

Research Article

On a Derivative-Free Variant of King's Family with Memory

M. Sharifi,¹ S. Karimi Vanani,¹ F. Khaksar Haghani,¹ M. Arab,¹ and S. Shateyi²

¹ Department of Mathematics, Islamic Azad University, Shahrekord Branch, Shahrekord, Iran

² Department of Mathematics and Applied Mathematics, University of Venda, Thohoyandou 0950, South Africa

Correspondence should be addressed to S. Shateyi; stanford.shateyi@univen.ac.za

Received 16 July 2014; Revised 29 August 2014; Accepted 1 September 2014

Academic Editor: Juan R. Torregrosa

Copyright © 2015 M. Sharifi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The aim of this paper is to construct a method with memory according to King's family of methods without memory for nonlinear equations. It is proved that the proposed method possesses higher R -order of convergence using the same number of functional evaluations as King's family. Numerical experiments are given to illustrate the performance of the constructed scheme.

1. Introduction

Many problems arising in diverse disciplines of mathematical sciences can be described by a nonlinear equation of the following form (see, e.g., [1]):

$$f(x) = 0, \quad (1)$$

where $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$ is a sufficiently differentiable function in a neighborhood D of a simple zero α of (1). If we are interested in approximating the root α , we can do it by means of an iterative fixed-point method in the following form:

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

provided that the starting point x_0 is given.

In this work, we are concerned with the fixed-point methods that generate sequences presumably convergent to the true solution of a given single smooth equation. These schemes can be divided into one-point and multipoint schemes. We remark that the one-point methods can possess 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 waste information that had already been used. This approach provides the construction of efficient iterative root-finding methods [2].

In such circumstance, special attention is devoted to multipoint methods with memory that use already computed

information to considerably increase convergence rate without additional computational costs. This would be the focus of this paper.

Traub in [2] proposed the following method with memory (TM):

$$w_k = x_k + \beta_k f(x_k), \quad \beta_k = -\frac{1}{f[x_k, x_{k-1}]}, \quad k = 0, 1, 2, \dots, \\ x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad (3)$$

with the order of convergence $1 + \sqrt{2}$.

The iterative methods with memory can improve the order of convergence of the without memory method without any additional functional calculations, and this results in a higher computational efficiency index. We remark that it is assumed that an initial approximation x_0 close enough to the sought simple zero and β_0 are given for iterative methods of type (3).

Recently, authors in [3] designed an approach to make derivative-free families with low complexity out of optimal methods. In fact, they conjectured that every time that one applies the approximation of the derivative $f'(x_n) \approx f[x_n, w_n]$, with $w_n = x_n + \beta f(x_n)^l$, on an optimal order $2q$, we will need $l \geq q$ for preserving the order of convergence.

For instance, choosing the well-known optimal two-step family of King (KM) [4],

$$y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots, \tag{4}$$

$$x_{k+1} = y_k - \frac{f(y_k)}{f'(x_k)} \frac{f(x_k) + \gamma f(y_k)}{f(x_k) + (\gamma - 2)f(y_k)}, \quad \gamma \in \mathbb{R},$$

and the conjecture of Cordero-Torregrosa, one may propose the following method (DKM):

$$y_k = x_k - \frac{f(x_k)}{\mathfrak{F}\mathfrak{D}}, \quad k = 0, 1, 2, \dots, \tag{5}$$

$$x_{k+1} = y_k - \frac{f(y_k)}{\mathfrak{F}\mathfrak{D}} \frac{f(x_k) + \gamma f(y_k)}{f(x_k) + (\gamma - 2)f(y_k)}, \quad \gamma \in \mathbb{R},$$

wherein

$$\mathfrak{F}\mathfrak{D} = \frac{f(x_k) - f(w_k)}{x_k - w_k}, \quad w_k = x_k + \beta f(x_k)^2, \tag{6}$$

$$\beta \in \mathbb{R} \setminus \{0\}.$$

In this work, we propose a two-step method with memory possessing a high efficiency index according to the well-known family of King's methods (5).

Our inspiration and motivation for constructing a higher-order method are linked in a direct manner with the fundamental concept of numerical analysis that any numerical method should give as accurate as possible output results with minimal computational cost. To state the matter differently, it is necessary to pursue methods of higher computational efficiency.

For more background concerning this topic, one may refer to [5, 6].

The paper is organized as follows. In Section 2, the aim of this paper is presented by contributing an iterative method with memory based on (5) for solving nonlinear equations. The proposed scheme is an extension over (4) and has a simple structure with an increased computational efficiency. In Section 3, we compare the theoretical results by applying the definition of efficiency index and further supports are furnished whereas numerical reports are stated. Some concluding remarks will be drawn in Section 4 to end the paper.

2. A New Method with Memory

In this section, we propose the following iterative method with memory based on (5):

$$y_k = x_k - \frac{f(x_k)}{\mathfrak{F}\mathfrak{D}}, \quad w_k = x_k + \beta_k f(x_k)^2, \quad k = 0, 1, 2, \dots, \tag{7}$$

$$x_{k+1} = y_k - \frac{f(y_k)}{\mathfrak{F}\mathfrak{D}} \frac{f(x_k) + \gamma f(y_k)}{f(x_k) + (\gamma - 2)f(y_k)}, \quad \gamma = -\frac{1}{2},$$

wherein the self-accelerating parameter is β_k . The error equation of (5) is ($\gamma = -1/2$)

$$e_{k+1} = -c_2 (f'(\alpha)^2 \beta c_2 + c_3) e_k^4 + O(e_k^5), \tag{8}$$

where $c_j = (1/j!)(f^{(j)}(\alpha)/f'(\alpha))$. We now must find a way so as to vanish the asymptotic error constant $\eta = -c_2(f'(\alpha)^2 \beta c_2 + c_3)$.

Toward this goal, one can increase the R -order by considering the following substitution:

$$\beta = -\frac{c_3}{f'(\alpha)^2 c_2}. \tag{9}$$

Since the zero is not known, relation (9) cannot be used in its exact form and we must approximate it recursively. This builds a variant with memory for King's family by using

$$\beta_k \approx -\frac{\bar{c}_3}{f'(\alpha)^2 \bar{c}_2}, \tag{10}$$

where $\bar{c}_j \approx c_j$. Now if we consider $N_3(t)$ to be Newton's interpolation polynomial of third degree set through four available approximations $x_k, x_{k-1}, y_{k-1}, w_{k-1}$ at the end of each cycle, we can propose the following new method with memory:

$$\beta_k = -\frac{N_3'''(x_k)}{3N_3'(x_k)^2 N_3''(x_k)}, \tag{11}$$

$$y_k = x_k - \frac{f(x_k)}{\mathfrak{F}\mathfrak{D}}, \quad w_k = x_k + \beta_k f(x_k)^2, \tag{11}$$

$$k = 0, 1, 2, \dots,$$

$$x_{k+1} = y_k - \frac{f(y_k)}{\mathfrak{F}\mathfrak{D}} \frac{f(x_k) - 1/2 f(y_k)}{f(x_k) - 5/2 f(y_k)}.$$

Note that, for example, we have the following formulation for the interpolating polynomial:

$$N_3'(x_k) = \left[\frac{d}{dt} N_3(t) \right]_{t=x_k} \tag{12}$$

$$= f[x_k, x_{k-1}] + f[x_k, x_{k-1}, y_{k-1}](x_k - x_{k-1})$$

$$+ f[x_k, x_{k-1}, y_{k-1}, w_{k-1}](x_k - x_{k-1})(x_k - y_{k-1}).$$

Acceleration in convergence for (11) is based on the use of a variation of one free nonzero parameter in each iterative step. This parameter is calculated using information from the current and previous iteration(s) so that the developed method may be regarded as method with memory according to Traub's classification [2].

We are at the time to write about the theoretical aspects of our proposed solver (11).

Theorem 1. Let the function $f(x)$ be sufficiently differentiable in a neighborhood of its simple zero α . If an initial approximation x_0 is sufficiently close to α , then, the R -order of convergence of the two-step method (11) with memory is at least 4.23607.

Proof. Let $\{x_k\}$ be a sequence of approximations generated by an iterative method. The error relations with the self-accelerating parameter $\beta = \beta_k$ for (11) are in what follows:

$$\hat{e}_k = w_k - \alpha \sim c_{k,1}e_k, \tag{13}$$

$$\tilde{e}_k = y_k - \alpha \sim c_{k,2}e_k^2, \tag{14}$$

$$e_{k+1} = x_{k+1} - \alpha \sim c_{k,4}e_k^4. \tag{15}$$

Using a symbolic computations, we attain that

$$-c_2 (f'(\alpha))^2 \beta c_2 + c_3 \sim e_{k-1}. \tag{16}$$

Substituting the value of $-c_2 (f'(\alpha))^2 \beta c_2 + c_3$ from (16) in (15), one may obtain

$$e_{k+1} \sim c_{k,4}e_{k-1}^4. \tag{17}$$

Note that in general we know that the error equation should read $e_{k+1} \sim Ae_k^p$, where A and p are to be determined. Hence, one has $e_k \sim Ae_{k-1}^p$, and subsequently

$$e_{k-1} \sim A^{-1/p}e_k^{1/p}. \tag{18}$$

Thus, it is easy to obtain

$$e_k^p \sim A^{-1/p}Ce_k^{4+1/p}, \tag{19}$$

wherein C is a constant. This results in

$$p = 4 + \frac{1}{p}, \tag{20}$$

with two solutions $\{-0.236068, 4.23607\}$. Clearly the value for $p = 4.23607$ is acceptable and would be the convergence R -order of method (11) with memory. The proof is complete. \square

The increase of R -order is attained without any (new) additional function calculations so that the novel method with memory possesses a high computational efficiency index. This technique is an extension over scheme (5) to increase the R -order from 4 to 4.23607.

The accelerating method (11) is new, simple, and useful, providing considerable improvement of convergence rate without any additional function evaluations in contrast to the optimal two-step methods without memory.

We also remark that an alternative form of our proposed method with memory could be deduced using backward finite difference formula at the beginning of the first substep and a minor modification in the accelerators; that is to

say, we have the following alternative method with memory possessing 4.23607 as its R -order (APM) as well:

$$\beta_k = \frac{N_3'''(x_k)}{3N_3'(x_k)^2 N_3''(x_k)},$$

$$y_k = x_k - \frac{f(x_k)}{\mathfrak{F}\mathfrak{D}}, \quad w_k = x_k - \beta_k f(x_k)^2, \tag{21}$$

$$x_{k+1} = y_k - \frac{f(y_k) f(x_k) - 1/2 f(y_k)}{\mathfrak{F}\mathfrak{D} f(x_k) - 5/2 f(y_k)}.$$

Theorem 2. Let the function $f(x)$ be sufficiently differentiable in a neighborhood of its simple zero α . If an initial approximation x_0 is sufficiently close to α , then, the R -order of convergence of the two-step method (21) with memory is at least 4.23607.

Proof. The proof of this theorem is similar to Theorem 1. It is hence omitted. \square

3. Numerical Computations

Computational efficiency of different iterative methods with and without memory can be measured in a prosperous manner by applying the definition of efficiency index. For an iterative method with convergence (R -)order r that requires θ functional evaluations, the efficiency index (also named computational efficiency) is calculated by Ostrowski-Traub's formula [2]:

$$E = r^{1/\theta}. \tag{22}$$

According to this, we find

$$E(\text{SM}) \approx 1.4142 < E(3) \approx 1.5737 = E(4) \approx 1.5874$$

$$= E(5) \approx 1.5874 < E(11) \approx 1.6180, \tag{23}$$

where SM is the quadratically convergent method of Steffensen without memory [7].

It should be remarked that Džunić in [8] designed an efficient one-step Steffensen-type method with memory possessing $(1/2)(3 + \sqrt{17})$ R -order of convergence as follows:

$$w_k = x_k + \beta_k f(x_k),$$

$$\beta_k = -\frac{1}{N_2'(x_k)}, \quad p_k = -\frac{N_3''(w_k)}{2N_3'(w_k)}, \tag{24}$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k] + p_k f(w_k)},$$

TABLE 1: Results of comparisons for Example 3 and to find $\alpha = 2$.

Methods	$ f(x_1) $	$ f(x_2) $	$ f(x_3) $	$ f(x_4) $	coc
KM	18.577	64890.	7.2226×10^{10}	3.2493×10^9	—
OM	4.7484	0.0023129	1.3928×10^{-16}	1.8313×10^{-69}	4.00000
DKM	0.53362	5.3207×10^{-7}	5.2711×10^{-31}	5.0774×10^{-127}	4.00000
PM	0.53362	1.9202×10^{-6}	3.6106×10^{-30}	1.6392×10^{-130}	4.22928

TABLE 2: Results of comparisons for Example 4.

Methods	$ f(x_1) $	$ f(x_2) $	$ f(x_3) $	$ f(x_4) $	coc
KM	2.0873	0.0095650	7.7971×10^{-12}	3.4597×10^{-48}	4.00000
OM	0.81344	0.0010884	1.5476×10^{-15}	6.3280×10^{-63}	4.00000
DKM	2.1909	0.013379	2.9909×10^{-11}	7.5008×10^{-46}	4.00000
PM	2.1909	0.0011772	7.0556×10^{-16}	8.4197×10^{-68}	4.23539
APM	1.9861	0.00089226	2.3251×10^{-16}	7.5243×10^{-70}	4.23526

and Cordero et al. in [9] presented a two-step biparametric Steffensen-type iterative method with memory possessing seventh R -order of convergence:

$$\begin{aligned}
 w_k &= x_k + \beta_k f(x_k), & \beta_k &= -\frac{1}{N'_3(x_k)}, \\
 p_k &= -\frac{N''_4(w_k)}{2N'_4(w_k)}, \\
 y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k] + p_k f(w_k)}, \\
 x_{k+1} &= y_k - \frac{f(y_k)}{f[x_k, y_k] + f[w_k, x_k, y_k](y_k - x_k)}.
 \end{aligned}
 \tag{25}$$

Note that our main aim was to develop King’s family in terms of efficiencies index and was not to achieve the highest possible efficiency index.

Although these methods possess higher computational efficiency indices than our proposed method (11), we exclude them from numerical comparisons since our method is not a Steffensen-type method and it is a Newton-type method with memory. For more refer to [10].

Now, we apply and compare the behavior of different methods for finding the simple zeros of some different nonlinear test functions in the programming package Mathematica [11] using multiple precision arithmetic to clearly reveal the high R -order of PM and APM. We compare methods with the same number of functional evaluations per cycle.

We notice that, by applying any root solver with local convergence, a special attention must be paid to the choice of initial approximations. If initial values are sufficiently close to the sought roots, then the expected (theoretical) convergence speed is obtainable in practice; otherwise, the iterative methods show slower convergence, especially at the beginning of the iterative process.

In this section, the computational order of convergence (coc) has been computed by

$$\text{coc} = \frac{\ln |f(x_k) / f(x_{k-1})|}{\ln |f(x_{k-1}) / f(x_{k-2})|}.
 \tag{26}$$

The calculated value coc estimates the theoretical order of convergence well when pathological behavior of the iterative method (i.e., slow convergence at the beginning of the implemented iterative method, oscillating behavior of approximations, etc.) does not exist.

Here the results of comparisons for the test functions are given by applying 1000 fixed floating point arithmetic using the stop termination $|f(x_k)| \leq 10^{-100}$.

Example 3. We consider the following nonlinear test function in the interval $D = [1.5, 2.5]$:

$$f(x) = (x - 2 \tan(x))(x^3 - 8),
 \tag{27}$$

using the initial approximation $x_0 = 1.7$. The results are provided in Table 1.

In this section, we have used $\beta_0 = 0.0001$ whenever required. Furthermore, for DKM we considered $\gamma = -1/2$.

Example 4. We compare the behavior of different methods for finding the complex solution of the following nonlinear equation:

$$g(x) = (-1 + 2I) + \frac{1}{x} + x + \sin(x),
 \tag{28}$$

using the initial approximation $x_0 = 1 - 3I$ where $\alpha = 0.28860 \dots - 1.24220 \dots I$. The results for this test are given in Table 2.

It is evident from Tables 1 and 2 that approximations to the roots possess great accuracy when the proposed method with memory is applied. Results of the fourth iterate in Tables 1 and 2 are given only for demonstration of convergence speed of the tested methods and in most cases they are not required for practical problems at present.

We also incorporated and applied the developed methods with memory (11) and (21) for different test examples and obtained results with the same behavior as above. Hence, we could mention that the theoretical results are upheld by numerical experiments and thus the new method is good with a high computational efficiency index.

4. Summary

In this paper, we have proposed a new two-step Steffensen-type iterative method with memory for solving nonlinear scalar equations. Using one self-correcting parameter calculated by Newton interpolatory polynomial, the R -order of convergence of the constructed method was increased from 4 to 4.23607 without any additional calculations.

The new method was compared in performance and computational efficiency with some existing methods by numerical examples. We have observed that the computational efficiency index of the presented method with memory is better than those of other existing two-step King-type methods.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contribution

The authors have made the same contribution. All authors read and approved the final paper.

References

- [1] A. S. Al-Fhaid, S. Shateyi, M. Z. Ullah, and F. Soleymani, "A matrix iteration for finding Drazin inverse with ninth-order convergence," *Abstract and Applied Analysis*, vol. 2014, Article ID 137486, 7 pages, 2014.
- [2] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, New York, NY, USA, 1964.
- [3] A. Cordero and J. R. Torregrosa, "Low-complexity root-finding iteration functions with no derivatives of any order of convergence," *Journal of Computational and Applied Mathematics*, 2014.
- [4] R. F. King, "A family of fourth order methods for nonlinear equations," *SIAM Journal on Numerical Analysis*, vol. 10, pp. 876–879, 1973.
- [5] N. Huang and C. Ma, "Convergence analysis and numerical study of a fixed-point iterative method for solving systems of nonlinear equations," *The Scientific World Journal*, vol. 2014, Article ID 789459, 10 pages, 2014.
- [6] J. P. Jaiswal, "Some class of third- and fourth-order iterative methods for solving nonlinear equations," *Journal of Applied Mathematics*, vol. 2014, Article ID 817656, 17 pages, 2014.
- [7] J. F. Steffensen, "Remarks on iteration," *Skandinavisk Aktuarietidskrift*, vol. 16, pp. 64–72, 1933.
- [8] J. Džunić, "On efficient two-parameter methods for solving nonlinear equations," *Numerical Algorithms*, vol. 63, no. 3, pp. 549–569, 2013.
- [9] A. Cordero, T. Lotfi, P. Bakhtiari, and J. R. Torregrosa, "An efficient two-parametric family with memory for nonlinear equations," *Numerical Algorithms*, 2014.
- [10] X. Wang and T. Zhang, "A new family of Newton-type iterative methods with and without memory for solving nonlinear equations," *Calcolo*, vol. 51, no. 1, pp. 1–15, 2014.
- [11] S. Wagon, *Mathematica in Action*, Springer, New York, NY, USA, 3rd edition, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

