−33)	2.35
(55)			diverges		
(56)			diverges		
(4)	1.10 (−2)	7.75 (−5)	3.79 (−9)	9.04 (−18)	2.00
(25)	1.10 (−2)	5.84 (−5)	4.72 (−16)	2.25 (−54)	3.45
$\alpha=1$ $x_0=1.3$ $\gamma_0=-0.1$ $p_0=-0.1$					
(1)	1.36 (−2)	1.20 (−4)	9.13 (−9)	5.30 (−17)	2.00
(3)	1.36 (−2)	1.08 (−4)	2.69 (−10)	1.28 (−23)	2.38
(55)	1.14 (−1)	2.06 (−2)	5.90 (−4)	4.48 (−7)	2.01
(56)	4.78 (−2)	1.69 (−4)	1.45 (−11)	9.20 (−33)	3.00
(4)	1.31 (−2)	1.03 (−4)	6.23 (−9)	2.27 (−17)	2.00
(25)	1.31 (−2)	2.83 (−8)	1.15 (−27)	3.52 (−95)	3.48

Table 3 One-step methods without and with memory applied to f_2

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$	r_c (57)
$\alpha=2$ $x_0=1.6$ $\gamma_0=-0.01$ $p_0=-5$					
(1)	diverges				
(3)	4.00 (−1)	6.56 (−2)	4.63 (−3)	7.68 (−6)	2.29
(55)	1.39 (−1)	3.18 (−2)	2.27 (−3)	1.28 (−5)	1.91
(56)	8.95 (−2)	2.25 (−3)	4.97 (−8)	5.42 (−22)	3.00
(4)	diverges				
(25)	4.00 (−1)	1.86 (−1)	9.15 (−4)	9.37 (−11)	2.81
$\alpha=8$ $x_0=8.4$ $\gamma_0=-0.01$ $p_0=-5$					
(1)	diverges				
(3)	converges to 7				
(55)	converges to 7				
(56)	2.92 (−1)	8.94 (−2)	1.42 (−3)	4.78 (−9)	3.08
(4)	diverges				
(25)	4.00 (−1)	1.16 (−1)	2.88 (−2)	2.54 (−7)	8.19
$\alpha=16$ $x_0=16.4$ $\gamma_0=-0.01$ $p_0=-5$					
(1)	diverges				
(3)	4.00 (−1)	4.77 (−3)	6.10 (−6)	2.72 (−13)	2.54
(55)	2.13 (−2)	6.04 (−4)	4.49 (−7)	2.49 (−13)	2.00
(56)	1.08 (−1)	2.17 (−3)	2.31 (−8)	2.80 (−23)	3.00
(4)	diverges				
(25)	4.00 (−1)	4.77 (−3)	6.10 (−6)	2.72 (−13)	2.54

Table 4 One-step methods without and with memory applied to f_3

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	$\lvert x_4-\alpha\rvert$	r_c (57)
$\alpha=\pi$	$x_0=6$	$\gamma_0=-0.05$	$p_0=-0.05$		
(1)	1.78 (−1)	2.44 (−3)	4.12 (−7)	1.18 (−14)	2.00
(3)	1.78 (−1)	2.06 (−3)	1.56 (−8)	9.37 (−21)	2.39
(55)	9.55 (−1)	1.56 (−1)	3.86 (−3)	2.05 (−6)	2.03
(56)	3.45 (−1)	8.91 (−4)	6.92 (−11)	3.24 (−32)	3.00
(4)	1.44 (−1)	1.08 (−3)	5.09 (−8)	1.14 (−16)	2.00
(25)	1.44 (−1)	8.90 (−7)	1.79 (−23)	6.27 (−83)	3.56
$\alpha=\pi$	$x_0=7$	$\gamma_0=-0.05$	$p_0=-0.05$		
(1)	7.29 (−3)	3.65 (−6)	9.21 (−13)	5.88 (−26)	2.00
(3)	7.29 (−3)	3.66 (−6)	1.81 (−15)	2.24 (−37)	2.35
(55)	1.45 (0)	3.29 (−1)	1.86 (−2)	4.87 (−5)	2.04
(56)	6.29 (−1)	8.21 (−4)	5.39 (−11)	1.53 (−32)	3.00
(4)	5.92 (−3)	1.52 (−6)	1.02 (−13)	4.57 (−28)	2.00
(25)	5.92 (−3)	1.13 (−11)	1.70 (−40)	8.55 (−144)	3.58
$\alpha=\pi$	$x_0=9$	$\gamma_0=-0.02$	$p_0=-0.08$		
(1)	1.45 (0)	2.51 (−1)	8.32 (−3)	7.67 (−6)	2.03
(3)	1.45 (0)	2.01 (−1)	1.55 (−3)	1.00 (−8)	2.44
(55)	2.50 (0)	7.84 (−1)	1.07 (−1)	1.78 (−3)	1.95
(56)	1.28 (0)	4.05 (−2)	5.60 (−6)	1.71 (−17)	2.98
(4)	9.43 (−1)	7.62 (−2)	3.24 (−4)	4.77 (−9)	2.03
(25)	9.43 (−1)	3.61 (−3)	4.96 (−10)	2.54 (−35)	3.69

methods encounter serious difficulties in finding the zeros of Wilkinson-like polynomials.

Complex test functions f_4 and f_5 are used to show that the proposed methods are applicable to the complex domain too. As noted by Geum and Kim in [4], functions of the form like f_4 and f_5 arise from the real-life problems

Table 5 One-step methods without and with memory applied to f_4

Methods	$\lvert x_1-\alpha\rvert$	$\lvert x_2-\alpha\rvert$	$\lvert x_3-\alpha\rvert$	$\lvert x_4-\alpha\rvert$	r_c (57)
$\alpha\approx 0.29-i1.24$	$x_0=-1-3i$	$\gamma_0=-0.2$	$p_0=0.2$		
(1)	5.87 (−1)	3.09 (−2)	6.80 (−5)	3.16 (−10)	2.01
(3)	5.87 (−1)	5.35 (−2)	9.77 (−5)	2.26 (−11)	2.42
(55)	1.29 (0)	4.95 (−1)	1.95 (−2)	7.51 (−5)	1.70
(56)	5.51 (−1)	6.90 (−2)	7.07 (−5)	7.15 (−14)	3.02
(4)	6.31 (−1)	2.54 (−2)	2.85 (−5)	3.50 (−11)	2.00
(25)	6.31 (−1)	2.69 (−3)	1.93 (−11)	1.63 (−39)	3.45
$\alpha\approx 0.29-i1.24$	$x_0=-i/2$	$\gamma_0=-0.02$	$p_0=0.2$		
(1)	3.36 (−2)	7.66 (−5)	4.01 (−10)	1.10 (−20)	2.00
(3)	3.36 (−2)	4.19 (−5)	2.48 (−12)	1.09 (−29)	2.40
(55)	2.85 (−1)	1.37 (−2)	3.92 (−5)	3.17 (−10)	2.00
(56)	5.67 (−1)	3.27 (−2)	6.71 (−6)	6.13 (−17)	3.00
(4)	2.47 (−2)	2.51 (−5)	2.71 (−11)	3.16 (−23)	2.00
(25)	2.47 (−2)	1.30 (−6)	2.40 (−23)	1.93 (−69)	3.50

Table 6 One-step methods without and with memory applied to f_5

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$	r_c (57)
$\alpha=1+i\sqrt{2}$,	$x_0=i$	$\gamma_0=-0.1$	$p_0=0.2$		
(1)	2.26 (−1)	2.99 (−2)	5.61 (−4)	1.91 (−7)	2.01
(3)	2.26 (−1)	2.57 (−2)	9.84 (−5)	1.63 (−10)	2.40
(55)	6.39 (−1)	2.17 (−1)	3.26 (−2)	8.65 (−4)	1.84
(56)	2.22 (−1)	3.82 (−3)	2.56 (−8)	7.76 (−24)	3.00
(4)	2.16 (−1)	2.66 (−2)	4.09 (−4)	9.53 (−8)	2.00
(25)	2.16 (−1)	1.99 (−3)	5.89 (−12)	3.44 (−41)	3.43
$\alpha\approx 0.502+i0.058$,	$x_0=0$	$\gamma_0=-0.01$	$p_0=-1$		
(1)	2.37 (−1)	2.57 (−2)	5.94 (−4)	3.73 (−7)	1.97
(3)	2.37 (−1)	4.45 (−3)	1.49 (−6)	7.21 (−15)	2.39
(55)	1.77 (−1)	8.12 (−3)	5.60 (−5)	2.67 (−9)	2.00
(56)	7.09 (−2)	1.02 (−3)	2.82 (−9)	5.93 (−26)	3.00
(4)	3.15 (−1)	1.27 (−1)	1.65 (−2)	1.96 (−4)	2.20
(25)	3.15 (−1)	3.23 (−3)	4.59 (−10)	2.74 (−32)	3.25

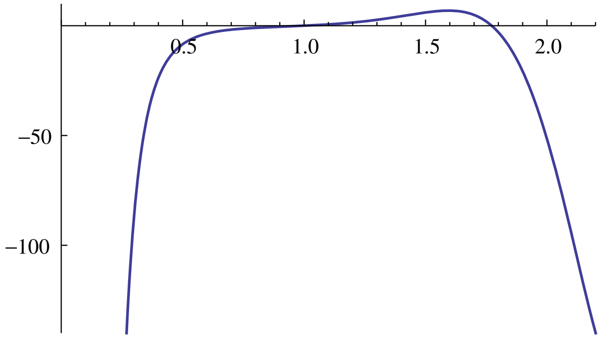


Fig. 1 The graph of $f_1(x)=(x-1)(x^6+x^{-6}+4)\sin x^2$

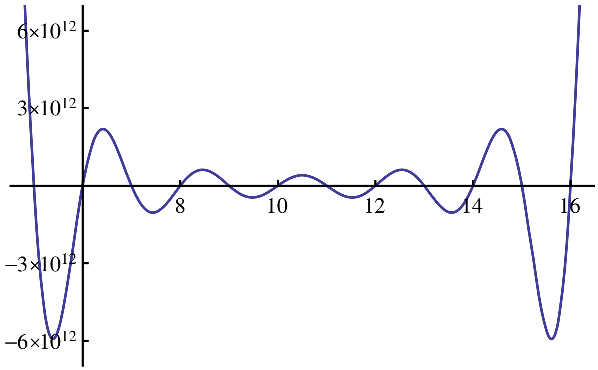


Fig. 2 The graph of $f_2(x)=\prod_{i=1}^{20}(x-i)$ on the interval [5, 17]

Table 7 Two-step methods without and with memory applied to f_1

Methods	$g(t)$	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	r_c (57)
$\alpha=1$	$x_0=-1.5$ $\gamma_0=-0.1$ $p_0=-0.01$				
(12)	$1+t$	6.36 (−3)	1.94 (−10)	2.48 (−40)	3.97
(44)	$1+t$	6.36 (−3)	1.47 (−15)	9.48 (−103)	6.90
(12)	$1/(1-t)$	6.36 (−3)	6.15 (−10)	6.13 (−38)	3.99
(44)	$1/(1-t)$	6.36 (−3)	1.47 (−15)	9.48 (−103)	6.90
$\alpha=1$	$x_0=1.3$ $\gamma_0=-0.1$ $p_0=-0.1$				
(12)	$1+t$	2.14 (−4)	5.45 (−16)	2.31 (−62)	4.00
(44)	$1+t$	2.14 (−4)	2.50 (−25)	3.98 (−171)	6.96
(12)	$1/(1-t)$	2.06 (−4)	8.29 (−16)	2.19 (−61)	4.00
(44)	$1/(1-t)$	2.06 (−4)	1.80 (−25)	4.08 (−172)	6.96

Table 8 Two-step methods without and with memory applied to f_2

Methods	$g(t)$	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	r_c (57)
$\alpha=2$	$x_0=1.6$ $\gamma_0=-0.01$ $p_0=-5$				
(12)	$1+t$	diverges			
(44)	$1+t$	4.00 (−1)	1.01 (−1)	4.72 (−7)	6.55
(12)	$1/(1-t)$	1.39 (−1)	3.18 (−2)	2.27 (−3)	1.58
(44)	$1/(1-t)$	1.39 (−1)	3.18 (−2)	5.03 (−10)	10.50
$\alpha=8$	$x_0=8.4$ $\gamma_0=-0.01$ $p_0=-5$				
(12)	$1+t$	diverges			
(44)	$1+t$	4.00 (−1)	8.72 (−2)	1.43 (−9)	17.52
(12)	$1/(1-t)$	converges to 7			
(44)	$1/(1-t)$	converges to 7			
$\alpha=16$	$x_0=16.4$ $\gamma_0=-0.01$ $p_0=-5$				
(12)	$1+t$	diverges			
(44)	$1+t$	4.00 (−1)	5.93 (−3)	1.40 (−16)	7.06
(12)	$1/(1-t)$	2.13 (−2)	6.04 (−4)	4.49 (−7)	2.04
(44)	$1/(1-t)$	2.13 (−2)	6.04 (−4)	2.55 (−23)	1.26

Table 9 Two-step methods without and with memory applied to f_3

Methods	$g(t)$	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	r_c (57)
$\alpha=\pi$	$x_0=6$ $\gamma_0=-0.05$ $p_0=-0.05$				
(12)	$1+t$	3.48 (−3)	2.90 (−13)	1.39 (−53)	4.00
(44)	$1+t$	3.48 (−3)	2.33 (−19)	2.61 (−132)	6.98
(12)	$1/(1-t)$	3.36 (−3)	2.61 (−13)	9.62 (−54)	4.00
(44)	$1/(1-t)$	3.36 (−3)	2.06 (−19)	1.10 (−132)	6.99
$\alpha=\pi$	$x_0=7$ $\gamma_0=-0.05$ $p_0=-0.05$				
(12)	$1+t$	2.70 (−6)	1.05 (−25)	2.42 (−103)	4.00
(44)	$1+t$	2.70 (−6)	1.54 (−39)	1.48 (−273)	7.04
(12)	$1/(1-t)$	2.70 (−6)	1.10 (−25)	3.04 (−103)	4.00
(44)	$1/(1-t)$	2.70 (−6)	1.55 (−39)	1.53 (−273)	7.04
$\alpha=\pi$	$x_0=9$ $\gamma_0=-0.02$ $p_0=-0.08$				
(12)	$1+t$	1.81 (−1)	3.38 (−6)	4.70 (−25)	3.98
(44)	$1+t$	1.81 (−1)	6.48 (−11)	2.79 (−73)	6.59
(12)	$1/(1-t)$	1.77 (−1)	3.39 (−6)	4.88 (−25)	3.98
(44)	$1/(1-t)$	1.77 (−1)	3.76 (−11)	6.14 (−75)	6.59

Table 10 Two-step methods without and with memory applied to f_4

Methods	$g(t)$	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	r_c (57)
$\alpha \approx 0.29 - i1.24$	$x_0 = -1-3i$	$\gamma_0 = -0.2$	$p_0 = 0.2$		
(12)	$1+t$	7.41 (−2)	6.62 (−8)	4.08 (−32)	4.00
(44)	$1+t$	7.41 (−2)	1.76 (−10)	1.06 (−70)	6.98
(12)	$1/(1-t)$	9.10 (−2)	1.56 (−7)	1.30 (−30)	4.00
(44)	$1/(1-t)$	9.10 (−2)	3.63 (−10)	1.65 (−68)	6.90
$\alpha \approx 0.29 - i1.24$	$x_0 = -i/2$	$\gamma_0 = -0.02$	$p_0 = 0.2$		
(12)	$1+t$	1.01 (−3)	2.24 (−15)	5.32 (−62)	4.00
(44)	$1+t$	1.01 (−3)	1.37 (−22)	2.08 (−155)	7.04
(12)	$1/(1-t)$	1.02 (−3)	2.43 (−15)	7.65 (−62)	4.00
(44)	$1/(1-t)$	1.02 (−3)	1.43 (−22)	2.83 (−155)	7.04

Table 11 Two-step methods without and with memory applied to f_5

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	$\|x_4-\alpha\|$	r_c (57)
$\alpha = 1+i\sqrt{2}$,	$x_0 = i$	$\gamma_0 = -0.1$	$p_0 = 0.2$		
(12)	$1+t$	5.10 (−2)	4.07 (−6)	1.51 (−22)	4.01
(44)	$1+t$	5.10 (−2)	3.23 (−10)	1.43 (−67)	7.00
(12)	$1/(1-t)$	4.91 (−2)	2.60 (−6)	1.84 (−23)	4.01
(44)	$1/(1-t)$	4.91 (−2)	2.68 (−10)	3.85 (−68)	7.00
$\alpha \approx 0.502 + i0.058$,	$x_0 = 0$	$\gamma_0 = -0.01$	$p_0 = -1$		
(12)	$1+t$	1.34 (−1)	7.13 (−4)	8.99 (−13)	3.95
(44)	$1+t$	1.34 (−1)	9.60 (−8)	7.72 (−49)	6.71
(12)	$1/(1-t)$	6.19 (−2)	5.04 (−5)	2.10 (−17)	4.02
(44)	$1/(1-t)$	6.19 (−2)	8.17 (−9)	3.52 (−56)	6.90

Table 12 Methods without and with memory applied to f_2 for random initial values and $g(t) = 1+t$

Methods	$\|x_1-\alpha\|$	$\|x_2-\alpha\|$	$\|x_3-\alpha\|$	r_c (57)	α
$x_0 = 3.5$					
(12), $\gamma_0 = -10^{-2}$, $p_0 = 0.1$	diverges				
(44), $\gamma_0 = -10^{-2}$, $p_0 = 0.1$	5.00 (−1)	3.71 (−1)	1.05 (−5)	3.55	4
(55)	7.80 (−2)	1.25 (−2)	2.33 (−4)	2.39	4
(56)	2.23 (−1)	1.72 (−2)	1.28 (−5)	2.58	4
$x_0 = 4.5$					
(12), $\gamma_0 = -10^{-2}$, $p_0 = 0.1$	diverges				
(44), $\gamma_0 = -10^{-2}$, $p_0 = 0.1$	1.50 (0)	1.16 (−2)	1.68 (−13)	4.94	4
(55)	2.23 (−1)	1.68 (−1)	1.81 (−2)	0.99	5
(56)	2.59 (−1)	2.57 (−2)	3.58 (−5)	2.65	5
$x_0 = 5.5$					
(12), $\gamma_0 = -10^{-2}$, $p_0 = 0.1$	diverges				
(44), $\gamma_0 = -10^{-2}$, $p_0 = 0.1$	4.50 (0)	2.58 (−1)	5.91 (−5)	2.88	10
(55)	*div.*				
(56)	2.97 (−1)	4.15 (−2)	1.30 (−4)	2.79	6
$x_0 = 6.5$					
(12), $\gamma_0 = -10^{-2}$, $p_0 = 0.1$	diverges				
(44), $\gamma_0 = -10^{-2}$, $p_0 = 0.1$	4.50 (0)	2.73 (−1)	1.72 (−4)	3.76	11
(55)	3.18 (−1)	6.71 (−2)	3.39 (−3)	1.84	8
(56)	3.36 (−1)	7.09 (−2)	5.73 (−4)	3.08	7

related to steady-state heat flow, electrostatic potential and fluid flow. For example, two-dimensional system

$$u(x,y)=2+y+\frac{x}{x^2+y^2}+\cos x\sinh y,\quad v(x,y)=-1+x+\frac{y}{x^2+y^2}+\sin x\cosh y,$$

related to complex potentials u and v, can be transformed to the analytic function $f_4(z)=u(x,y)+iv(x,y)=z+\sin z+\frac{i}{z}-1+2i$ $(z=x+iy)$ whose zeros can be determined by the proposed methods.

We have used the computational software package *Mathematica* with multiple-precision arithmetic. The errors $|x_k-\alpha|$ of approximations to the zeros are given in Tables 2–12, where $A(-h)$ denotes $A\times 10^{-h}$. These tables include the values of the computational order of convergence r_c calculated by the formula [8]

$$r_c=\frac{\log|f(x_k)/f(x_{k-1})|}{\log|f(x_{k-1})/f(x_{k-2})|}, \tag{57}$$

taking into consideration the last three approximations in the iterative process.

It is evident from Tables 2–12 that approximations to the roots possess great accuracy when the proposed methods with memory are applied. Results of the third iteration in Tables 7, 8, 9, 10, 11 and 12 and fourth iteration in Tables 2–6 are given only for demonstration of convergence speed of the tested methods and in most cases they are not required for practical problems at present. From Tables 3, 8 and 12 we observe that all tested methods work with efforts (even the basic fourth-order method (11) diverges in three cases) when they are applied to the polynomial f_2 of Wilkinson's type [14], which is, actually, a well-known fact from practice. We also notice that the values of the computational order of convergence r_c somewhat differ from the theoretical ones for the methods with memory. This fact is not strange having in mind that formula (57) for r_c is derived for the methods without memory when r_c usually fits very well the theoretical order.

Acknowledgement I would like to express my gratitude to anonymous referee for helpful suggestions and valuable remarks.

References

1. Aberth, O.: Iteration methods for finding all zeros of a polynomial simultaneously. Math. Comput. **27**, 339–344 (1973)
2. Džunić, J., Petković, M.S.: On generalized multipoint root-solvers with memory. J. Comput. Appl. Math. **236**, 2909–2920 (2012)
3. Džunić, J., Petković, M.S., Petković, L.D.: Three-point methods with and without memory for solving nonlinear equations. Appl. Math. Comput. **218**, 4917–4927 (2012)
4. Geum, Y.H., Kim, Y.I.: A multi-parameter family of three-step eighth-order iterative methods locating a simple root. Appl. Math. Comput. **215**, 3375–3382 (2010)
5. Jenkins, M.A., Traub, J.F.: Principles of testing polynomial zero-finding programs. ACM Trans. Math. Softw. **34**, 1–26 (1975)
6. Kung, H.T., Traub, J.F.: Optimal order of one-point and multipoint iteration. J. ACM **21**, 643–651 (1974)

7. Ostrowski, A.M.: Solution of Equations and Systems of Equations. Academic Press, New York (1960)
8. Petković, M.S.: Remarks on "On a general class of multipoint root-finding methods of high computational efficiency". SIAM J. Numer. Anal. **49**, 1317–1319 (2011)
9. Petković, M.S., Džunić, J., Petković, L.D.: A family of two-point methods with memory for solving nonlinear equations. Appl. Anal. Discrete Math. **5**, 298–317 (2011)
10. Petković, M.S., Yun, B.I.: Sigmoid-like functions and root finding methods. Appl. Math. Comput. **204**, 784–793 (2008)
11. Rice, J.R.: A set of 74 test functions for nonlinear equation solvers. Computer Science Technical Reports 69-034, paper 269 (1969)
12. Steffensen, I.F.: Remarks on iteration. Skand. Aktuarietidskr. **16**, 64–72 (1933)
13. Traub, J.F.: Iterative Methods for the Solution of Equations. Prentice-Hall, Englewood Cliffs (1964)
14. Wilkinson, J.H.: Rounding Errors in Algebraic Processes. Prentice Hall, Englewood Cliffs (1963)
15. Yun, B.I.: A non-iterative method for solving nonlinear equations. Appl. Math. Comput. **198**, 691–699 (2008)
16. Yun, B.I., Petković, M.S.: Iterative methods based on the signum function approach for solving nonlinear equations. Numer. Algorithms **52**, 649–662 (2009)
17. Yun, B.I.: Iterative methods for solving nonlinear equations with finitely any roots in an interval. J. Comput. Appl. Math. **236**, 3308–3318 (2012)