Applied Mathematics and Computation 226 (2014) 635–660

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Multipoint methods for solving nonlinear equations: A survey ☆

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

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

ARTICLE INFO

Keywords:
Nonlinear equations
Iterative methods
Multipoint methods
Computational efficiency
Convergence rate
Acceleration of convergence

ABSTRACT

Multipoint iterative methods belong to the class of the most efficient methods for solving nonlinear equations. Recent interest in the research and development of this type of methods has arisen from their capability to overcome theoretical limits of one-point methods concerning the convergence order and computational efficiency. This survey paper is a mixture of theoretical results and algorithmic aspects and it is intended as a review of the most efficient root-finding algorithms and developing techniques in a general sense. Many existing methods of great efficiency appear as special cases of presented general iterative schemes. Special attention is devoted to multipoint methods with memory that use already computed information to considerably increase convergence rate without additional computational costs. Some classical results of the 1970s which have had a great influence to the topic, often neglected or unknown to many readers, are also included not only as historical notes but also as genuine sources of many recent ideas. To a certain degree, the presented study follows in parallel main themes shown in the recently published book (Petković et al., 2013) [53], written by the authors of this paper.

1. Introduction

The solution of nonlinear equations and systems of nonlinear equations has been one of the most investigated topics in applied mathematics that has produced a vast literature; see, for example Ostrowski [46], Traub [63], Ortega and Rheinboldt [45], Neta [38], McNamee [37] and references therein. In this paper we are concerned with fixed point methods that generate sequences presumably convergent to the solution of a given single equation. This class of methods can be divided into one-point and multipoint schemes. The one point methods can attain high order by using higher derivatives of the function, which is expensive from a computational point of view. On the other hand, the multipoint methods are allowing the user not to throw away information that had already been computed. This approach provides the construction of very efficient root-finding methods, which explains recent increased interest in study of multipoint root-finding methods.

Any one-point iterative method for finding a simple root, such as Newton's, Halley's, Laguerre's, Euler–Cauchy's method and members of the Traub–Schröder basic sequence, which depends explicitly on f and its first $r-1$ derivatives, cannot attain an order higher than r. Therefore, the informational efficiency (see Section 2 for definition) of one-point methods, expressed as the ratio of the order of convergence and the number of required function evaluations per iteration, cannot exceed 1. Multipoint methods are of great practical importance, since they overcome the theoretical limits of any one-point method concerning the convergence order and informational and computational efficiency. The so-called optimal n-point methods always have informational efficiency greater than 1 for $n \geqslant 2$.

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

* Corresponding author.
E-mail address: miodragpetkovic@gmail.com (M.S. Petković).

0096-3003/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2013.10.072

Traub's 1964 book [63], as well as papers published in the 1970s and in 1980s, presented several multipoint methods. A renewed interest in multipoint methods has arisen in the early years of the twenty-first century due to the rapid development of digital computers, advanced computer arithmetics (multi-precision arithmetic and interval arithmetic) and symbolic computation. The mentioned improvements in hardware and software were ultimately indispensable since multipoint methods produce approximations of great accuracy and require complicated convergence analysis that is feasible only by symbolic computation.

During the last ten years, at least 200 multipoint methods have been published in various journals for applied and computer mathematics. However, many methods turned out to be either inefficient or slight modifications/variations of already known methods. In numerous cases "new" methods were, in fact, only rediscovered methods. For these reasons, the authors of this paper decided to make a systematic review of multipoint methods, concentrating mainly on the most efficient methods and techniques for developing multipoint methods, including procedures for their unified presentation. Historical notes are also included which point to the importance of classic results dating since 1970's. A result of our three-year-long investigation is the book *"Multipoint methods for solving nonlinear equations"* [53] published in 2013 by Elsevier/Academic Press.

This survey paper, actually a mixture of theoretical results and algorithmic aspects, is intended as a review of the most important contributions in the topic, many of which are presented in the mentioned book [53]. It also includes some new parts concerned with general techniques for designing multipoint methods as well as some old ideas that go back to 1970's, which have had a great influence on many results in the considered area.

The paper is divided into eight sections and organized as follows. In Section 2 we give classification of root-finders in the same way as done by Traub [63]. Section 3 contains some basic measures necessary for the quality estimation of iterative methods and their comparison. Some general methods for constructing multipoint root-finders by interpolation and weight functions are the subject of Section 4. A review of two-point and three-point optimal methods is given in Sections 5 and 6, respectively. They are, actually, particular examples constructed using general developing techniques given in Section 4. The necessity of higher order multipoint methods for solving real-life problems is discussed at the end of Section 6. Multipoint methods with memory, constructed by inverse interpolation using two and three initial approximations, are considered in Section 7. A special attention is paid to the proper application of Herzberger's matrix method in determining order of convergence. Finally, in Section 8 we present generalized multipoint methods with memory that use self-accelerating parameters calculated by Newton's interpolation with divided differences. Convergence analysis is more general than the one given in [15] and it is exposed here in a condensed form.

We emphasize that a large part of this paper is devoted to multipoint methods with memory since it turns out that this class of root-finders possesses the greatest computational efficiency at present. We omit numerical examples since they can be found in the corresponding references cited throughout this paper.

We hope that this survey paper, together with the book [53] by the same authors, will help readers to understand various developing techniques, the convergence behavior and computational efficiency of the various multipoint methods for solving nonlinear equations.

2. Classification of root-finders

Let f be a real single-valued function of a real variable. If $f(\alpha)=0$ then α is said to be a *zero* of f or, equivalently, a *root* of the equation $f(x)=0$. It is customary to say that α is a root or zero of an algebraic polynomial f, but just a zero if f is not a polynomial.

We give a classification of iterative methods, as presented by Traub in [63]. We will always assume that f has a certain number of continuous derivatives in the neighborhood of the zero α. We most commonly solve the equation approximately, that is, we find an approximation to the zero α by applying some iterative method starting from an initial guess x_0.

(i) Let an iterative method be of the form

$$x_{k+1}=\phi(x_k)\quad (k=0,1,2,\ldots),$$

where x_k is an approximation to the zero α and ϕ is an *iteration* function. The iterative method starts with an initial guess x_0 and at every step we use only the last known approximate. In this case, we call the method *one-point*. The function ϕ may depend on derivatives of f in order to increase the order. In fact, to get a method of order r, one has to use all derivatives up to order $r-1$, see Traub [63, Th. 5.3]. The most commonly used one-point iterative method is given by

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

known as Newton's method or Newton–Raphson's method.

(ii) Suppose that real numbers $x_{k-n},\ldots,x_{k-1},x_k$ are approximations to the zero α obtained from the current and previous iterations, and let us define the mapping

$$x_{k+1}=\phi(x_k;x_{k-1},\ldots,x_{k-n}). \tag{2}$$

The approximation x_{k+1} is calculated by ϕ using the previous $n+1$ approximations. The iteration function ϕ of the form (2) is called an *one-point iteration function with memory*. An example of iteration function with memory is the well-known secant method

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k) \quad (k = 1, 2, \ldots). \tag{3}$$

(iii) Another type of iteration functions is derived by using the expressions $w_1(x_k), w_2(x_k), \ldots, w_n(x_k)$, where x_k is the common argument. The iteration function ϕ, defined as

$$x_{k+1} = \phi(x_k, w_1(x_k), \ldots, w_n(x_k)), \tag{4}$$

is called a *multipoint iteration function without memory*. The simplest examples are Steffensen's method [60]

$$x_{k+1} = x_k - \frac{f(x_k)^2}{f(x_k + f(x_k)) - f(x_k)} \quad \text{with} \quad w_1(x_k) = x_k + f(x_k) \tag{5}$$

and Traub–Steffensen's method [63]

$$x_{k+1} = \mathcal{S}(x_k) := x_k - \frac{\gamma f(x_k)^2}{f(x_k + \gamma f(x_k)) - f(x_k)} \quad \text{with} \quad w_1(x_k) = x_k + \gamma f(x_k). \tag{6}$$

Another example is the iterative two-point cubically convergent method

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

which was presented in the paper [69]. This paper was cited in many papers although the last iterative formula was derived by Traub [63, p. 164] almost forty years earlier.

(iv) Assume that iterative function ϕ has arguments z_j, where each argument represents $n+1$ quantities $x_j, w_1(x_j), \ldots, w_n(x_j)$ $(n \geqslant 1)$. Then ϕ can be represented in the general form as

$$x_{k+1} = \phi(z_k; z_{k-1}, \ldots, z_{k-n}). \tag{7}$$

The iteration function ϕ is called a *multipoint iteration function with memory*. In each iterative step we have to preserve information of the last n approximations x_j, and for each approximation we have to calculate n expressions $w_1(x_j), \ldots, w_n(x_j)$.

In this paper we treat the cases of multipoint methods without and with memory for finding a simple zero, defined respectively by (4) and (7).

3. General preliminaries

One of the most important features of iterative methods is their convergence rate defined by the *order of convergence*. Let $\{x_k\}$ be a sequence converging to α and let $\varepsilon_k = x_k - \alpha$. If there exists a real number p and a nonzero positive constant C_p such that

$$\lim_{k \to +\infty} \frac{|\varepsilon_{k+1}|}{|\varepsilon_k|^p} = C_p,$$

then p is called the *order* of the sequence $\{x_k\}$ and C_p is the *asymptotic error constant.* Some examples show that this definition is rather restrictive, which motivated Ortega and Rheinboldt [45, Ch. 9] to introduce more general concept of Q- and R-order of convergence. However, it can be proved (see Example 9.3–4 in [45, Ch. 9]) that the Q-, R- and Traub's C-order are identical when $0 < C_p < +\infty$ exists for some $p \in [1, +\infty]$. Since the asymptotic error constant C_p always satisfies this condition for all methods considered in this paper, we will not emphasize particularly this fact in the sequel.

When testing new methods, either to check the order of convergence or to estimate how much it differs from the theoretical order in practical implementation, it is of interest to use *computational order of convergence* (COC) defined by

$$\tilde{r} = \frac{\log |(x_k - \alpha)/(x_{k-1} - \alpha)|}{\log |(x_{k-1} - \alpha)/(x_{k-2} - \alpha)|}, \tag{8}$$

where $x_{k-2},\ x_{k-1}$ and x_k are the last three successive approximations to the sought root α obtained in the iterative process $x_{k+1} = \phi(x_k)$. This old result has been rediscovered by Weerakoon and Fernando [69] although formula (8) is only of theoretical value.

The value of the zero α is unknown in practice. Using the factorization $f(x) = (x - \alpha)g(x)$ and (8), we can derive the approximate formula for COC

$$r_c = \frac{\log |f(x_k)/f(x_{k-1})|}{\log |f(x_{k-1})/f(x_{k-2})|}, \tag{9}$$

which is of much better practical importance. This formula in a more general form may be found in [24]. The calculated value r_c estimates the theoretical order of convergence well when "pathological behavior" of the iterative method (for instance, slow convergence at the beginning of the implemented iterative method, "oscillating" behavior of approximations, etc.) does not exist.

There are other measures for comparing various iterative techniques. Traub [63] introduced the informational efficiency and efficiency index, which can be expressed in terms of the order (r) of the method and the number of function- (and derivative-) evaluations (θ_f). The *informational efficiency* of an iterative method (M) is defined as

$$I(M) = \frac{r}{\theta_f}. \tag{10}$$

The *efficiency index* (or *computational efficiency*) is given by

$$E(M) = r^{1/\theta_f}, \tag{11}$$

the definition that was introduced by Ostrowski [46] several years before Traub [63].

Neta [38] has collected many algorithms and listed their efficiency. Another tool for comparison of the various algorithms is the notion of basin of attraction based on graphic (most often fractal) visualization. Stewart [61] was one of the first who carried out the comparison of several second and third-order methods using computer graphics. Amat et al. [1–3], Neta et al. [41,43,44], Scott et al. [56], Chun et al. [12], and Varona [65] have expanded on this and included a variety of algorithms of different orders of convergence for simple and multiple roots. Kalantari wrote an excellent book [25] that offers fascinating and modern perspectives into the theory and practice of iterative methods for finding polynomial roots using computer graphics. This subject is of paramount importance but it is also very voluminous so it is not considered here; instead, we refer the above-mentioned references for a profound investigation.

Remark 1. It is worth emphasizing that the maximal order of convergence is not the only goal in constructing root-finding methods and, consequently, the ultimate measure of efficiency of the designed method. Complexity of the formulae involved, often called *combinatorial cost*, makes another important parameter, which should be taken into account, see [31,64]. See Section 4 for further discussion.

4. Methods for constructing multipoint root-finders

One major goal in designing new numerical methods is to obtain a method with the best possible computational efficiency. Each memory-free iteration consists of

–new function evaluations, and
–arithmetic operations used to combine the available data.

Minimizing the total number of arithmetic operations through an iterative process which would provide the zero-approximation of the desired accuracy, would be very much dependent on the particular properties of a function f whose zero is sought. However, in most cases, function or derivative evaluations are far more expensive in terms of arithmetic operations (it may even involve subroutines), than any combinatory cost of the available data. Regarding the definition (10) or (11), this means that it is desirable to achieve as high as possible convergence order with the fixed number of function evaluations per iteration. Nevertheless, working with weight functions (see Section 4.2), it is preferable to avoid complicated forms (or combinations of weight functions) in several variables.

For example, methods (1) and (5) have been proven [31] to be of least combinatorial cost among all the methods which use two function evaluations. In the case of multipoint methods without memory this demand is related to the construction of methods with the optimal order of convergence, considered in the Kung–Traub conjecture [32] from 1974:

Kung–Traub's conjecture: *Multipoint iterative methods without memory, costing $n+1$ function evaluations per iteration, have order of convergence at most 2^n.*

This conjecture was proved for some classes of multipoint methods by Woźniakowski in [72].

Multipoint methods that satisfy the Kung–Traub conjecture are usually called *optimal methods* (see [31,32]) and, naturally, they are of particular interest. Consequently, the *optimal order* is $r = 2^n$ so that the *optimal efficiency index* is

$$E_n^{(o)} = 2^{n/(n+1)}.$$

A class of optimal n-point methods, reaching the order 2^n with $n+1$ function evaluations per iteration, will be denoted by Ψ_{2^n} $(n \geqslant 1)$. The Kung–Traub conjecture is supported by the families of multipoint methods of arbitrary order n, proposed in [32,49,73], and also by a number of particular multipoint methods developed after 1960.

Let us consider the case of root-finding iterative methods without memory for simple roots that use Hermitian type of information (H-information for short). This type of information implies that if we use the derivative $f^{(d)}(y)$ at a certain point y, then all $f^{(j)}(y),\ 0 \leqslant j \leqslant d$, are used as well. Most of the developed iterative root-solvers are based on H-information. The first optimal methods that differ from this class (do not use Hermitian type of information) are Jarratt's families of two-point methods, see [22,23]. The required information for these families usually called general (sparse) Hermite information, or Hermite–Birkhoff type of information, and are in close relation to Hermite–Birkhoff interpolation, often called general Hermite interpolation.

H-information based iterations are widely constructed and investigated in details. Woźniakowski [71] proved for iterations based on H-information $f^{(i)}(y_{k,j}),\ 0 \leqslant i \leqslant d_j,\ 0 \leqslant j \leqslant m-1$, that they have a very specific form of the error relation

$$x_{k+1} - \alpha \sim \prod_{j=1}^{m-1} (y_{k,j} - \alpha)^{r_j}, \ \text{where}\ r_j \leqslant d_j + 1. \tag{12}$$

Traub's detailed research [63] states that for the class of interpolatory iterations the equality $r_j = d_j + 1$ holds in (12). The symbol $\sim$ in (12) and later in the text means that infinitesimally small quantities g and h are of the same order in magnitude, denoted as $g \sim Ch$ or $g = O(h)$, if $g/h \to C$, where C is a nonzero constant.

In the sequel $f[x,y] = (f(x) - f(y))/(x-y)$ will denote a divided difference. Divided differences of higher order are defined recursively by the formula

$$f[x_0, x_1, \ldots, x_i] = \frac{f[x_1, \ldots, x_i] - f[x_0, \ldots, x_{i-1}]}{x_i - x_0} \quad (i > 1).$$

The assertions proved in [8,63,72] show that in the class of iterations based on H-information, interpolatory type methods reach the maximal order of convergence

$$r(\phi) = (d_0 + 1) \prod_{i=1}^{m-1} (d_i + 2)$$

and that the Kung–Traub hypothesis holds for this class of methods.

Let ϕ denote an iteration function and let $v(\phi)$ be the total number of function evaluations used to compute $\phi(f)(x)$ per iteration. Kung and Traub [32] stated the following conditions for the highest (optimal) informational efficiency of interpolatory type of iterations based on H-information of the fixed volume n:

Theorem 1. *Let $d_i \geqslant 0$ be integers. Let $v(\phi) = \sum_{i=0}^{m-1}(d_i+1) = n$ be fixed. Then the order $r(\phi) = (d_0+1)\prod_{i=1}^{m-1}(d_i+2)$ is maximized exactly when*

$$m = n, \quad d_i = 0 \quad (i = 0, \ldots, n-1) \tag{13}$$

or

$$m = n-1, \quad d_0 = 1,\ d_i = 0 \quad (i = 1, \ldots, n-2). \tag{14}$$

Theorem 1 states that in order to achieve as high as possible (*optimal*) order of convergence 2^{n-1} with n function evaluations of Hermitian type, a multipoint scheme has to start with a method of Newton's or Traub–Steffensen's type. All of the following steps of such multipoint scheme consume only one additional function evaluation of f (none of the derivatives) at the latest calculated approximation to the sought zero α. We will call such schemes *optimal Hermitian scheme*, or shorter *OH-schemes*, where Hermitian stands for the *type of information* used in iteration function.

According to the above discussion, developing techniques for multipoint root-finders will be displayed and explored on schemes that use Newton's or Traub–Steffensen's method as pre-conditioners. As proved in [31], method (5) is of least combinatorial cost, along with (1). However, parameter γ has been proved as a beneficial addendum, worthy of the investment. Let us consider the scheme that consumes in total n function evaluations per iteration,

$$\begin{cases} y_{k,1} = \mathcal{N}(x_k), \quad y_{k,0} = x_k \quad \text{or} \quad y_{k,1} = \mathcal{S}(x_k), \quad y_{k,0} = x_k + \gamma f(x_k), \\ y_{k,j} = \phi_j(x_k; y_{k,0}, \ldots, y_{k,j-1}), \quad 2 \leqslant j \leqslant n-1, \quad \phi_j \in \Psi_{2^j}, \\ x_{k+1} = y_{k,n-1}. \end{cases} \tag{15}$$

Here

$$\mathcal{N}(x_k) = x_k - \frac{f(x_k)}{f'(x_k)} \quad \text{(Newton's iteration)},$$

$$\mathcal{S}(x_k) = x_k - \frac{f(x_k)}{f[x_k, x_k + \gamma f(x_k)]} \quad \text{(Traub—Steffensen's iteration)},$$

were defined in (1) and (6), and $\phi_j \in \Psi_{2^j}$ denotes any *OH*-scheme with the order of convergence 2^j which uses $j+1$ function evaluations. Errors of approximations to the sought zero α will be denoted by

$$\varepsilon_{k,j} = y_{k,j} - \alpha, \; 0 \leqslant j \leqslant n-1, \quad \text{and} \quad \varepsilon_k = x_k - \alpha.$$

Based on the construction of scheme (15), it follows

$$\varepsilon_{k,j} = O\left(\varepsilon_k^{2^j}\right) = O\left(\varepsilon_k \prod_{i=0}^{j-1} \varepsilon_{k,i}\right), \quad (j = 0, \ldots, n-1). \tag{16}$$

There are two ways to raise the order of convergence (and, consequently, the informational efficiency) of the method (15): (1) by the reuse of old information (methods with memory), or (2) raising the order of convergence at the expense of an additional function evaluation per iteration.

Methods with memory that use optimal multipoint methods and self-accelerating parameters for further increase of convergence order will be discussed in Sections 7 and 8. For comparison, multipoint methods with memory can achieve order 2^n with n new function evaluations per iteration only if all the information, starting from x_0 are used in all iterations. Undoubtedly, such kind of information usage reduces every step of any multipoint method to the following

$$x_{k+1} = x_k - \frac{f(x_k)}{P'(x_k)} + O(\varepsilon_k^2),$$

with one new function evaluation (in fact, $f(x_k)$) per iteration, where $P(t; x_k, x_{k-1}, \ldots, x_0)$ is an interpolating polynomial based on all available information from x_0 to x_k. Efficiency index 2 is obtained in this manner. On the other hand, the only way to obtain order 2^n without the use of old information with the *OH*-scheme is to perform $n+1$ fresh function evaluations per iteration. In this section we will focus on developing higher order root-finders without memory.

Let us start with a non-optimal scheme based on *H*-information

$$\begin{cases} y_{k,1} = \mathcal{N}(x_k), \;\; y_{k,0} = x_k \quad \text{or} \quad y_{k,1} = \mathcal{S}(x_k), \;\; y_{k,0} = x_k + \gamma f(x_k), \\ y_{k,j} = \phi_j(x_k; y_{k,0}, \ldots, y_{k,j-1}), \quad 2 \leqslant j \leqslant n-1, \quad \phi_j \in \Psi_{2^j}, \\ x_{k+1} = \mathcal{N}(y_{k,n-1}) = y_{k,n-1} - \frac{f(y_{k,n-1})}{f'(y_{k,n-1})}. \end{cases} \tag{17}$$

Obviously, (17) represents a composition of *OH*-scheme (15) and Newton's iteration in the last step. According to Traub's theorem of composition of iterative functions [63, Th. 2.4], the scheme (17) obtains the desired augmented order 2^n but achieves it with $n+2$ function evaluations. To optimize (17), we will cut down by one the number of function evaluations with the approximation of $f'(y_{k,n-1})$ based on the rest of the available data from the current iteration. This approximation has to be of such quality that the newly developed scheme retains the order 2^n.

In Sections 4.1 and 4.2 we will consider the construction of some classes of general multipoint methods based on the scheme (17) and approximations of the derivative. In Section 4.3 we abandon the scheme (17) and present inverse interpolation approach in order to increase the order of convergence.

4.1. Direct interpolation

Let g be a sufficiently differentiable function that coincides with f at $m+1$ *H*-information points $z_0, \ldots, z_m \in \{x_k, y_{k,0}, \ldots, y_{k,n-1}\}, 1 \leqslant m \leqslant n$. The interpolating conditions are based on the *H*-information type function evaluations used in the current iteration. Nodes $z_0, \ldots, z_m$ are lexicographically ordered by their indices, which means that we assume that if $z_i = y_{k,j_i}$ then $j_i < j_{i+1}$. Therefore, if $x_k \in \{z_0, \ldots, z_m\}$ then $z_0 = x_k$, or if $y_{k,n-1} \in \{z_0, \ldots, z_m\}$ then $z_m = y_{k,n-1}$. The interpolating conditions are $g(z_j) = f(z_j)$ for $0 \leqslant j \leqslant m$, (if $z_0 = z_1 = x_k$ then $g'(x_k) = f'(x_k)$ is among the interpolating conditions instead of $g(z_1) = f(z_1)$) and depend on the type of the first step in the scheme (17).

We shall use an approximate $f'(y_{k,n-1}) \approx g'(y_{k,n-1})$ in the final step of (17). The new iterative scheme becomes

$$\begin{cases} y_{k,1} = \mathcal{N}(x_k), \; y_{k,0} = x_k \quad \text{or} \quad y_{k,1} = \mathcal{S}(x_k), \; y_{k,0} = x_k + \gamma f(x_k), \\ y_{k,j} = \phi_j(x_k; y_{k,0}, \ldots, y_{k,j-1}), \quad 2 \leqslant j \leqslant n-1, \quad \phi_j \in \Psi_{2^j}, \\ x_{k+1} = y_{k,n-1} - \frac{f(y_{k,n-1})}{g'(y_{k,n-1})}. \end{cases} \tag{18}$$

The symbol $I(a_0, a_1, \ldots, a_s)$ will denote the minimal interval which contains points $a_0, \ldots, a_s$. For easier inscription, errors are introduced as $\varepsilon_{z_j} = z_j - \alpha$, to emphasize that these are the interpolating points at which g coincides with f.

According to Cauchy mean value theorem, for t in a close neighborhood of the zero α, there exists a $\xi_t \in I(t, z_0, \ldots, z_m)$ (thus $\xi_t - \alpha$ is at least $O(\max\{|\varepsilon_{z_0}|, |t-\alpha|\})$) such that

$$f(t) - g(t) = \frac{(f-g)^{(m)}(\xi_t)}{m!} \prod_{j=0}^{m} (t - z_j) \sim \frac{(f-g)^{(m)}(\alpha)}{m!} \prod_{j=0}^{m} (t - z_j). \tag{19}$$

After differentiating and taking $t = y_{k,n-1}$ in (19), having in mind Taylor's development and relations (16), we obtain

$$g'(y_{k,n-1}) = f'(\alpha) + O\left(\prod_{j=0}^{m-1} \varepsilon_{z_j}\right). \tag{20}$$

According to (20) and Taylor's representation we have

$$\varepsilon_{k+1} = \varepsilon_{k,n-1} - \frac{f(y_{k,n-1})}{g'(y_{k,n-1})} = \varepsilon_{k,n-1} - \frac{f'(\alpha)\varepsilon_{k,n-1} + O(\varepsilon_{k,n-1}^2)}{f'(\alpha) + O\left(\prod_{j=0}^{m-1} \varepsilon_{z_j}\right)} = \varepsilon_{k,n-1}\left[1 - \frac{1 + O(\varepsilon_{k,n-1})}{1 + O\left(\prod_{j=0}^{m-1} \varepsilon_{z_j}\right)}\right].$$

From the last relation, we find the error estimate

$$\varepsilon_{k+1} = O\left(\varepsilon_{k,n-1} \prod_{j=0}^{m-1} \varepsilon_{z_j}\right). \tag{21}$$

Our goal is to achieve $\varepsilon_{k+1} = O(\varepsilon_k^{2^n}) = O\left(\varepsilon_k \prod_{j=0}^{n-1} \varepsilon_{k,j}\right)$, (see (16)). Hence, the iterative scheme (18) will be optimal if and only if $m = n$, in other words, if all available function evaluations from the current iteration are used in approximating $f'(y_{k,n-1})$.

When constructing multipoint root-solvers, if we use the presented approach from the second step onward (for calculating $y_{k,2}, y_{k,3}, \ldots$), we can obtain varieties of classes of iterative methods that can be regarded as interpolatory methods in a wider sense [71] than the one defined by Traub in [63].

While constructing multipoint methods, beside a high order of convergence, complexity of formulae involved (combinatorial cost) must be taken into account [64]. For this reason, the complexity of the derivative of the interpolating function g is essential when choosing g. In practice, polynomials or rational functions do make the obvious and most common choice for the function g. Minimal degree interpolating polynomials are mostly preferred, not only because of their wide and exhaustive study, but also due to the fact that we lose a dose of 'uncertainty' ($g^{(m)}(\xi_t)$ is annihilated in (19)) when extrapolating f from such polynomials. Among many examples we mention here the n-step families of methods: the Hermite interpolation based family [49] and the derivative free Zheng–Li–Huang family [73]. In Section 8 we devote more attention to the latter family.

4.2. Weight functions

Another technique has distinguished itself during the last decade. It has been used in the construction of *OH* iterative methods that are not necessarily of interpolatory type, even in a wide sense, and the construction of non-*H*-information iterative methods, just as well. The general idea will be presented on *H*-information based iterative methods.

Again, start from the non optimal scheme (17) of order 2^n. To optimize (17), as mentioned above, we need a very good approximation of $f'(y_{k,n-1})$. In order to preserve low computational cost, an approximate value of $f'(y_{k,n-1})$ should be based on some close value already calculated in one of the previous steps of the ongoing iteration, say $g'(y_{k,s}),\ s < n-1$. Usually $f'(x_k)$ or $f[x_k, y_{k,0}]$ are used in practice for $g'(y_{k,s})$, depending on the first predictor step in (17). However, such approximation to $f'(y_{k,n-1})$ can hardly give the desired optimal order of convergence because it does not rely on all available information. To get to the optimal 2^n we 'boost' the derivative approximation $g'(y_{k,s})$ by involving all the available information. The key is to find a minimal degree multivariate polynomial $P(t_1, \ldots, t_n)$ where variables $t_1, \ldots, t_n$ are a combination of fractions

$$\frac{f(y_{k,n-1})}{f(y_{k,n-2})}, \frac{f(y_{k,n-2})}{f(y_{k,n-3})}, \ldots, \frac{f(y_{k,1})}{f(y_{k,0})}, \frac{f(y_{k,1})}{f(x_k)},$$

or

$$\frac{f(y_{k,n-1})}{f(y_{k,n-2})}, \frac{f(y_{k,n-2})}{f(y_{k,n-3})}, \ldots, \frac{f(y_{k,1})}{f(x_k)}, \frac{f(x_k)}{f'(x_k)},$$

based on the available information depending on the predictor step in (17). The multivariate polynomial P should satisfy the following condition

$$f'(y_{k,n-1}) = \frac{g'(y_{k,s})}{P(t_1, \ldots, t_n)} + O(\varepsilon_{k,n-1}) \tag{22}$$

so that the newly designed scheme

$$\begin{cases} y_{k,1} = N(x_k), y_{k,0} = x_k \quad \text{or} \quad y_{k,1} = S(x_k), y_{k,0} = x_k + \gamma f(x_k), \\ y_{k,j} = \phi_j(x_k; y_{k,0}, \ldots, y_{k,j-1}), \quad 2 \leqslant j \leqslant n-1, \quad \phi_j \in \Psi_{2^j}, \\ x_{k+1} = N(y_{k,n-1}) = y_{k,n-1} - \frac{f(y_{k,n-1})}{g'(y_{k,s})} P(t_1, \ldots, t_n), \end{cases} \tag{23}$$

retains order 2^n.

Observe that in an *OH*-scheme of the form (23) the following is valid

$$\frac{f(y_{k,j+1})}{f(y_{k,j})} \to 0, \quad \text{and} \quad \frac{f(y_{k,1})}{f(x_k)} \to 0, \quad \frac{f(x_k)}{f'(x_k)} \to 0, \quad \text{when } k \to \infty \tag{24}$$

for all $j \in \{0, \dots, n-2\}$. For this reason, when the required polynomial P exists, it can be regarded as the Taylor expansion of a multivariate function $W(t_1, \dots, t_n)$ in the neighborhood of $T = (0, \dots, 0)$, formally called a *weight function*. Thus (23) becomes

$$\begin{cases} y_{k,1} = N(x_k), y_{k,0} = x_k \quad \text{or} \quad y_{k,1} = S(x_k), y_{k,0} = x_k + \gamma f(x_k), \\ y_{k,j} = \phi_f(x_k; y_{k,0}, \dots, y_{k,j-1}), \quad 2 \leqslant j \leqslant n-1, \quad \phi_j \in \Psi_{2^j}, \\ x_{k+1} = y_{k,n-1} - \frac{f(y_{k,n-1})}{g'(y_{k,n-1})} W(t_1, \dots, t_n). \end{cases} \tag{25}$$

Properties of the weight function W, sufficient for obtaining the optimal order 2^n of (25), are then expressed by the coefficients of the polynomial P as values of corresponding partial derivatives of W at the point $T = (0, \dots, 0)$.

Evidently, the enlargement of the number of variables in P, and thus in W, leads to the increase of the complexity of the function W; besides, sufficient conditions become more and more complicated, even when symbolic computation is applied. It is worth emphasizing that managing great number of variables of W is useless if such an approach does not considerably improve convergence characteristics of the designed method. Furthermore, recall that more complicated forms increase combinatorial cost.

When dealing with non-*H*-information methods, such as of Jarratt's type, limits (24) do not necessarily hold. Then the central point T of Taylor's expansion for W has to be determined from case to case.

The presented technique of convergence acceleration includes techniques presented in Sections 4.1 and 4.3.

We close this section with a comment on additional criteria for choosing weight functions and free parameters in iterative multipoint methods. In solving nonlinear equations we endeavor to find fixed points, that are candidates for zeros of the given equation. However, many multipoint methods have fixed points that are not desired zeros of the function. These points are called *extraneous* fixed points, see Vrscay and Gilbert [70]. As described in [40], the extraneous points could be attractive, which leads to the iteration trap producing undesirable results. To prevent this inconvenient behavior of multipoint methods based on weight functions, weight functions or involved free parameters have to be suitably chosen. Their choice should be carried out in such a manner to restrict the extraneous fixed point to a suitable domain (usually the boundary of a basin of attraction), say the imaginary axis, as done in [40] using conjugacy maps for quadratic polynomials.

4.3. Inverse interpolation

We will consider the following *OH*-scheme

$$\begin{cases} y_{k,1} = \mathcal{N}(x_k), y_{k,0} = x_k \quad \text{or} \quad y_{k,1} = \mathcal{S}(x_k), y_{k,0} = x_k + \gamma f(x_k), \\ y_{k,j} = \phi_j(x_k; y_{k,0}, \dots, y_{k,j-1}), \quad 2 \leqslant j \leqslant n-1, \quad \phi_j \in \Psi_{2^j}, \\ x_{k+1} = \mathcal{R}(0), \end{cases} \tag{26}$$

which is a composition of (15) and an inverse interpolating step

$$x_{k+1} = \mathcal{R}(0) = \mathcal{R}(0; y_{k,n-1}, \dots, y_{k,0}, x_k)$$

for the final approximation. An additional $(n+1)-$st function evaluation $f(y_{k,n-1})$ at the point $y_{k,n-1}$ is used in constructing the inverse interpolatory polynomial $\mathcal{R}(t)$ to raise the order of convergence from 2^{n-1} of the scheme (15) to 2^n of the new scheme (26).

Let $\mathcal{R}(t)$ represent a minimal degree polynomial that satisfies interpolating conditions

$$\begin{cases} \mathcal{R}(f(x_k)) = x_k, \quad \mathcal{R}(f(y_{k,j})) = y_{k,j}, \ j = 1, \dots, n, \quad \text{and} \\ \mathcal{R}(f(y_{k,0})) = y_{k,0} \text{ if } y_{k,0} = x_k + \gamma f(x_k), \quad \text{or} \quad \mathcal{R}'(f(x_k)) = 1/f'(x_k) \text{ if } y_{k,0} = x_k. \end{cases} \tag{27}$$

From Traub's study of interpolatory iterations [63] there follows

$$\varepsilon_{k+1} = O\left(\varepsilon_k \prod_{j=0}^{n-1} \varepsilon_{k,j}\right) = O\left(\varepsilon_k^{2^n}\right),$$

so that the scheme (26) is really an *OH*-scheme.

Remark 2. Low computational cost is the reason for restricting $\mathcal{R}$ to the polynomial form. Any function satisfying conditions (27) would give the same convergence order.

By applying the presented accelerating technique based on inverse interpolation from second step onward (for calculating $y_{k,2}, \dots$), Kung and Traub [32] obtained their famous n-point families of arbitrary order of convergence. More attention to these families will be given in Section 8.

Remark 3. It can be proved that using less interpolating points in (27) gives lower order of convergence for method (26), see Section 4.1.

Special cases of the general schemes (18) and (25) in the form of specific two- and three-point iterative methods are considered in Sections 5 and 6, while inverse interpolation scheme (26) is studied in Section 7. Generalized n-point optimal methods without memory of Traub–Steffensen's type are presented in Section 8 as the base for constructing n-point methods with memory.

5. Two-point optimal methods

Traub's extensive study of cubically convergent two-point methods, given in his book [63], is the first systematic research of multipoint methods. Although Truab's methods are not optimal, the presented techniques for their derivation have had great influence to later development of multipoint methods. The first optimal two-point method was constructed by Ostrowski [46], four years before Traub's investigation in this area described in [63]. Ostrowski's method is given by the two-step scheme

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \frac{f(y_k)}{f'(x_k)} \cdot \frac{f(x_k)}{f(x_k)-2f(y_k)}, \end{cases} \quad (k = 0, 1, \ldots). \tag{28}$$

Five different derivations of the method (28) are presented in [53]. Ostrowski's method (28) can be obtained as a special case of many families of two-point methods developed in later papers. Besides, this method often gives the best results in practice in the class of methods with similar characteristics. These facts point that Ostrowski's method could be regarded as an "essential method" which makes the "core" of many families of two-point methods.

A generalization of Ostrowski's method (28) was proposed by King [26] in the form

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \frac{f(y_k)}{f'(x_k)} \cdot \frac{f(x_k)+\beta f(y_k)}{f(x_k)+(\beta-2)f(y_k)}, \end{cases} \quad (k = 0, 1, \ldots), \tag{29}$$

where $\beta \in \boldsymbol{R}$ is a parameter. King's family (29) of two-point methods is optimal and has order four. It is easy to see that Ostrowski's method (28) is a special case of (29) for $\beta = 0$. Interestingly, some authors derived their own optimal two-point methods about thirty years after King's paper, even though they appear to be just special cases of King's method; for example, Kou et al. [27] ($\beta = 1$), Chun [10] ($\beta = 2$), Chun and Ham [11], Kou et al. [29].

Consider the iterative formula (29) and other similar methods where the first step is Newton's iteration. As explained in Section 4, we observe that the construction of optimal two-point methods arises from general scheme (17) for $n = 2$ ($y_{k,1} = y_k$), which is really just the doubled Newton's method:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - \frac{f(y_k)}{f'(y_k)}, \end{cases} \quad (k = 0, 1, \ldots). \tag{30}$$

Our aim is to replace $f'(y_k)$ in the second step by a suitable approximation which does not require new information. A fruitful approach is to put $f'(y_k) = w(t_k)f'(x_k)$ ($t_k = f(y_k)/f(x_k)$) in (30) and find the form of the *weight function* $w(t)$ so that the new method has order four. In this way only three function evaluations would be required. For example, Chun [10] showed that the family with the weight function w will have order four if w is a real function satisfying $w(0) = 1,\ w'(0) = -2$ and $|w''(0)| < \infty$.

It turned out that the choice $w(t) = 1/g(t)$ under the conditions $g(0) = 1,\ g'(0) = 2,\ |g''(0)| < \infty$, presented in [52], is slightly more convenient since it produces two-point methods without altering the weight function or its development into series. The corresponding family of two-point methods has the form

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = y_k - g(t_k)\frac{f(y_k)}{f'(x_k)} \quad (t_k = f(y_k)/f(x_k)), \end{cases} \quad (k = 0, 1, \ldots). \tag{31}$$

We give five forms of the weight function g:

$$\begin{aligned} g(t) &= \frac{1+\beta t}{1+(\beta-2)t}, \quad \beta \in \boldsymbol{R}, \\ g(t) &= \left(1+\frac{2}{r}t\right)^r, \quad r \in \boldsymbol{Q}, \\ g(t) &= \frac{1+\gamma t^2}{1-2t}, \quad \gamma \in \boldsymbol{R}, \\ g(t) &= \frac{1}{1-2t+at^2}, \quad a \in \boldsymbol{R}, \\ g(t) &= \frac{t^2+(c-2)t-1}{ct-1}, \quad c \in \boldsymbol{R}. \end{aligned}$$

As described in [53], such choice in (31) produces either directly or as special cases two-point families (or particular methods) presented in [9,10,26,27,36,46].

Another example of the scheme (25), presented in [52], uses the following approximations in the doubled Newton's method (30):

$$f'(x) \approx \phi(x) = \frac{f(x+\gamma f(x)) - f(x)}{\gamma f(x)},$$

$$f'(y) \approx \frac{\phi(x)}{h(t,s)},$$

where $h(t,s)$ is a differentiable function in two real variables

$$t = \frac{f(y)}{f(x)}, \quad s = \frac{f(y)}{f(x+\gamma f(x))}.$$

In this manner the following family of two-point methods was constructed in [52]

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{\phi(x_k)}, \\ x_{k+1} = y_k - h(t_k,s_k)\frac{f(y_k)}{\phi(x_k)}, \end{cases} \quad (k=0,1,\ldots). \tag{32}$$

By the means of symbolic computation, the *weight function h* was determined to provide the fourth-order of convergence of (32).

Theorem 2. *Let $h(t,s)$ be a differentiable function in two variables that satisfies the conditions $h(0,0) = h_t(0,0) = h_s(0,0) = 1$, where the subscript indices t and s point to the partial derivatives to the arguments t and s. If an initial approximation x_0 is sufficiently close to a zero α of f, then the order of the family of two-point methods (32) is equal to four.*

The choice of different weight functions h, satisfying the condition given in Theorem 2, produces various optimal two-point methods without derivatives. Here are several samples:

$$h(t,s) = \frac{1+t}{1-s}, \tag{33}$$

$$h(t,s) = \frac{1}{(1-t)(1-s)}, \tag{34}$$

$$h(t,s) = \frac{1}{1-t-s}, \tag{35}$$

$$h(t,s) = (1+t)(1+s). \tag{36}$$

For example, the form (34) gives a special case of Kung–Traub's family (of order four), see Section 8. Methods which are either very similar to (35) or arise from (35) were considered in the papers [35,47,55,57].

Previously considered two-point methods use quadratically convergent methods (either Newton's or Traub–Steffensen's iteration) in the first step. Another type of optimal two-point methods with non-H-information starts with the iteration of linear convergence in the first step, such as follows:

$$\begin{cases} y_k = x_k - \frac{2}{3}\frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = x_k - \Phi(f'(x_k), f'(x_k), f'(y_k)), \end{cases} \quad (k=0,1,\ldots). \tag{37}$$

First methods of this type were developed by Jarratt [22,23] so that the first step in (37) is usually called "Jarratt's step."

Now we consider a family of Jarratt's type of two-point methods which produces some existing and some new methods, see [53, Ch. 2]. Let

$$y = x - \frac{2}{3}u(x), \quad u(x) = \frac{f(x)}{f'(x)}, \quad t = t(x) = \frac{f'(y)}{f'(x)} \quad \text{and} \quad t_k = t(x_k).$$

We start from the iterative scheme

$$\begin{cases} y_k = x_k - \frac{2}{3}u(x_k), \\ x_{k+1} = x_k - q(t_k)u(x_k), \end{cases} \tag{38}$$

where $q(t)$ is a weight function to be determined such that the method (38) is of order four. It is suitable to approximate the weight function $q(t)$ by its Taylor's polynomial of third degree, thus

$$q(t) \approx q_0 + q_1(t-1) + \frac{1}{2}q_2(t-1)^2 + \frac{1}{6}q_3(t-1)^3, \quad q_r = q^{(r)}(1) \quad (r=0,1,2,3).$$

By the help of symbolic computation we arrive at the required conditions

$$q_0 = q(1) = 1, \quad q_1 = q'(1) = -\frac{3}{4}, \quad q_2 = q''(1) = \frac{9}{4}, \quad |q'''(1)| < \infty. \tag{39}$$

The corresponding error relation is given by

$$\varepsilon_{k+1} = \left(-c_2c_3 + \frac{1}{9}c_4 + c_2^3\left(5 + \frac{32}{81}q'''(1)\right)\right)\varepsilon_k^4 + O(\varepsilon_k^5). \tag{40}$$

By virtue of (39) and (40) we can state the following theorem.

Theorem 3. *Let $f : D \subset \mathbf{R} \to \mathbf{R}$ be a sufficiently differentiable function having a simple zero α in an open interval $V_f \subset D$. If x_0 is close enough to α and the conditions (39) hold, then the family (38) is of order four.*

A number of Jarratt-like methods can be generated choosing various forms of $q(t)$. We can obtain a rather general two-parameter family with Jarratt's step using the rational function

$$q(t) = \frac{16(1 - a - b + bt + at^2)}{-5 + 8a - 4b + (30 - 48a - 8b)t + (-9 + 40a + 12b)t^2}. \tag{41}$$

For example, taking $q(t) = \frac{3t+1}{6t-2}$ $(a = 0,\ b = \frac{3}{4})$ in (38), we get a particular case of Jarratt's method (rediscovered much later by Basu [4])

$$\begin{cases} y_k = x_k - \frac{2}{3}\frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = x_k - \frac{1}{2}\frac{f(x_k)}{f'(x_k)} + \frac{f(x_k)}{f'(x_k) - 3f'(y_k)}, \end{cases} \quad (k = 0, 1, \ldots).$$

Note that Basu's method [4] is obtained from (38) for $q(t) = t/\left(\frac{3}{16}t^2 + \frac{11}{8}t - \frac{9}{16}\right)$ $(a = 0,\ b = 1)$. Here are three other examples arising from (41):

$$\begin{aligned} &(\text{J-1}): \quad (a = 0,\ b = 15/4) &&\Rightarrow \quad q(t) = \tfrac{15t-11}{9t^2-5}, \\ &(\text{J-2}): \quad (a = 0,\ b = 0) &&\Rightarrow \quad q(t) = -\tfrac{16}{9t^2-30t+5}, \\ &(\text{J-3}): \quad (a = 9/8,\ b = -3) &&\Rightarrow \quad q(t) = \tfrac{1}{8}(23 - 24t + 9t^2). \end{aligned}$$

Note that the weight function (J-3) is the only one from the class satisfying (39) that has a polynomial form.

More generally, taking

$$q(t) = \frac{t^2(3 + 2\theta) + 4t(\theta - 1) + 9 - 6\theta}{8t(1 + (t - 1)\theta)}$$

(for $a = (3 + 2\theta)/8,\ b = (\theta - 1)/2$ in (41), $\theta \neq 0, 1$), the Jarratt family proposed in [22] follows from (38).

Using a different derivation technique, Chun et al. [12] constructed a generalized family of Jarratt's type, which is essentially the same as (38) and deals with the "shifted" argument,

$$\begin{cases} y_k = x_k - \frac{2}{3}\frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = x_k - H(s_k)\frac{f(x_k)}{f'(x_k)}, \end{cases} \tag{42}$$

where

$$s_k = \frac{3}{2}\frac{f'(x_k) - f'(y_k)}{f'(x_k)} = \frac{3}{2}(1 - t_k).$$

The family (42) is of order four if

$$H(0) = 1, \quad H'(0) = \frac{1}{2}, \quad H''(0) = 1,$$

as shown in [12]. Note that these conditions are equivalent to (39) since $g(t) = g\left(1 - \frac{2s}{3}\right) = H(s)$.

A number of old and new methods of Jarratt's type can be obtained from (42). For example, choosing

$$H(s) = \frac{3}{4}\cdot\frac{s\left(\gamma s + \frac{3}{2}\right)}{\left(\delta s + \frac{3}{2}\right)\left(\beta s + \frac{3}{2}\right)}$$

in (42), where $\gamma = \delta + \beta - \frac{3}{2},\ \delta, \beta \in \mathbf{R}$, we obtain the two-point family presented in [28].

It is interesting to note that a linearly convergent iteration has been used as the predictor step for the construction of optimal two-point methods for multiple zeros, see [53, Sec. 5]. We present here only the generalized method of Zhou et al. [74],

$$\begin{cases} y_k = x_k - \theta \frac{f(x_k)}{f'(x_k)}, \\ x_{k+1} = x_k - \psi(t_k)\frac{f(x_k)}{f'(x_k)}, \quad t_k = \frac{f'(y_k)}{f'(x_k)}, \end{cases} \tag{43}$$

where θ is a real parameter and ψ is at least twice differentiable. The parameter θ and the function ψ have to be determined so that the two-point method (43) reaches fourth order. This task is considered in the following theorem proved in [74].

Theorem 4. *Let α be a multiple zero of multiplicity m of a function $f : V_f \subset \mathbf{R} \to \mathbf{R}$ for an open interval V_f. If an initial approximation x_0 is sufficiently close to α, then the order of convergence of the method (43) is at least four when the following conditions are satisfied:*

$$\theta = \frac{2m}{m+2},$$

$$\psi(\lambda) = m, \quad \psi''(\lambda) = -\frac{1}{4}m^{3-m}(2+m)^m, \quad \psi'(\lambda) = \frac{1}{4}m^4\left(\frac{m}{m+2}\right)^{-2m},$$

where $\lambda = \left(\frac{m}{m+2}\right)^{m-1}$.

The family of two-point methods (43) contains Li–Liao–Cheng's fourth-order method [33] and Sharma–Sharma's fourth-order method [59] as special cases. Note that an optimal three-point method for finding multiple zeros has not been constructed yet at present, neglecting very expensive "trick-methods" based on the use of the functions $f(x)^{1/m}$, $F(x) = f(x)/f'(x)$ or similar to those.

6. Three-point optimal methods

As explained in Section 4 (see the scheme (17) and (26)), once an optimal *OH*-two-point method (or family of methods) is stated, it is easy to construct optimal three-point methods of order eight that require four function evaluations. It can be constructed both by derivative estimation (Sections 4.1 and 4.2) or inverse interpolation (Section 4.3).

As presented in Section 4, a general scheme (17) for $n = 3$ is

$$\begin{cases} (1) \quad y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ (2) \quad z_k = \phi_f(x_k, y_k), \quad \phi_f \in \Psi_4, \\ (3) \quad x_{k+1} = z_k - \frac{f(z_k)}{f'(z_k)}. \end{cases} \tag{44}$$

Note that the first two steps define an optimal two-point method from the class Ψ_4 with the order $r_1 = 4$. Using Traub's theorem on composite iterative methods [63, Th. 2.4], the convergence order of (44) is equal to $r_1 \cdot r_2 = 8$ where $r_2 = 2$ is the order of Newton's method in the third step.

Note that the three-point method (44) is not optimal since it requires five function evaluations per iteration. To reduce the number of function evaluations, we approximate $f'(z_k)$ using the available data $f(x_k)$, $f'(x_k)$, $f(y_k)$ and $f(z_k)$. To do this, we can approximate $f'(z_k)$ using one of the following methods as described in Sections 4.1, 4.2 and 4.3:

(i) Construct Hermite's interpolating polynomial H_3 of degree 3 at the nodes $x,\ y,\ z$,

$$H_3(t) = a + b(t-x) + c(t-x)^2 + d(t-x)^3,$$

under the conditions

$$H(x_k) = f(x_k), \quad H(y_k) = f(y_k), \quad H(z_k) = f(z_k), \quad H'(x_k) = f'(x_k)$$

and utilize the approximation

$$f'(z_k) \approx H_3'(z_k) = 2(f[x_k, z_k] - f[x_k, y_k]) + f[y_k, z_k] + \frac{y_k - z_k}{y_k - x_k}(f[x_k, y_k] - f'(x_k))$$

in the third step of the iterative scheme (44).

This idea was employed in [30,54,49,67]. In this way we obtain the family of three-point methods

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k = \phi_f(x_k, y_k), \quad \phi_f \in \Psi_4, \\ x_{k+1} = z_k - \frac{f(z_k)}{H_3'(z_k)}. \end{cases} \tag{45}$$

Note that the use of Hermite's interpolating polynomial of degree higher than 3 cannot increase the order of convergence.

(ii) Form an interpolating rational function of the form $P_m(t)/Q_n(t)$, where $m + n = 3$ $(0 \leqslant m, n \leqslant 3)$ and one of the polynomials P and Q being monic. See the references [48,58]. In particular, for $m = 3,\ n = 0$ one obtains Hermite's interpolating polynomial applied in (i). For example, we can interpolate f by a rational function

$$r(t) = \frac{b_1 + b_2(t-x) + b_3(t-x)^2}{1 + b_4(t-x)} \quad (b_2 - b_1 b_4 \neq 0), \tag{46}$$

see [48]. From (46) we find

$$r'(t) = \frac{b_2 - b_1 b_4 + b_3(t-x)(2 + b_4(t-x))}{(1 + b_4(t-x))^2}. \tag{47}$$

The unknown coefficients $b_1, \ldots, b_4$ are determined from the conditions

$$r(x_k) = f(x_k), \quad r(y_k) = f(y_k), \quad r(z_k) = f(z_k), \quad r'(x_k) = f'(x_k)$$

and they are given by

$$b_1 = f(x_k), \quad b_3 = \frac{f'(x_k)f[y_k,z_k] - f[x_k,y_k]f[x_k,z_k]}{x_k f[y_k,z_k] + \frac{y_k f(z_k) - z_k f(y_k)}{y_k - z_k} - f(x_k)},$$

$$b_4 = \frac{b_3}{f[x_k,y_k]} + \frac{f'(x_k) - f[x_k,y_k]}{(y_k - x_k)f[x_k,y_k]}, \quad b_2 = f'(x_k) + b_4 f(x_k).$$

Substituting these coefficients in (47) yields $r'(z_k)$. The corresponding family has the form of (45) with $r'(z_k)$ instead of $H_3'(z_k)$.

Remark 4. In the recent paper [58] a three-point method with a rational approximation of the form $P_1(x)/Q_2(x)$ was considered. It is hard to say if this approximation is better or not than (46) of the form $r(x) = P_2(x)/Q_1(x)$ since the quality of approximation depends on the structure of the function approximated, see [5,6] for more details. However, it is clear that the method (45) is more general since an arbitrary optimal two-point method is used there, compared with a specific two-point method (King's family) applied in [58].

(iii) Apply a suitable function $w(t)$ that approximates $f(t)$ in such way that the three-point methods attain order eight. Note that $w(t)$ contains rational functions and Hermite's interpolating polynomial as special cases. It is possible to deal with weight functions of two or more arguments (see (25)), or combine two or more weight functions with one or more arguments. These weight functions and their arguments must use only available information to keep the number of function evaluations not greater than four. Several optimal three-point methods were constructed in this way, see, e.g., [18–20,34,62,68].

The approach presented in [17] consists of the weight function approach (Section 4.2) applied in two subsequent steps, which includes substitution of the derivatives $f'(y)$ and $f'(z)$ in the second and third step of

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - \frac{f(y_k)}{f'(y_k)}, \\ x_{k+1} = z_k - \frac{f(z_k)}{f'(z_k)} \end{cases} \tag{48}$$

by the approximations

$$f'(y) = \frac{f'(x)}{p(t)}, \quad f'(z) = \frac{f'(x)}{q(t,s)}, \quad \text{where } t = \frac{f(y)}{f(x)}, \quad s = \frac{f(z)}{f(y)}, \tag{49}$$

where p and q are some functions of one and two variables (respectively) that do not require any new information. These functions should be chosen so that designed three-point methods with fixed number of four function evaluations achieve order eight. Then the following thee-point iterative scheme can be constructed:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - p(t_k)\frac{f(y_k)}{f'(x_k)}, \\ x_{k+1} = z_k - q(t_k,s_k)\frac{f(z_k)}{f'(x_k)}. \end{cases} \tag{50}$$

The following theorem was proved in [13].

Theorem 5. *Let a, b and c be arbitrary constants. If p and q are arbitrary real functions with Taylor's series of the form*

$$p(t) = 1 + 2t + \frac{a}{2}t^2 + \frac{b}{6}t^3 + \cdots, \tag{51}$$

$$q(t,s) = 1 + 2t + s + \frac{2+a}{2}t^2 + 4ts + \frac{c}{2}s^2 + \frac{6a+b-24}{6}t^3 + \cdots, \tag{52}$$

then the family of three-point methods (50) is of order eight. It is assumed that higher order terms in (51) and (52), are represented by the dots, and they can take arbitrary values.

Slightly less general formula with specific values $a = 4,\ b = 0,\ c$ arbitrary was derived in [17].

Taking various functions p and q in (50) satisfying the conditions (51) and (52), some new and some existing three-point methods can be obtained from (50). To keep small computational costs, it is reasonable to choose p and q as simple as possible, for example, in the form of polynomials or rational functions as follows:

$$p_1(t) = 1 + 2t + 2t^2, \quad p_2(t) = \frac{1}{1 - 2t + 2t^2}, \quad p_3(t) = \frac{1 + t + t^2}{1 - t + t^2},$$
$$q_1(t,s) = 1 + 2t + s + 3t^2 + 4ts, \quad q_2(t,s) = \left(2t + \frac{5}{4}s + \frac{1}{1 + t + \frac{3}{4}s}\right)^2,$$
$$q_3(t,s) = \frac{1 - 4t + s}{(1 - 3t)^2 + 2ts}, \quad q_4(t,s) = \frac{1}{1 - 2t + t^2 + 4t^3 - s}.$$

Here are a few variants of three-point methods with weight functions. Starting from tripled Newton's method (48) and using approximations

$$f'(y) \approx \widetilde{f}'(y) = \frac{f'(x)}{p(t)}, \quad \text{and} \quad f'(z) \approx \frac{\widetilde{f}'(y)}{w(t,s)} = \frac{f'(x)}{p(t)w(t,s)},$$

computational cost of the method (50) can be slightly cut down. By means of symbolic computation it is easy to show that the order of the new scheme

$$\begin{cases} y_k = N(x_k) = x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - \frac{f(y_k)}{f'(x_k)} p(t_k), \quad t_k = \frac{f(y_k)}{f(x_k)}, \\ x_{k+1} = z_k - \frac{f(z_k)}{f'(x_k)} p(t_k) w(t_k, s_k), \quad s_k = \frac{f(z_k)}{f(y_k)}, \end{cases} \tag{53}$$

will be eight if p and w satisfy

$$p(t) = 1 + 2t + \frac{a}{2}t^2 + \cdots, \quad w(t,s) = 1 + s + t^2 + \frac{b}{2}s^2 + 2ts + (a - 6)t^3 + \cdots,$$

where, again, dots represent higher order terms that can take arbitrary values.

The next variant

$$\begin{cases} y_k = N(x_k) = x_k - u_k, \quad u_k = \frac{f(x_k)}{f'(x_k)}, \\ z_k = y_k - u_k p(t_k), \quad t_k = \frac{f(y_k)}{f(x_k)}, \\ x_{k+1} = z_k - u_k p(t_k) w(t_k, s_k), \quad s_k = \frac{f(z_k)}{f(y_k)}, \end{cases}$$

has also order eight if the weight functions p and w have the following Taylor expansions

$$\begin{cases} p(t) = t + 2t^2 + \frac{a}{6}t^3 + \cdots, \\ w(t,s) = s + s^2 + t^2 s + 2ts^2 + \frac{b}{6}s^3 + 0 \cdot t^4 + \frac{a-18}{3}t^3 s + \cdots \end{cases}$$

Derivative free variants based on weight functions can be derived in a similar way, see, e.g., [34,62,68]. For example, we start from the two point derivative free family (32) and add the third step

$$\begin{cases} y_k = S(x_k) = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \gamma f(x_k), \\ z_k = y_k - \frac{f(y_k)}{f[x_k, w_k]} h(t_k, s_k), \quad t_k = \frac{f(y_k)}{f(x_k)}, \quad s_k = \frac{f(y_k)}{f(w_k)} \\ x_{k+1} = z_k - \frac{f(y_k)}{f[x_k, w_k]} h(t_k, s_k) w(t_k, s_k, v_k), \quad v_k = \frac{f(z_k)}{f(y_k)}. \end{cases} \tag{54}$$

Using symbolic computation, it is easy to check that functions h and w with Taylor's expansions

$$\begin{cases} h(t,s) = 1 + t + s + \frac{a}{2}t^2 + bts + \frac{c}{2}s^2 + \cdots, \\ w(t,s,v) = 1 + v + \frac{d}{2}v^2 + ts + tv + sv + \frac{a-2}{2}t^3 + \frac{c-2}{2}s^3 + \frac{m}{6}v^3 + \frac{a+2b-4}{2}t^2 s + \frac{2b+c-4}{2}ts^2 + \cdots \end{cases}$$

guarantee order 8 of the method (54).

As presented in Section 4.3, some other techniques are possible. For example, consider the inverse interpolation

$$\mathcal{R}(f(x)) = a + b(f(x) - f(x_k)) + c(f(x) - f(x_k))^2 + d(f(x) - f(x_k))^2 (f(x) - f(y_k)). \tag{55}$$

Having in mind that

$$\begin{cases} f^{-1}[f(x_k),f(y_k)] = \frac{y_k-x_k}{f(y_k)-f(x_k)}, \\ f^{-1}[f(x_k),f(y_k),f(z_k)] = \frac{z_k-x_k}{f^{-1}[f(y_k),f(z_k)]-f^{-1}[f(x_k),f(y_k)]}, \\ f^{-1}[f(x_k),f(y_k),f(z_k),f(w_k)] = \frac{w_k-x_k}{f^{-1}[f(y_k),f(z_k),f(w_k)]-f^{-1}[f(x_k),f(y_k),f(z_k)]}, \end{cases} \tag{56}$$

we find the coefficients a, b, c, d appearing in (55)

$$\begin{aligned} &a = f^{-1}(f(x_k) = x_k, \quad b = f^{-1}[f(x_k),f(x_k)] = 1/f'(x_k), \\ &c = f^{-1}[f(x_k),f(x_k),f(y_k)] = \frac{y_k - x_k}{f^{-1}[f(x_k),f(y_k)] - f^{-1}[f(x_k),f(x_k)]}, \\ &d = f^{-1}[f(x_k),f(x_k),f(y_k),f(z_k)] = \frac{z_k - x_k}{f^{-1}[f(x_k),f(y_k),f(z_k)] - f^{-1}[f(x_k),f(x_k),f(y_k)]}. \end{aligned}$$

Then, substituting these coefficients in (55), we obtain the following presumably improved approximation

$$x_{k+1} = R(0) = \mathcal{N}(x_k) + c[f(x_k)]^2 - d[f(x_k)]^2 f(y_k). \tag{57}$$

As above, y_k is Newton's approximation and z_k is produced by any optimal fourth-order method. It was proved in [42] that the family of three-point methods (57) has the order eight.

There are arguments for and against root-solvers of a very high order. First of all, note that some families of optimal multipoint methods of arbitrary order could be of interest, at least from the theoretical point of view, if they generate particular methods of high computational efficiency (usually of reasonably low order of convergence). Typical examples are the Kung–Traub families [32] with optimal order 2^n for arbitrary $n \geqslant 1$.

In general, for solving most real-life problems (including mathematical models in many disciplines), double-precision arithmetic is good enough giving the accuracy of desired solutions or results of calculation with approximately 16 significant decimal digits, that is, an error of about 10^{-16}.

Investigations in the last decades have pointed out that there are some classes of problems when multi-precision capabilities are very important, such as Number theory, Experimental mathematics and many research fields including finite element modelling CAD, high energy physics, nonlinear process simulation, 3-D real-time graphic, statistics, security cryptography, and so on. In particular, the application of very fast iterative methods for solving nonlinear equations is justified if these methods serve for testing multi-precision arithmetic, whose improvement and development are a permanent task of many computer scientists and numerical analysts, see [7]. Nevertheless, although some special applications require the implementation of very fast algorithms, there is a reasonable limit in view of the desired accuracy. For example, approximations to the roots of nonlinear equations with, say, 200 or more accurate decimal digits are not required in practice at present.

In the book [53] the main interest is paid to multipoint methods with optimal order of convergence. We do the same in this paper. Namely, non-optimal methods with very high order are not of interest since they require extra function evaluations that additionally decrease their computational efficiency.

7. Inverse interpolation and multipoint methods with memory

Although the basic idea for the construction of multipoint methods with memory was launched by Traub almost fifty years ago in his book [63], this class of methods is very seldom considered in the literature in spite of high computational efficiency of this kind of root-solvers (see, e.g., [14–16,39,50,51,66]). Most of these methods are modifications of multipoint methods without memory with optimal order of convergence. They are constructed using mainly Newton's interpolation with divided differences for calculating self-correcting parameters. In this way, extremely fast convergence of new methods with memory is attained without additional function evaluations. As a consequence, these multipoint methods possess a very high computational efficiency. Other type of multipoint methods with memory is based on inverse interpolation (see [39,51]) and a special choice of initial approximations.

For illustration, we first consider a two-step method with memory constructed by inverse interpolation using Neta's idea from the paper [39] who derived in 1983 a very fast three-point method.

Let x_0, y_{-1} be two starting initial approximations to the sought root α. We first construct a two-point method calculating y_k by 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.

To compute y_k we use inverse interpolation starting from

$$x = \mathcal{R}(f(x)) = a + b(f(x) - f(x_k)) + c(f(x) - f(x_k))^2. \tag{58}$$

This polynomial of second degree has to satisfy the following conditions

$$x_k = \mathcal{R}(f(x_k)), \tag{59}$$

$$\frac{1}{f'(x_k)} = \mathcal{R}'(f(x_k)), \tag{60}$$

$$y_{k-1} = \mathcal{R}(f(y_{k-1})). \tag{61}$$

From (59) and (60) we get

$$a = x_k, \quad b = \frac{1}{f'(x_k)}. \tag{62}$$

Let us introduce a real function $\Phi(t)$ defined by

$$\Phi(t) = f^{-1}[f(x_k), f(x_k), f(t)] = \frac{1}{f(t) - f(x_k)}\left(\frac{t - x_k}{f(t) - f(x_k)} - \frac{1}{f'(x_k)}\right) \tag{63}$$

and let

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

denote Newton's iteration. According to (58) and (61) we find $c = \Phi(y_{k-1})$ so that, together with (62), it follows from (58)

$$y_k = \mathcal{R}(0) = x_k - \frac{f(x_k)}{f'(x_k)} + f(x_k)^2\Phi(y_{k-1}) = \mathcal{N}(x_k) + f(x_k)^2\Phi(y_{k-1}). \tag{64}$$

In the next step, we find x_{k+1} by carrying out the same calculation but using y_k instead of y_{k-1}. The constant c in (58) is now given by $c = \Phi(y_k)$ and we find from (58)

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} + f(x_k)^2\Phi(y_k) = \mathcal{N}(x_k) + f(x_k)^2\Phi(y_k), \tag{65}$$

where y_k is calculated by (64).

To start the iterative process (64) and (65), we request two initial approximations x_0 and y_{-1}. Here we meet a suitable fact that y_{-1} may take the value $\mathcal{N}(x_0)$ at the first iteration without any additional computational cost. Indeed, $\mathcal{N}(x_0)$ appears anyway in (64) and (65) for $k = 0$. In practical implementation such a choice of y_{-1} in (66) gives significant increase of the accuracy of obtained approximations, see numerical results given in [50].

The relations (64) and (65) define the two-point method with memory [50]:

$$\begin{cases} \text{Given } x_0, \quad y_{-1} = \mathcal{N}(x_0), \\ y_k = \mathcal{N}(x_k) + f(x_k)^2\Phi(y_{k-1}), \quad (k = 0, 1, \ldots), \\ x_{k+1} = \mathcal{N}(x_k) + f(x_k)^2\Phi(y_k), \end{cases} \tag{66}$$

where Φ is defined by (63).

As shown in [39], the determination of R-order of convergence of this type of methods can be carried out in an elegant manner using the following Herzberger's results [21]:

Theorem 6 (*Herzberger [21]*). *Let $x_{k+1} = \varphi(x_k, x_{k-1}, \ldots, x_{k-s+1})$ define a single step s-point method with memory. The matrix $M = (m_{ij})$ $(1 \leqslant i,j \leqslant s)$, associated with this method, has the elements*

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

The order of an n-step method $\varphi = \varphi_n \circ \varphi_{n-1} \circ \cdots \circ \varphi_1$ is the spectral radius of the product of matrices

$$M^{(n)} = M_n \cdot M_{n-1} \cdots M_1, \tag{67}$$

where the matrices M_r correspond to the iteration steps φ_r $(1 \leqslant r \leqslant n)$.

In the case of n-step methods for solving nonlinear equations, the matrix M_r is associated with the r-th step $(r = 1, \ldots, n)$, that is, M_n is concerned with the best approximation, etc., see the sketch of proof of Theorem 7. Observe that Herzberger's matrices are formed taking amount of information (function evaluations) required at a point, starting from the best to the worse approximation.

The order of convergence of the method (66) is given in the following theorem [50].

Theorem 7. *The two-point method (66) has R-order of convergence at least* $\rho(M^{(2)}) = (5+\sqrt{17})/2 \approx 4.561$, *where* $\rho(M^{(2)})$ *is the spectral radius of the matrix*

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

The proof of this theorem was given in [50] but with a slight flaw due to confused matrix multiplication so that we give here corrected proof. According to the relations (64) and (65) we form the respective matrices,

$$\begin{aligned} & x_{k+1} = \phi_1(y_k, x_k) \quad && y_k = \phi_2(x_k, y_{k-1}) \\ & M_2 = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}, \quad && M_1 = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}. \end{aligned}$$

Hence

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

The characteristic polynomial of the matrix $M^{(2)}$ 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 $\rho(M^{(2)}) \approx 4.561$, which gives the lower bound of the *R*-order of the method (66).

Remark 5. In the original proof given in [50] the matrices M_1 and M_2 were multiplied in reverse order, but with (incidently) the correct outcome: $r = 4.561\dot{2}$.

Using also inverse interpolation and the presented procedure, the following algorithms can be constructed:

Three-point method with memory, see [39]:

$$\begin{cases} \text{Given } x_0,\ y_{-1},\ z_{-1}, \\ y_k = \mathcal{N}(x_k) + (f(y_{k-1})\Phi(z_{k-1}) - f(z_{k-1})\Phi(y_{k-1}))\frac{f(x_k)^2}{f(y_{k-1})-f(z_{k-1})}, \\ z_k = \mathcal{N}(x_k) + (f(y_k)\Phi(z_{k-1}) - f(z_{k-1})\Phi(y_k))\frac{f(x_k)^2}{f(y_k)-f(z_{k-1})}, \\ x_{k+1} = \mathcal{N}(x_k) + (f(y_k)\Phi(z_k) - f(z_k)\Phi(y_k))\frac{f(x_k)^2}{f(y_k)-f(z_k)}. \end{cases} \tag{68}$$

Four-point method with memory, see [50]:

$$\begin{cases} \text{Given } x_0,\ y_{-1},\ z_{-1},\ w_{-1}, \\ y_k = \Psi(x_k, y_{k-1}, z_{k-1}, w_{k-1}), \\ z_k = \Psi(x_k, y_k, z_{k-1}, w_{k-1}), \\ w_k = \Psi(x_k, y_k, z_k, w_{k-1}), \\ x_{k+1} = \Psi(x_k, y_k, z_k, w_k), \end{cases} \tag{69}$$

where

$$\begin{aligned} \Psi(x,y,z,w) = N(x) + [&f(y)f(z)(f(y)-f(z))\Phi(w) + f(y)f(w)(f(w)-f(y))\Phi(z) \\ &- f(w)f(z)(f(w)-f(z))\Phi(y)]\frac{f(x)^2}{(f(w)-f(y))(f(w)-f(z))(f(y)-f(z))}. \end{aligned}$$

According to Herzberger's theorem, the associated matrices corresponding to the method (68) have the form

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

so that

$$M^{(3)} = M_3 \cdot M_2 \cdot M_1 = \begin{bmatrix} 8 & 3 & 2 \\ 4 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix}.$$

The associated matrices concerned with the method (69) are of the form

$$M_4 = \begin{bmatrix} 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad M_3 = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad M_1 = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

and hence

$$M^{(4)} = M_4 \cdot M_3 \cdot M_2 \cdot M_1 = \begin{bmatrix} 16 & 7 & 6 & 4 \\ 8 & 4 & 3 & 2 \\ 4 & 2 & 2 & 1 \\ 2 & 1 & 1 & 1 \end{bmatrix}.$$

The spectral radii of the resulting matrices $M^{(3)}$ and $M^{(4)}$ are ≈ 10.131 and ≈ 21.690, which gives the correct values of the R-order of convergence of the methods (68) and (69), respectively.

Remark 6. Since the form of all involved matrices is correct, we note that the correction of wrong results in the papers [39,50] is pretty obvious: matrices $M_1, \cdots, M_s$ (for $s = 2, 3, 4$ in the considered cases) should be multiplied in the order $M_s \cdot M_{s-1} \ldots M_1$, not in reverse order as was done.

Remark 7. The three-point methods with memory, considered by Wang, Džunić and Zhang in [66], also deal with Herzberger's matrix method and apply this matrix method in a proper way.

The above-presented multipoint methods in this section use the first derivative. In the similar fashion, using divided differences and the formulae (56), we can construct derivative free methods that are variants with memory of the Kung–Traub family (72) described in the next section.

For illustration, we give two derivative free iterative methods. The iterative scheme with three function evaluations and two initial approximations $(x_0,\ z_{-1})$ has the form

$$\begin{cases} y_k = x_k - f^{-1}[f(x_k), f(z_{k-1})]f(x_k) = x_k - \frac{f(x_k)(f(x_k)-f(z_{k-1}))}{x_k - z_{k-1}}, \\ z_k = x_k - f^{-1}[f(x_k), f(y_k)]f(x_k), \\ x_{k+1} = z_k + f^{-1}[f(x_k), f(y_k), f(z_k)]f(x_k)f(y_k). \end{cases} \tag{70}$$

The resulting matrix is the product of three matrices associated to $x_{k+1},\ z_k$, and y_k and reads

$$M^{(3)}(x_{k+1}, z_k, y_k) = \begin{bmatrix} 4 & 2 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Its spectral radius $\rho(M^{(3)}) = 5$ determines the order of the multipoint method (70).

The following iterative scheme with four function evaluations per iteration and three initial values $(x_0,\ y_{-1},\ z_{-1})$ can be constructed:

$$\begin{cases} w_k = x_k - f^{-1}[f(x_k), f(z_{k-1})]f(x_k) + f^{-1}[f(x_k), f(z_{k-1}), f(y_{k-1})]f(x_k)f(z_{k-1}), \\ y_k = x_k - f^{-1}[f(x_k), f(w_k)]f(x_k), \\ z_k = y_k + f^{-1}[f(x_k), f(w_k), f(y_k)]f(x_k)f(w_k) \\ x_{k+1} = z_k + f^{-1}[f(x_k), f(w_k), f(y_k), f(z_k)]f(x_k)f(w_k)f(y_k). \end{cases} \tag{71}$$

The resulting matrix is the product of four matrices associated to $x_{k+1},\ z_k,\ y_k$ and w_k and has the form

$$M^{(4)}(x_{k+1}, z_k, y_k, w_k) = \begin{bmatrix} 8 & 4 & 4 & 0 & 0 \\ 4 & 2 & 2 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Spectral radius of this matrix is $\rho(M^{(4)}) = 11$ so that the order of the multipoint method (71) is 11.

In the following section we will show an efficient way for accelerating derivative free methods.

8. Generalized multipoint methods with memory

In this section we study multipoint methods with memory based on multipoint methods of arbitrary order of convergence as presented in [15]. We restrict our attention to the Kung–Traub family [32] and the Zheng–Li–Huang family [73] for the following reasons:

(1) both families of n-point methods have similar structure, the order 2^n and require $n+1$ function evaluations per iteration, which means that they generate optimal methods in the sense of the Kung–Traub conjecture;
(2) both families represent examples of general interpolatory iteration functions as defined in [71];
(3) these families do not deal with derivatives, which is convenient in all situations when the calculation of derivatives of f is complicated.

As shown in [15], both families can be represented in a unique form. This unique representation facilitates in carrying the convergence analysis of both families simultaneously. These families are modified by a specific approach as to give very efficient generalized methods with memory.

Kung and Traub (1974) stated in [32] the following derivative free family (K–T for short) of iterative methods without memory.

K–T family: For an initial approximation x_0, arbitrary $n \in \boldsymbol{N}$ and $k = 0, 1, \ldots$, define the iteration function $\psi_j(f)$ $(j = -1, 0, \ldots, n)$ as follows:

$$\begin{cases} y_{k,0} = \psi_0(f)(x_k) = x_k, \quad y_{k,-1} = \psi_{-1}(f)(x_k) = x_k + \gamma_k f(x_k), \quad \gamma_k \in \boldsymbol{R} \setminus \{0\}, \\ y_{k,j} = \psi_j(f)(x_k) = \mathcal{R}_j(0), \quad j = 1, \ldots, n, \ \text{for } n > 0, \\ x_{k+1} = y_{k,n} = \psi_n(f)(x_k), \end{cases} \tag{72}$$

where $\mathcal{R}_j(\tau)$ represents an inverse interpolatory polynomial of degree no greater then j such that

$$\mathcal{R}_j(f(y_{k,m})) = y_{k,m}, \quad m = -1, 0, \ldots, j-1.$$

Zheng, Li and Huang proposed in [73] other derivative free family (Z–L–H for short) of n-point methods of arbitrary order of convergence 2^n $(n \geqslant 1)$. This family is constructed using Newton's interpolation with forward divided differences. Equating the error factor $R_{j,k}$, which originally appears in [73], to 0, the simplified Z–L–H family gets the following form.

Z–L–H family: For an initial approximation x_0, arbitrary $n \in \boldsymbol{N}, \gamma_k \in \boldsymbol{R} \setminus \{0\}$ and $k = 0, 1, \ldots$, the n-point method is defined by

$$\begin{cases} y_{k,0} = x_k, \quad y_{k,-1} = y_{k,0} + \gamma_k f(y_{k,0}), \\ y_{k,1} = y_{k,0} - \frac{f(y_{k,0})}{f[y_{k,0}, y_{k,-1}]}, \\ y_{k,2} = y_{k,1} - \frac{f(y_{k,1})}{f[y_{k,1}, y_{k,0}] + f[y_{k,1}, y_{k,0}, y_{k,-1}](y_{k,1} - y_{k,0})}, \\ \vdots \\ y_{k,n} = y_{k,n-1} - \frac{f(y_{k,n-1})}{f[y_{k,n-1}, y_{k,n-2}] + \sum_{j=1}^{n-1} f[y_{k,n-1}, \ldots, y_{k,n-2-j}] \prod_{i=1}^{j} (y_{k,n-1} - y_{k,n-1-i})}, \\ x_{k+1} = y_{k,n}. \end{cases} \tag{73}$$

In what follows, if the parameter γ_k in (72) and (73) is a constant, we will put $\gamma_k = \gamma$. Assuming that a real parameter γ_k in the above families (72) and (73) has a constant value, as done in [32,73], the order of convergence of the families $(72)_{\gamma_k=\gamma}$ and $(73)_{\gamma_k=\gamma}$ is 2^n. Since these families require $n+1$ function evaluations, they are optimal.

Now we will show that the Kung–Traub family $(72)_{\gamma_k=\gamma}$ and the Zheng–Li–Huang family $(73)_{\gamma_k=\gamma}$ can be extremely accelerated without any additional function evaluations. The construction of new families of n-point derivative free methods is based on the variation of a free parameter γ_k in each iterative step. This parameter is calculated using information from the current and previous iteration so that the presented methods may be regarded as methods with memory.

The error relations concerning the families $(72)_{\gamma_k=\gamma}$ and $(73)_{\gamma_k=\gamma}$ can be presented in the unified form (see [15], [53, Ch. 6])

$$\varepsilon_{k,-1} \sim (1 + \gamma_k f'(\alpha))\varepsilon_k, \quad \varepsilon_{k,j} \sim a_{k,j}(1 + \gamma_k f'(\alpha))^{2^{j-1}} \varepsilon_k^{2^j} \quad (j = 1, \ldots, n), \tag{74}$$

where

$$\varepsilon_k = y_{k,0} - \alpha = x_k - \alpha, \quad \varepsilon_{k,j} = y_{k,j} - \alpha \quad (j = -1, 0, 1, \ldots, n),$$

k being the iteration index. Constants $a_{k,j}$ depend on the considered families and they were given in the papers [32,73], see also [53, Ch. 6]. The use of the unique relation (74) enables us to construct and analyze simultaneously both families with memory based on $(72)_{\gamma_k=\gamma}$ and $(73)_{\gamma_k=\gamma}$. Let us note that (74) also gives the common final error relation

$$\varepsilon_{k+1} = \varepsilon_{k,n} = y_{k,n} - \alpha \sim a_{k,n}(1+\gamma_k f'(\alpha))^{2^{n-1}} \varepsilon_k^{2^n}. \tag{75}$$

As mentioned in [15,50,51], the factor $1+\gamma_k f'(\alpha)$ in the error relation (75) plays the key role in constructing families with memory.

We observe from (75) that the order of convergence of the families $(72)_{\gamma_k=\gamma}$ and $(73)_{\gamma_k=\gamma}$ is 2^n when γ_k is not close to $-1/f'(\alpha)$. It is not difficult to show that the order of these families would be $2^n + 2^{n-1}$ if we could provide $\gamma_k = -1/f'(\alpha)$. However, the value $f'(\alpha)$ is not known in practice and we could use only an approximation $\widetilde{f}'(\alpha) \approx f'(\alpha)$, calculated based on available information. Then, setting $\gamma_k = -1/\widetilde{f}'(\alpha)$, we achieve order of convergence of the modified methods exceeding 2^n without using any new function evaluations.

The beneficial approach in approximating

$$\gamma_k = -1/\widetilde{f}'(\alpha) \approx -1/f'(\alpha)$$

is to use only available information, in other words, we can increase the convergence rate without additional computational cost. We present the following model for approximating $f'(\alpha)$:

$$\widetilde{f}'(\alpha) = N_m'(y_{k,0}) \quad \text{(Newton's interpolation with divided differences)},$$

where

$$N_m(\tau) = N_m(\tau; y_{k,0}, y_{k-1,j_1}, \ldots, y_{k-1,j_m}), \quad -1 \leqslant j_m < j_{m-1} < \cdots < j_1 \leqslant n-1 \tag{76}$$

represents Newton's interpolating polynomial of degree m $(1 \leqslant m \leqslant n-1)$, set through $m+1$ available approximations (nodes) $y_{k,0}, y_{k-1,j_1}, \ldots, y_{k-1,j_m}$. Then the formula for calculation γ_k is given by:

$$\gamma_k = -\frac{1}{N_m'(y_{k,0})} \approx -\frac{1}{f'(\alpha)}. \tag{77}$$

Let $I_m = \{y_{k,0}, y_{k-1,j_1}, \ldots, y_{k-1,j_m}\}$ denote the set of interpolation nodes. Substituting the fixed parameter γ_k in the iterative formulae $(72)_{\gamma_k=\gamma}$ and $(73)_{\gamma_k=\gamma}$ by the varying parameter γ_k calculated by (77), we state the families of multipoint methods with memory given by (72) and (73). For example, as it was done in [15], for $m = 1,2,3$, from (77) we obtain

$$N_1'(y_{k,0}) = \frac{f(y_{k,0}) - f(y_{k-1,n-1})}{y_{k,0} - y_{k-1,n-1}}, \tag{78}$$

$$N_2'(y_{k,0}) = f[y_{k,0}, y_{k-1,n-1}] + f[y_{k,0}, y_{k-1,n-1}, y_{k-1,n-2}](y_{k,0} - y_{k-1,n-1}), \tag{79}$$

$$\begin{aligned} N_3'(y_{k,0}) = {} & f[y_{k,0}, y_{k-1,n-1}] + f[y_{k,0}, y_{k-1,n-1}, y_{k-1,n-2}](y_{k,0} - y_{k-1,n-1}) + f[y_{k,0}, y_{k-1,n-1}, y_{k-1,n-2}, y_{k-1,n-3}](y_{k,0} - y_{k-1,n-1}) \\ & \times (y_{k,0} - y_{k-1,n-2}). \end{aligned} \tag{80}$$

Note that (78) is, actually, secant method applied by Traub [63, p. 186] for constructing an accelerating method with memory of order $1+\sqrt{2}$.

It is obvious that the Zheng–Li–Huang family $(73)_{\gamma_k=\gamma}$ is very suitable for applying Newton's interpolating approaches (79) and (80) since divided differences are already calculated in the implementation of the iterative scheme $(73)_{\gamma_k=\gamma}$. The use of Newton's interpolation of higher order is also feasible but it requires increased number of steps in the iterative scheme, which is not of interest for solving most practical problems.

In what follows we give a condensed form of the results concerning the order of convergence of the described generalized families with memory (72) and (73). Note that these results are summarized from the assertions given in [13,15,53].

First we give an important lemma proved in [15], recalling that interpolation nodes are indexed as in (76).

Lemma 1. *Let $N_m(t)$ be Newton's interpolating polynomial of degree m that interpolates a given function f at $m+1$ distinct interpolation nodes $y_{k,0}, y_{k-1,1}, \ldots, y_{k-1,m} \in I_m$, contained in a neighborhood V_f of a zero α of f. Let the derivative $f^{(m+1)}$ be continuous in V_f. Define the differences $\varepsilon_{k-1,j} = y_{k-1,j} - \alpha$ $(j \in \{1, \ldots, m\})$, $\varepsilon_k = y_{k,0} - \alpha$ and assume*

(1) *all nodes $y_{k,0}, y_{k-1,n-1}, \ldots, y_{k-1,n-m}$ are sufficiently close to the zero α;*
(2) *the condition $\varepsilon_{k,0} = o(\varepsilon_{k-1,1} \ldots \varepsilon_{k-1,m})$ holds when $k \to \infty$.*

Then

$$N'_m(y_{k,0}) \sim f'(\alpha)\left(1 + (-1)^{m+1}c_{m+1}\prod_{j=1}^{m}\varepsilon_{k-1,j}\right), \quad c_{m+1} = \frac{f^{(m+1)}(\alpha)}{(m+1)!f'(\alpha)}. \tag{81}$$

We distinguish convergence analysis of the methods (72) and (73) with memory to the following three cases, depending on the use of approximations $y_{k-1,0}$ and $y_{k-1,-1}$.

Method I: $j_m > 0$, that is, $y_{k-1,0}, y_{k-1,-1} \notin I_m$.
According to (81) given in Lemma 1, we have

$$N'_m(y_{k,0}) \sim f'(\alpha)\left(1 + (-1)^{m+1}c_{m+1}\prod_{i=1}^{m}\varepsilon_{k-1,j_i}\right),$$

that is (in view of (77))

$$1 + \gamma_k f'(\alpha) \sim (-1)^{m+1}c_{m+1}\prod_{i=1}^{m}\varepsilon_{k-1,j_i}. \tag{82}$$

Assuming that

$$\varepsilon_{k+1} \sim A_{k,n}\varepsilon_k^r \quad \text{and} \quad \varepsilon_{k,j} \sim A_{k,j}\varepsilon_k^{r_j} \tag{83}$$

and using (83), we can derive the following relations (see [15] for more details)

$$\varepsilon_{k+1} \sim A_{k,n}\varepsilon_k^r \sim A_{k,n}A_{k-1,n}^r\varepsilon_{k-1}^{r^2}, \tag{84}$$

$$\varepsilon_{k,j_s} \sim A_{k,j_s}\varepsilon_k^{r_{j_s}} \sim A_{k,j_s}A_{k-1,n}^{r_{j_s}}\varepsilon_{k-1}^{rr_{j_s}}, \quad 1 \leqslant s \leqslant m. \tag{85}$$

Combining (74) and (82)–(85) we obtain error relations in a general form

$$\varepsilon_{k+1} \sim a_{k,n}c_{m+1}^{2^{n-1}}A_{k-1,n}^{2^n}\left(\prod_{i=1}^{m}A_{k-1,j_i}\right)^{2^{n-1}}\varepsilon_{k-1}^{2^n r + 2^{n-1}(r_{j_1}+\cdots+r_{j_m})}, \tag{86}$$

$$\varepsilon_{k,j_s} \sim a_{k,j_s}c_{m+1}^{2^{j_s-1}}A_{k-1,n}^{2^{j_s}}\left(\prod_{i=1}^{m}A_{k-1,j_i}\right)^{2^{j_s-1}}\varepsilon_{k-1}^{2^{j_s}r + 2^{j_s-1}(r_{j_1}+\cdots+r_{j_m})}, \tag{87}$$

for $1 \leqslant s \leqslant m$. Equating exponents of ε_{k-1} in pairs of relations (84)∧(86), and (85)∧(87) for each $1 \leqslant s \leqslant m$, we arrive at the following system of $m+1$ equations

$$\begin{cases} r^2 - 2^n r - 2^{n-1}(r_{j_1} + \cdots + r_{j_m}) = 0, \\ rr_{j_s} - 2^{j_s}r - 2^{j_s-1}(r_{j_1} + \cdots + r_{j_m}) = 0, \quad 1 \leqslant s \leqslant m, \end{cases} \tag{88}$$

in the unknowns $r, r_{j_1}, \ldots, r_{j_m}$. Solving this system we obtain $r_{j_i} = 2^{j_i-n}r$ which reduces (88) to the quadratic equation

$$r^2 - r\left(2^n + \sum_{i=1}^{m}2^{j_i-1}\right) = 0.$$

Its positive solution gives the sought order of convergence

$$r = 2^n + \sum_{i=1}^{m}2^{j_i-1}. \tag{89}$$

In view of (89) we observe that maximal order of convergence, for a given fixed degree m of the polynomial N_m, is attained taking maximal j_i, in other words, using the best attainable approximations $y_{k,0}, y_{k-1,n-1}, \ldots, y_{k-1,n-m}$. In this case order of convergence equals

$$r = 2^n + \sum_{i=1}^{m}2^{n-i-1} = \begin{cases} 2^n + 2^{n-1} - 2^{n-m-1}, & m > 1 \\ 2^n + 2^{n-2}, & m = 1. \end{cases} \tag{90}$$

According to (89) or (90), Method I attains the highest order for the highest possible degree $m = n-1$. Then $r = 2^n + 2^{n-1} - 1$.

Method II: $j_m = 0$, that is, $y_{k-1,0} \in I_m \wedge y_{k-1,-1} \notin I_m$.

By virtue of (81), in this case the following is valid:

$$N'_m(y_{k,0}) \sim f'(\alpha)\left(1 + (-1)^{m+1} c_{m+1}\varepsilon_{k-1}\prod_{i=1}^{m-1}\varepsilon_{k-1,j_i}\right),$$

that is (in view of (77)),

$$1 + \gamma_k f'(\alpha) \sim (-1)^{m+1} c_{m+1}\varepsilon_{k-1}\prod_{i=1}^{m-1}\varepsilon_{k-1,j_i}. \tag{91}$$

Relation (84) is still valid, while the number of relations in (85) is reduced by one ($r_m = 1$ is not unknown since $\varepsilon_{k-1,j_m} = \varepsilon_{k-1}$) and reads

$$\varepsilon_{k,j_s} \sim A_{k,j_s}\varepsilon_k^{r_{j_s}} \sim A_{k,j_s}(A_{k-1,n})^{r_{j_s}}\varepsilon_{k-1}^{rr_{j_s}}, \quad 1 \leqslant s \leqslant m-1. \tag{92}$$

Combining (74), (84), (91) and (92), in a similar way as for Method I we find first the errors ε_{k+1} and ε_{k,j_s} $(1 \leqslant s \leqslant m-1)$. Then we form the corresponding system of equations in the unknowns $r,\ r_{j_1},\ldots,r_{j_m}$ that gives the order of convergence

$$r = 2^{n-1} + \sum_{i=1}^{m-1} 2^{j_i-2} + \sqrt{\left(2^{n-1} + \sum_{i=1}^{m-1} 2^{j_i-2}\right)^2 + 2^{n-1}}. \tag{93}$$

Remark 8. Note that Traub's basic secant accelerating technique is included for $m = 1$. Then the order of convergence of the method with memory equals $r = 2^{n-1} + \sqrt{2^{2(n-1)} + 2^{n-1}}$. In particular, for $n = 1$ and $m = 1$ the accelerated Traub–Steffensen method with order $1 + \sqrt{2}$ is obtained, see [63, p. 186].

Remark 9. Maximal acceleration by Method II is attained taking $m = n$; the order of convergence is then $r = \frac{1}{2}\left(2^n + 2^{n-1} - 1 + \sqrt{9 \cdot 2^{2(n-1)} - 2^n + 1}\right)$.

Method III $j_m = -1$, that is, $y_{k-1,-1} \in I_m$.

We will distinguish two subcases when $j_{m-1} = 0$ and $j_{m-1} > 0$. From (81) two estimates follow:

$$\begin{aligned}
&\text{(a)}\quad 1 + \gamma_k f'(\alpha) \sim (-1)^{m+1} c_{m+1}\varepsilon_{k-1}\varepsilon_{k-1,-1}\prod_{i=1}^{m-2}\varepsilon_{k-1,j_i} \quad (y_{k-1,0} \in I_m),\\
&\text{(b)}\quad 1 + \gamma_k f'(\alpha) \sim (-1)^{m+1} c_{m+1}\varepsilon_{k-1,-1}\prod_{i=1}^{m-1}\varepsilon_{k-1,j_i} \quad (y_{k-1,0} \notin I_m).
\end{aligned} \tag{94}$$

Aside from (74), we also need the estimate

$$\varepsilon_{k,-1} \sim A_{k,-1}\varepsilon_k^{r_{j_m}} \sim A_{k,-1}A_{k-1,n}^{r_{j_m}}\varepsilon_{k-1}^{rr_{j_m}}. \tag{95}$$

Case (a): If $j_{m-1} = 0$, the next $m-2$ estimates are relevant ($r_{j_{m-1}} = 1$)

$$\varepsilon_{k,j_s} \sim A_{k,j_s}\varepsilon_k^{r_{j_s}} \sim A_{k,j_s}(A_{k-1,n})^{r_{j_s}}\varepsilon_{k-1}^{rr_{j_s}}, \quad 1 \leqslant s \leqslant m-2. \tag{96}$$

Then combining (94a), (84), (95) and (96), in a similar way as above we form the system of equations in the unknown $r,\ r_{j_1},\ldots,r_{j_m}$ that gives the order of convergence

$$r = 2^n + 1 + \sum_{i=1}^{m-2} 2^{j_i-1}. \tag{97}$$

The greatest acceleration is attained for $m = n+1$, that is, when all approximations from the previous iteration are used. In this case the order is $r = 2^n + 2^{n-1}$. For example, starting from Traub–Steffensen's method (6) ($n = 1$), we obtain for $m = 2$ the accelerated method with memory with order 3.

Case (b): If $j_{m-1} > 0$, then using analogous procedure and the relations (74), (84), (92) and (94b) we obtain the order of convergence

$$r = 2^{n-1} + \sum_{i=1}^{m-1} 2^{j_i-2} + \frac{1}{2} + \sqrt{\left(2^{n-1} + \sum_{i=1}^{m-1} 2^{j_i-2} + \frac{1}{2}\right)^2 - 2^{n-1}}. \tag{98}$$

This case is of less importance than (a) since the node $y_{k-1,0}$ is not taken into account. However, the interpolating polynomial $N_m(t; y_k, 0, y_{k,j_1}, \ldots, y_{k,j_{m-1}}, y_{k,0})$ gives worse accelerating results than the polynomial of the same degree $N_m(t; y_k, 0, y_{k,j_1}, \ldots, y_{k,j_{m-1}}, y_{k,-1})$.

The highest order is obtained for $m = n$ and it is equal to

$$r = 2^{n-1} + 2^{n-2} + \sqrt{2^{2n-1} + 2^{2n-4} - 2^{n-1}}.$$

For example, the two-point method with memory ($n = 2$) has the order $r = 3 + \sqrt{7} \approx 5.646$, while the three-point method with memory ($n = 3$) has the order $r = 6 + 4\sqrt{2} \approx 11.657$.

From Table 1 we observe that the order of convergence of the families (72) and (73) with memory is considerably increased relative to the corresponding basic families without memory (entries in the last row). The increment in percentage is also displayed and we can see that the improvement of the order is up to 50%. It is worth noting that the improvement of convergence order in all cases is attained without any additional function evaluations, which points to a very high computational efficiency of the proposed methods with memory. Several values of the efficiency index

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

where r is the order of the considered iterative method (IM) and θ_f is the number of function evaluations per iteration, are given in Table 2.

We end this section with a remark that recent investigations presented in [16] have shown that further acceleration of generalized multipoint methods can be attained by constructing biparametric multipoint methods. The increase of convergence order of this kind of methods with memory is up to 75% (that is, 1.75×2^n) relative to the corresponding methods $(72)_{\gamma_k=\gamma}$ and $(73)_{\gamma_k=\gamma}$ without memory. This improvement is attained using available data only from the current and previous iteration. The biparametric multipoint methods have the form

$$\begin{cases} y_{k,1} = \phi_1(f)(x_k) = x_k + \gamma f(x_k), \\ y_{k,2} = \phi_2(f)(x_k) = x_k - \frac{f(x_k)}{f[x_k, y_{k,1}] + pf(y_{k,1})}, \\ y_{k,j} = \phi_j(f)(x_k), \quad j = 3, \ldots, n, \\ x_{k+1} = y_{k,n+1} = \phi_{n+1}(f)(x_k), \quad k = 0, 1, \ldots, \end{cases} \tag{99}$$

where $\gamma \neq 0$ and p are real parameters, see [16]. The first two steps of the iterative scheme (99) define the two-parameter Steffensen-like method

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, x_k + \gamma f(x_k)] + pf(x_k + \gamma f(x_k))}, \quad k = 0, 1, \ldots. \tag{100}$$

The next $n - 1$ steps $y_{k,j} = \phi_j(f)(x_k)$, $j = 3, \ldots, n + 1$, use interpolatory iteration functions

$$y_{k,j} = \phi_j(f)(x_k) = \phi_j(y_{k,0}, y_{k,1}, \ldots, y_{k,j-1}).$$

For more details on interpolatory iteration functions see the book [63, Ch. 4]. The order of convergence of the n-point method without memory (99) is 2^n, assuming that γ and p are constants.

Remark 10. As shown in [16], for some concrete two- or three-point methods it is possible to choose certain suitable functions (involving weight functions or approximations of derivatives, for example) instead of interpolatory iteration functions. See the example presented at the end of this paper.

It is not difficult to show that the error relation of Steffensen-like method (100) is given by

$$\varepsilon_{k+1} \sim (c_2 + p)(1 + \gamma f'(\alpha))\varepsilon_k^2,$$

Table 1
The lower bounds of the convergence order given in bold.

$n \rightarrow$	1	2	3	4
$m = 1$				
$j = 0$	**2.414** (20.7%)	**4.449** (11.2%)	**8.472** (6%)	**16.485** (3%)
$j = 1$		**5** (25%)	**9** (12.5%)	**17** (6.25%)
$j = 2$			**10** (25%)	**18** (12.5%)
$j = 3$				**20** (25%)
$m = 2$	**3** (50%)	**5.372** (34%)	**11** (37.5%)	**22** (37.5%)
$m = 3$		**6** (50%)	**11.35** (41.9%)	**23** (43.7%)
without memory	**2**	**4**	**8**	**16**

Table 2
The efficiency indices of multipoint methods with/without memory.

n	N_1				N_2	N_3	without memory
	$j=0$	$j=1$	$j=2$	$j=3$			
1	1.554				1.732		1.414
2	1.645	1.710			1.751	1.817	1.587
3	1.706	1.732	1.778		1.821	1.836	1.682
4	1.759	1.762	1.783	1.820	1.856	1.872	1.741

where $\varepsilon_k = x_k - \alpha$. This error relation has a key role in accelerating convergence order of the multipoint method with memory since its error relation contains $(c_2+p)(1+\gamma f'(\alpha))$ as a factor. Using a suitable calculation of the parameters p and γ to minimize the factors c_2+p and $1+\gamma f'(\alpha)$, we considerably increase the convergence rate of the accelerated method.

The presented model for approximating $f'(\alpha)$ and c_2 uses Newton's interpolation with divided differences

$$\widetilde{f}'(\alpha) = N'_m(y_{k,0}), \quad \text{and} \quad \widetilde{c}_2 = \frac{N''_{m+1}(y_{k,1})}{2N'_{m+1}(y_{k,1})}.$$

Here

$$N_m(\tau) = N_m(\tau; y_{k,0}, y_{k-1,n-j_1}, \ldots, y_{k-1,n-j_m}),$$
$$N_{m+1}(\tau) = N_{m+1}(\tau; y_{k,1}, y_{k,0}, y_{k-1,n-j_1}, \ldots, y_{k-1,n-j_m}),\ 0 \leqslant j_1 < j_2 < \cdots < j_m \leqslant n,$$

are Newton's interpolating polynomials set through $m+1$ and $m+2$ available approximations from the current and previous iteration. Obviously, the fastest acceleration is achieved when best available approximations are used as nodes for Newton's interpolating polynomials giving

$$N_m(\tau) = N_m(\tau; y_{k,0}, y_{k-1,n}, \ldots, y_{k-1,n-m+1}), \tag{101}$$

$$N_{m+1}(\tau) = N_{m+1}(\tau; y_{k,1}, y_{k,0}, y_{k-1,n}, \ldots, y_{k-1,n-m+1}). \tag{102}$$

for $m \leqslant n+1$. Hence, the formulae for calculating γ_k and p_k are given by

$$\gamma_k = -\frac{1}{N'_m(y_{k,0})}, \quad m \geqslant 1, \tag{103}$$

$$p_k = -\frac{N''_{m+1}(y_{k,1})}{2N'_{m+1}(y_{k,1})}, \quad m \geqslant 1, \tag{104}$$

where N_m and N_{m+1} are defined by (101) and (102), respectively.

Substituting constant parameters γ and p in the iterative formula (99) by the varying γ_k and p_k defined by (103) and (104), we construct the family of n-point methods with memory

$$\begin{cases} y_{k,1} = x_k + \gamma_k f(x_k), \\ y_{k,2} = x_k - \frac{f(x_k)}{f[x_k, y_{k,1}] + p_k f(y_{k,1})}, \\ y_{k,j} = \phi_j(f)(x_k), \quad j = 3, \ldots, n, \\ x_{k+1} = y_{k,n+1} = \phi_{n+1}(f)(x_k), \quad k = 0, 1, \ldots. \end{cases} \tag{105}$$

The following theorem has been proved in [16].

Theorem 8. *Let x_0 be an initial approximation sufficiently close to a simple zero α of a function f. Then the convergence order of the family of n-point methods ($n \geqslant 2$) with memory (105) with the varying γ_k and p_k, calculated by (103) and (104), is given by*

$$r = \begin{cases} 2^n + 2^{n-1} + 2^{n-2} - 3\cdot 2^{n-m-2} = 2^{n-m-2}(7\cdot 2^m - 3), & 1 \leqslant m < n, \\ 7\cdot 2^{n-3} + 2^{\frac{n}{2}-3}\sqrt{49\cdot 2^n - 48}, & m = n, \\ 2^n + 2^{n-1} + 2^{n-2} = 1.75\cdot 2^n, & m = n+1,\ n \geqslant 2. \end{cases} \tag{106}$$

for $1 \leqslant m \leqslant n+1$.

We observe from the third formula of (106) that the improvement of convergence order of the family with memory (105) is up to 75% related to the order of the method without memory (99). This improvement is attained using only available data from the current and previous iteration.

We end this paper with a particular example of biparametric's type. Let us consider the two-point family without memory,

$$\begin{cases} y_{k,2} = x_k - \frac{f(x_k)}{f[x_k,y_{k,1}]+pf(y_{k,1})}, & y_{k,1} = x_k + \gamma f(x_k), \\ x_{k+1} = y_{k,2} - g(u_k)\frac{f(y_{k,2})}{f[y_{k,2},y_{k,1}]+pf(y_{k,1})}, & u_k = \frac{f(y_{k,2})}{f(x_k)}, \end{cases} \quad (k = 0, 1, \ldots), \tag{107}$$

where g is a weight function (compare to the two-point method (31)). Assuming that the parameters γ and p are constant, this family reaches the optimal order four under the conditions

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

Varying parameters γ and p in (107) using (103) and (104), we obtain the family of two-point methods with memory of order 7, which is the improvement of the convergence rate of 75%. Some examples of the weight function g of simple form are given below,

$$g(u) = 1 + u, \quad g(u) = \frac{1}{1-u}, \quad g(u) = \frac{1}{1-u-u^2}.$$

Acknowledgments

The authors are thankful to Dr. Melvin Scott for his helpful comments which improve the presentation.

References

[1] S. Amat, S. Busquier, S. Plaza, Review of some iterative root-finding methods from a dynamical point of view, Scientia 10 (2004) 3–35.

[2] S. Amat, S. Busquier, S. Plaza, Dynamics of a family of third-order iterative methods that do not require using second derivatives, Appl. Math. Comput. 154 (2004) 735–746.

[3] S. Amat, S. Busquier, S. Plaza, Dynamics of the King and Jarratt iterations, Aequationes Mathematicae 69 (2005) 212–2236.

[4] D. Basu, From third to fourth order variant of Newton's method for simple roots, Appl. Math. Comput. 202 (2008) 886–892.

[5] J.-P. Berrut, Rational functions for guaranteed and experimentally well-conditioned global interpolation, Comput. Math. Appl. 15 (1988) 1–16.

[6] J.-P. Berrut, L.N. Trefethen, Barycentric Lagrange interpolation, SIAM Rev. 46 (2004) 501–517.

[7] R. Brent, P. Zimmermann, Modern Computer Arithmetic, Cambridge University Press, Cambridge, 2011.

[8] R. Brent, S. Winograd, P. Wolfe, Optimal iterative processes for root-finding, Numer. Math. 20 (1973) 327–341.

[9] C. Chun, Some variants of King's fourth-order family of methods for nonlinear equations, Appl. Math. Comput. 190 (2007) 57–62.

[10] C. Chun, Some fourth-order iterative methods for solving nonlinear equations, Appl. Math. Comput. 195 (2008) 454–459.

[11] C. Chun, Y. Ham, Some fourth-order modifications of Newton's method, Appl. Math. Comput. 197 (2008) 654–658.

[12] C. Chun, M.Y. Lee, B. Neta, J. Džunić, On optimal fourth-order iterative methods free from second derivative and their dynamics, Appl. Math. Comput. 218 (2012) 6427–6438.

[13] J. Džunić, Multistep methods for solving nonlinear equations, Ph.D. Thesis (in Serbian), University of Niš, Niš, 2012.

[14] J. Džunić, On efficient two-parameter methods for solving nonlinear equations, Numer. Algorithms 63 (2013) 549–569.

[15] J. Džunić, M.S. Petković, On generalized multipoint root-solvers with memory, J. Comput. Appl. Math. 236 (2012) 2909–2920.

[16] J. Džunić, M.S. Petković, On generalized biparametric multipoint root finding methods with memory, J. Comput. Appl. Math. 255 (2014) 362–375.

[17] 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.

[18] 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.

[19] Y.H. Geum, Y.I. Kim, A uniparametric family of three-step eighth-order multipoint iterative methods for simple roots, Appl. Math. Lett. 24 (2011) 929–935.

[20] Y.H. Geum, Y.I. Kim, A biparametric family of eighth-order methods with their third-step weighting function decomposed into a one-variable linear fraction and a two-variable generic function, Comput. Math. Appl. 61 (2011) 708–714.

[21] J. Herzberger, Über Matrixdarstellungen für Iterationverfahren bei nichtlinearen Gleichungen, Computing 12 (1974) 215–222.

[22] P. Jarratt, Some fourth order multipoint methods for solving equations, Math. Comput. 20 (1966) 434–437.

[23] P. Jarratt, Some efficient fourth-order multipoint methods for solving equations, BIT 9 (1969) 119–124.

[24] L.O. Jay, A note on Q-order of convergence, BIT 41 (2001) 422–429.

[25] B. Kalantari, Polynomial Root Finding and Polynomiography, World Scientific Publishing Co., Singapore, 2009.

[26] R.F. King, A family of fourth order methods for nonlinear equations, SIAM J. Numer. Anal. 10 (1973) 876–879.

[27] J. Kou, Y. Li, X. Wang, A composite fourth-order iterative method for solving nonlinear equations, Appl. Math. Comput. 184 (2007) 471–475.

[28] J. Kou, Y. Li, X. Wang, Fourth-order iterative methods free from second derivative, Appl. Math. Comput. 184 (2007) 880–885.

[29] J. Kou, Y. Li, X. Wang, A family of fourth-order methods for solving nonlinear equations, Appl. Math. Comput. 188 (2007) 1031–1036.

[30] J. Kou, X. Wang, Y. Li, Some eighth-order root-finding three-step methods, Commun. Nonlinear Sci. Numer. Simul. 15 (2010) 536–544.

[31] H.T. Kung, J.F. Traub, Optimal order for iterations using two evaluations, Report, Computer Sci. Dep., Carnegie-Mellon U., Pittsburgh, 1973.

[32] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iteration, J. ACM 21 (1974) 643–651.

[33] S. Li, X. Liao, L. Cheng, A new fourth-order iterative method for finding multiple roots of nonlinear equations, Appl. Math. Comput. 215 (2009) 1288–1292.

[34] L. Liu, X. Wang, Eighth-order methods with high efficiency index for solving nonlinear equations, Appl. Math. Comput. 215 (2010) 3449–3454.

[35] Z. Liu, Q. Zheng, P. Zhao, A variant of Steffensen's method of fourth-order convergence and its applications, Appl. Math. Comput. 216 (2010) 1978–1983.

[36] A.K. Maheshwari, A fourth-order iterative method for solving nonlinear equations, Appl. Math. Comput. 211 (2009) 383–391.

[37] J.M. McNamee, Numerical Methods for Roots of Polynomials, Part I, Elsevier, Amsterdam, 2007.

[38] B. Neta, Numerical methods for the solution of equations, Net-A-Sof, California, 1983.

[39] B. Neta, A new family of higher order methods for solving equations, Int. J. Comput. Math. 14 (1983) 191–195.

[40] B. Neta, C. Chun, M. Scott, Choosing weight functions in iterative methods for simple roots (submitted for publication).

[41] B. Neta, C. Chun, M. Scott, Basins of attraction for optimal eight order methods to find simple roots of nonlinear equations (submitted for publication).

[42] B. Neta, M.S. Petković, Construction of optimal order nonlinear solvers using inverse interpolation, Appl. Math. Comput. 217 (2010) 2448–2455.

[43] B. Neta, M. Scott, C. Chun, Basin attractors for various methods for multiple roots, Appl. Math. Comput. 218 (2012) 5043–5066.
[44] B. Neta, M. Scott, C. Chun, Basins of attraction for several methods to find simple roots of nonlinear equations, Appl. Math. Comput. 218 (2012) 10548–10556.
[45] J.M. Ortega, W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
[46] A.M. Ostrowski, Solution of Equations and Systems of Equations, Academic Press, New York, 1960.
[47] Y. Peng, H. Feng, Q. Li, X. Zhang, A fourth-order derivative-free algorithm for nonlinear equations, J. Comput. Appl. Math. 235 (2011) 2551–2559.
[48] L.D. Petković, M.S. Petković, J. Džunić, A class of three-point root-solvers of optimal order of convergence, Appl. Math. Comput. 216 (2010) 671–676.
[49] M.S. Petković, On a general class of multipoint root-finding methods of high computational efficiency, SIAM J. Numer. Anal. 47 (2010) 4402–4414.
[50] M.S. Petković, J. Džunić, B. Neta, Interpolatory multipoint methods with memory for solving nonlinear equations, Appl. Math. Comput. 218 (2011) 2533–2541.
[51] 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.
[52] 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.
[53] M.S. Petković, B. Neta, L.D. Petković, J. Džunić, Multipoint Methods for Solving Nonlinear Equations, Elsevier, Amsterdam, 2013.
[54] 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.
[55] 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.
[56] M. Scott, B. Neta, C. Chun, Basin attractors for various methods, Appl. Math. Comput. 218 (2011) 2584–2599.
[57] J.R. Sharma, R.K. Goyal, Fourth-order derivative-free methods for solving nonlinear equations, Int. J. Comput. Math. 83 (2006) 101–106.
[58] J.R. Sharma, R.K. Guha, P. Gupta, Improved King's methods with optimal order of convergence based on rational approximations, Appl. Math. Lett. 26 (2013) 473–480.
[59] J.R. Sharma, R. Sharma, Modified Jarratt method for computing multiple roots, Appl. Math. Comput. 217 (2010) 878–881.
[60] I.F. Steffensen, Remarks on iteration, Skand. Aktuarietidskr. 16 (1933) 64–72.
[61] B.D. Stewart, Attractor basins of various root-finding methods, M.S. Thesis, Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA, June 2001.
[62] 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.
[63] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
[64] J.F. Traub, H. Wozniakowski, Strict lower and upper bounds on iterative computational complexity, Computer Science Department, Carnegie-Mellon University, Pittsburg, 1975, Paper 2232.
[65] J.L. Varona, Graphic and numerical comparison between iterative methods, Math. Intelligencer 24 (2002) 37–46.
[66] X. Wang, J. Džunić, T. Zhang, On an efficient family of derivative free three-point methods for solving nonlinear equations, Appl. Math. Comput. 219 (2012) 1749–1760.
[67] X. Wang, L. Liu, Modified Ostrowski's method with eighth-order convergence and high efficiency index, Appl. Math. Lett. 23 (2010) 549–554.
[68] X. Wang, L. Liu, New eighth-order iterative methods for solving nonlinear equations, J. Comput. Appl. Math. 234 (2010) 1611–1620.
[69] S. Weerakoon, S., T.G.I. Fernando, A variant of Newton's method with accelerated third-order convergence, Appl. Math. Lett. 13 (2000) 87–93.
[70] E.R. Vrscay, W.J. Gilbert, Extraneous fixed points, basin boundaries and chaotic dynamics for Schröder and König rational iteration functions, Numer. Math. 52 (1998) 1–16.
[71] H. Woźniakowski, Generalized information and maximal order of iteration for operator equations, SIAM J. Numer. Anal. 12 (1975) 121–135.
[72] H. Woźniakowski, Maximal order of multipoint iterations using *n* evaluations, in: J.F. Traub (Ed.), Analytic Computational Complexity, Academic Press, New York, 1976, pp. 75–107.
[73] Q. Zheng, J. Li, F. Huang, Optimal Steffensen-type families for solving nonlinear equations, Appl. Math. Comput. 217 (2011) 9592–9597.
[74] X. Zhou, X. Chen, Y. Song, Construction of higher order methods for multiple roots of nonlinear equations, J. Comput. Appl. Math. 235 (2011) 4199–4206.