

Research Article

A Family of Derivative-Free Methods with High Order of Convergence and Its Application to Nonsmooth Equations

**Alicia Cordero, José L. Hueso,
Eulalia Martínez, and Juan R. Torregrosa**

*Instituto de Matemática Multidisciplinar and Instituto de Matemática Pura y Aplicada,
Universitat Politècnica de València, 46022 Valencia, Spain*

Correspondence should be addressed to Alicia Cordero, acordero@mat.upv.es

Received 4 January 2012; Accepted 14 March 2012

Academic Editor: Muhammad Aslam Noor

Copyright © 2012 Alicia Cordero 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.

A family of derivative-free methods of seventh-order convergence for solving nonlinear equations is suggested. In the proposed methods, several linear combinations of divided differences are used in order to get a good estimation of the derivative of the given function at the different steps of the iteration. The efficiency indices of the members of this family are equal to 1.6266. Also, numerical examples are used to show the performance of the presented methods, on smooth and nonsmooth equations, and to compare with other derivative-free methods, including some optimal fourth-order ones, in the sense of Kung-Traub's conjecture.

1. Introduction

Finding iterative methods for solving nonlinear equations is an important area of research in numerical analysis, and it has interesting applications in various branches of Science and Engineering. In this study, we describe new iterative methods to find a simple root α of a nonlinear equation $f(x) = 0$, where $f : I \subset \mathbb{R} \rightarrow \mathbb{R}$ is a scalar function on an open interval I . The known Newton's method for finding α uses the iterative expression

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, \dots, \quad (1.1)$$

which converges quadratically in some neighborhood of α . If the derivative $f'(x_k)$ is replaced by the forward-difference approximation

$$f'(x_k) \approx \frac{f(x_k + f(x_k)) - f(x_k)}{f(x_k)}, \quad (1.2)$$

the Newton's method becomes

$$x_{k+1} = x_k - \frac{(f(x_k))^2}{f(x_k + f(x_k)) - f(x_k)}, \quad (1.3)$$

which is the known Steffensen's method (SM), (see [1]). Newton and Steffensen's methods are of second order, both require two functional evaluations per step, but in contrast to Newton's method, Steffensen's method is derivative-free.

The procedure of removing the derivatives usually increases the number of functional evaluations per iteration. Commonly in the literature the efficiency of an iterative method is measured by the *efficiency index* defined as $I = p^{1/d}$ (see [2]), where p is the order of convergence and d is the total number of functional evaluations per step. Kung and Traub conjectured in [3] that the order of convergence of any multipoint method cannot exceed the bound 2^{d-1} , (called *the optimal order*). Thus, the optimal order for methods with 3 or 4 functional evaluations per step would be 4 or 8, respectively.

To improve the convergence properties, many variants of Steffensen's method have been proposed in the last years. Some of these methods use forward or central divided differences for approximating the derivatives. For example, by composing Steffensen and Newton's methods and using a particular approximation of the first derivative, Liu et al. derive in [4] an optimal fourth-order method, that we denote (LZM), with three functional evaluations per step. The iterative expression is

$$x_{k+1} = y_k - \frac{f[x_k, y_k] - f[y_k, z_k] + f[x_k, z_k]}{f[x_k, y_k]^2} f(y_k), \quad k = 0, 1, \dots, \quad (1.4)$$

where y_k is the approximation of the Steffensen's method, $z_k = x_k + f(x_k)$ and $f[x, y] = (f(x) - f(y))/(x - y)$ is the divided difference of order one.

Dehghan and Hajarian [5] proposed a variant of Steffensen's method (DHM), which is written as

$$x_{k+1} = x_k - \frac{2f(x_k)[f(z_{k+1}) - f(x_k)]}{f(x_k + f(x_k)) - f(x_k - f(x_k))}, \quad (1.5)$$

where $z_{k+1} = x_k + 2(f(x_k))^2 / (f(x_k + f(x_k)) - f(x_k - f(x_k)))$. The method is obtained by replacing the forward-difference approximation in Steffensen's method by the central-difference approximation. However, it is still a method of third order and requires four functional evaluations per iteration.

The authors have also presented in [6] a one-parameter family of optimal fourth-order derivative-free methods, denoted by (CTM), which will be used in this paper as a base in order to achieve higher orders of convergence. The iterative expression of this family is

$$\begin{aligned}
 y_k &= x_k - \frac{f(x_k)^2}{f(z_k) - f(x_k)}, \\
 x_{k+1} &= y_k - \frac{f(y_k)}{(f(y_k) - \beta f(z_k))/(y_k - z_k) + (f(y_k) - \delta f(x_k))/(y_k - x_k)},
 \end{aligned}
 \tag{1.6}$$

where parameters β and δ must verify $\beta + \delta = 1$. In the numerical section, we will work with the element of this family obtained by taking $\beta = 1$ and $\delta = 0$.

In this paper, the technique used to improve the local order of convergence consists of the composition of two iterative methods of order p and q , respectively, to obtain a method of order pq (see [1]). Specifically, we compose Newton's method and the CTM family (1.6). In addition, some particular approximations of the derivative will be made in order to obtain a Steffensen-type method. As we will show, the obtained family of methods is of seventh-order of convergence and requires four evaluations of the function $f(x)$; therefore, this class of methods has efficiency index $7^{1/4} \approx 1.6266$, which is higher than $2^{1/2} \approx 1.4142$ of the Steffensen's method, $3^{1/4} \approx 1.3161$ of the DHM method (1.5), $4^{1/3} \approx 1.5874$ of the LZM (1.4) and CTM (1.6) methods. Therefore, although the methods of the new family are not optimal in the sense of Kung-Traub's conjecture, they are competitive from the point of view of the efficiency index.

Recently, some seventh-order methods have appeared in the literature: for example, Hu and Fang in [7] design a Jarratt-type scheme of order of convergence seven. Its iterative expression is

$$\begin{aligned}
 z_k &= y_k - \frac{2f'(x_k) - f'(y_k)}{f'(x_k)} \frac{f(y_k)}{f'(x_k)}, \\
 x_{k+1} &= z_k + \frac{f'(x_k) + f'(y_k)}{f'(x_k) - 3f'(y_k)} \frac{f(z_k)}{f'(x_k)},
 \end{aligned}
 \tag{1.7}$$

where y_k is the Newton's iteration. We will denote this scheme by HFM. Let us note that this method is not derivative-free, and it uses five functional evaluations. So, its efficiency index is $7^{1/5} \approx 1.4758$.

Noor et al. in [8, Algorithm 2.5] show an iterative method free from second derivative of order seven, with five functional evaluations. Its efficiency index is $7^{1/5} \approx 1.4758$ and its iterative expression is

$$\begin{aligned}
 z_k &= y_k - \frac{2f(y_k)f'(y_k)}{2(f'(y_k))^2 - f(y_k)P_f(x_k, y_k)}, \\
 x_{k+1} &= z_k - \frac{f'(x_k) + 3f'(y_k)}{6f'(y_k) - 2f'(x_k)} \frac{f(z_k)}{f'(x_k)},
 \end{aligned}
 \tag{1.8}$$

where $P_f(x_k, y_k) = (2/(y_k - x_k))(2f'(y_k) + f'(x_k) - 3f[y_k, x_k])$. We denote this scheme by NM.

Soleymani and Khattri in [9] (Theorem 1), design a derivative-free seventh-order method with four functional evaluations. Its iterative expression is

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \\ z_k &= y_k - \frac{f(y_k)}{f[x_k, w_k]} \left(1 + \frac{f(y_k)}{f(x_k)} + \frac{f(y_k)}{f(w_k)} \right), \\ x_{k+1} &= z_k - \frac{f(z_k)}{f[x_k, w_k]} \left(1 + (2 - f[x_k, w_k]) \frac{f(y_k)}{f(w_k)} + \left(\frac{1}{1 - f[x_k, w_k]} \right) \left(\frac{f(y_k)}{f(x_k)} \right)^2 + \frac{f(z_k)}{f(y_k)} \right), \end{aligned} \quad (1.9)$$

where $w_k = x_k - f(x_k)$. We denote this method by SKM. Its efficiency index is $7^{1/4} \approx 1.6266$.

The rest of the paper is organized as follows: in Section 2, we describe our family of methods and analyze its convergence order for smooth equations. In Section 3, different numerical tests confirm the theoretical results and allow us to compare this family with other known methods mentioned in this section. We also analyze in this numerical section the behavior of the new family on nonsmooth equations.

2. The Methods and Analysis of Convergence

By direct composition of the CTM family (1.6) and Newton's method, it is easy to see that the scheme

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)^2}{f(z_k) - f(x_k)}, \\ u_k &= y_k - \frac{f(y_k)}{(f(y_k) - \beta f(z_k))/(y_k - z_k) + (f(y_k) - \delta f(x_k))/(y_k - x_k)}, \\ x_{k+1} &= u_k - \frac{f(u_k)}{f'(u_k)}, \end{aligned} \quad (2.1)$$

where $z_k = x_k + f(x_k)$, is of eighth-order convergence. In order to avoid the evaluation of the first derivative in the last step, we extend the estimation used in (1.6), by replacing $f'(u_k)$ by a linear combination of several divided differences:

$$\begin{aligned} f'(u_k) \approx D(x_k) &= \frac{b_1 f(u_k) - b_2 f(z_k)}{u_k - z_k} + \frac{b_3 f(u_k) - b_4 f(x_k)}{u_k - x_k} + \frac{b_5 f(u_k) - b_6 f(y_k)}{u_k - y_k} \\ &+ \frac{b_7 f(y_k) - b_8 f(z_k)}{y_k - z_k} + \frac{b_9 f(y_k) - b_{10} f(x_k)}{y_k - x_k} + \frac{b_{11} f(z_k) - b_{12} f(x_k)}{z_k - x_k}, \end{aligned} \quad (2.2)$$

where $b_1, b_2, \dots, b_{12} \in \mathbb{R}$ are parameters. So, we are going to prove that for some values of the parameters the family of methods described by

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{(f(z_k) - f(x_k))/f(x_k)}, \\ u_k &= y_k - \frac{f(y_k)}{(f(y_k) - \beta f(z_k))/(y_k - z_k) + (f(y_k) - \delta f(x_k))/(y_k - x_k)}, \\ x_{k+1} &= u_k - \frac{f(u_k)}{D(x_k)}, \end{aligned} \quad (2.3)$$

is of seventh-order of convergence.

Theorem 2.1. *Let $\alpha \in I$ be a simple zero of a sufficiently differentiable function $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ in an open interval I . If x_0 is sufficiently close to α , then the iterative method defined by (2.3) has seventh-order of convergence for $\beta = 1 - \delta$, $b_2 = 1 - b_8 - b_4 - b_{10} - b_6$, $b_{11} = b_{12} = 0$, $b_7 = b_8 + b_4 + b_{10}$, $b_6 = 1 - b_9 - b_4$, $b_5 = 1$, and $b_8 = -1 + b_9 + b_4 - b_{10}$ and satisfies the error equation*

$$\begin{aligned} x_{k+1} - \alpha &= -(1 + f'(\alpha))^3 c_2^2 (c_2^2 - c_3) \left[(-1 + b_1 + b_3 + f'(\alpha)(b_3 - b_4)) c_2^2 \right. \\ &\quad \left. - (-2 + b_1 + b_3 + f'(\alpha)(b_3 - 1)) c_3 \right] e_k^7 + O(e_k^8), \end{aligned} \quad (2.4)$$

where $e_k = x_k - \alpha$ and $c_k = (1/k!)(f^{(k)}(\alpha)/f'(\alpha))$, $k = 2, 3, \dots$ and $\delta, \beta, b_1, b_2, \dots, b_{12} \in \mathbb{R}$.

Proof. By using Taylor's expansion around $x = \alpha$, it is easy to observe (see [6]) that if $\beta + \delta = 1$ then,

$$\begin{aligned} u_k - \alpha &= (1 + f'(\alpha))^2 c_2 (c_2^2 - c_3) e_k^4 \\ &\quad - (1 + f'(\alpha)) \left(2(2 + 2f'(\alpha) + f'(\alpha)^2) c_2^4 \right. \\ &\quad \quad \left. - 2(4 + 5f'(\alpha) + 2f'(\alpha)^2) c_2^2 c_3 \right. \\ &\quad \quad \left. + (2 + 3f'(\alpha) + f'(\alpha)^2) c_3^2 + (2 + 3f'(\alpha) + f'(\alpha)^2) c_2 c_4 \right) e_k^5 + O(e_k^6) \end{aligned} \quad (2.5)$$

and then,

$$\begin{aligned} f(u_k) &= f'(\alpha) (1 + f'(\alpha))^2 c_2 (c_2^2 - c_3) e_k^4 \\ &\quad - f'(\alpha) (1 + f'(\alpha)) \left(2(2 + 2f'(\alpha) + f'(\alpha)^2) c_2^4 \right. \\ &\quad \quad \left. - 2(4 + 5f'(\alpha) + 2f'(\alpha)^2) c_2^2 c_3 + (2 + 3f'(\alpha) + f'(\alpha)^2) c_3^2 \right. \\ &\quad \quad \left. + (2 + 3f'(\alpha) + f'(\alpha)^2) c_2 c_4 \right) e_k^5 + O(e_k^6). \end{aligned} \quad (2.6)$$

Now, the approximation of $f'(u_k)$, $D(x_k)$, can be written as

$$\begin{aligned}
 D(x_k) = & [b_{11}(1 + f'(\alpha)) - b_{12} + (b_2 + b_8 + b_4 + b_{10} + b_6) \\
 & + f'(\alpha)(b_4 - b_7 + (1 + f'(\alpha))(b_2 - b_9) + (2 + f'(\alpha))(b_{10} + b_8 + b_{11}))] c_2 e_k \\
 & + f'(\alpha) \left[(b_7 + b_9 + f'(\alpha)(b_2 + b_7 + b_8 + b_{10} + b_{11}) - (1 + f'(\alpha))(2b_6 - b_5)) c_2^2 \right. \\
 & \quad + (b_2 + b_4 + b_5 - b_6 - 2b_7 + 3b_8 + 2b_9 + f'(\alpha)(2b_2 + b_5 - b_6 - b_7 + 3b_8 - 3b_9) \\
 & \quad \left. + f'(\alpha)^2(b_2 + b_8 - b_9) + (3 + 3f'(\alpha) + f'(\alpha)^2)(b_{10} + b_{11}) \right] c_3 e_k^2 + O(e_k^3), \tag{2.7}
 \end{aligned}$$

and calculating the last step of the iterative process (2.3), we have

$$\begin{aligned}
 x_{k+1} - \alpha = & (1 + f'(\alpha))^2 c_2 (c_2^2 - c_3) \\
 & \times \left(1 - \frac{f'(\alpha)}{b_{11}(1 + f'(\alpha)) + f'(\alpha)(b_2 + b_8 + b_4 + b_{10} + b_6) - b_{12}} \right) e_k^4 + O(e_k^5). \tag{2.8}
 \end{aligned}$$

It is necessary to assign the following values to the parameters in order to assure the sixth-order of convergence: $b_2 = 1 - b_8 - b_4 - b_{10} - b_6$, $b_{11} = b_{12} = 0$, $b_7 = b_8 + b_4 + b_{10}$, and $b_6 = 1 - b_9 - b_4$. So, the error equation can be expressed as

$$\begin{aligned}
 x_{k+1} - \alpha = & (1 + f'(\alpha))^3 c_2 (c_2^2 - c_3) \left((2 - b_4 - b_5 + b_8 - b_9 + b_{10}) c_2^2 \right. \\
 & \left. + (-1 + b_5) c_3 \right) e_k^6 + O(e_k^7). \tag{2.9}
 \end{aligned}$$

Finally, if $b_5 = 1$ and $b_8 = -1 + b_9 + b_4 - b_{10}$, we have

$$\begin{aligned}
 x_{k+1} - \alpha = & -(1 + f'(\alpha))^3 c_2^2 (c_2^2 - c_3) \left[(-1 + b_1 + b_3 + f'(\alpha)(b_3 - b_4)) c_2^2 \right. \\
 & \left. - (-2 + b_1 + b_3 + f'(\alpha)(b_3 - 1)) c_3 \right] e_k^7 + O(e_k^8). \tag{2.10}
 \end{aligned}$$

□

In terms of computational cost, the methods of this family require only four functional evaluations per step. So, they have efficiency indices $7^{1/4} = 1.6266$. If we denote by M7 any element of this family, we can establish

$$I_{M7} = I_{SKM} > I_{CTM} = I_{LZM} > I_{NM} = I_{HFM} > I_{SM} > I_{DHM}. \tag{2.11}$$

In the next section, we use the element of family (2.3) obtained by choosing (for simplicity) $\delta = b_1 = b_3 = b_4 = b_9 = b_{10} = b_{11} = b_{12} = 0$ and, therefore, $\beta = 1$, $b_2 = b_6 = 1$, and $b_7 = b_8 = -1$. So, the resulting iterative expression of the method is

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)^2}{f(z_k) - f(x_k)}, \\ u_k &= y_k - \frac{f(y_k)}{(f(y_k) - f(z_k))/(y_k - z_k) + f(y_k)/(y_k - x_k)}, \\ x_{k+1} &= u_k - \frac{f(u_k)}{D(x_k)}, \end{aligned} \quad (2.12)$$

where $D(x_k) = (f(u_k) - f(y_k))/(u_k - y_k) - f(z_k)/(u_k - z_k) - (f(y_k) - f(z_k))/(y_k - z_k)$ and $z_k = x_k + f(x_k)$.

It is well known that if α is a multiple zero of $f(x)$, then α is a simple zero of $f(x)/f'(x)$. In a similar way, for Steffensen-type methods, it is easy to prove that if we use $f_0(x) = (f(x))^2/(f(x + f(x)) - f(x))$, we transform the problem of solving multiple roots of $f(x) = 0$ into a simple roots one, $f_0(x) = 0$. Theoretically, this idea improves the order of convergence but in practice, results are not satisfactory.

As we have seen, the method (2.12) has seventh-order convergence for smooth equations but, what is its behavior for nonsmooth equations? As we will see in the next section, for this class of equations our method, in general, loses the seventh-order convergence and stability problems appear. For nonsmooth functions, Amat and Busquier in [10] presented an strategy to control the approximation of the derivative and the stability of the iteration. They applied this idea to Steffensen's method, obtaining a new scheme (STM):

$$x_{k+1} = x_k - \frac{f(x_k)}{[f(x_k + \alpha_k |f(x_k)|f(x_k)) - f(x_k)]/\alpha_k |f(x_k)|f(x_k)}, \quad (2.13)$$

where the parameters $\alpha_k \in \mathbb{R}$ allow to control the approximation of the derivative. This procedure can be applied to any other derivative-free scheme. The authors showed the second-order convergence of (2.13) for nonsmooth functions and mentioned that, in order to control the stability in practice, the parameters α_k should verify

$$\text{tol}_c \ll |\alpha_k |f(x_k)|f(x_k)| \leq \text{tol}_u, \quad (2.14)$$

where tol_c is related to the computer precision and tol_u is a free parameter.

In the following section, we will apply this strategy to our proposed method, M7, obtaining a modified scheme that will be denoted as M7mod. Then, we will analyze how this new scheme improves in nonsmooth cases, although the order of convergence at singular points decreases to four.

3. Numerical Results

This numerical section is divided into two parts: one devoted to compare the different methods on smooth equations and other in which we analyze the behavior of our method on nonsmooth test functions.

In the first part of this section, we use some test functions in order to check the effectiveness of the new high-order method (2.12), we compare it with the classical Steffensen's method, SM, the method DHM, and the optimal fourth-order methods, LZM and CTM with $\beta = 1$ and $\delta = 0$. These methods are employed to find the zeros of some nonlinear functions, specifically,

- (i) $f_1(x) = \sin^2 x - x^2 + 1, \alpha \approx 1.404492,$
- (ii) $f_2(x) = x^2 - e^x - 3x + 2, \alpha \approx 0.257530,$
- (iii) $f_3(x) = \cos x - x, \alpha \approx 0.739085,$
- (iv) $f_4(x) = (x - 1)^3 - 1, \alpha = 2,$
- (v) $f_5(x) = x^3 - 10, \alpha \approx 2.154435,$
- (vi) $f_6(x) = \cos(x) - xe^x + x^2, \alpha \approx 0.639154,$
- (vii) $f_7(x) = e^x - 1.5 - \arctan(x), \alpha \approx 0.767653,$
- (viii) $f_8(x) = x^3 + 4x^2 - 10, \alpha \approx 1.365230,$
- (ix) $f_9(x) = 8x - \cos(x) - 2x^2, \alpha \approx 0.128077,$
- (x) $f_{10}(x) = \arctan(x), \alpha = 0.$

The complexity of the iterative expressions plays an important role in the computational efficiency of the different methods. So, some authors use another index in order to compare the iterative methods, taking also into account the number of products and quotients involved in each step of the iterative process. The *computational efficiency index* is defined as $CI = p^{1/(d+op)}$, (see [11]), where p is the order of convergence, d is the number of functional evaluations, and op is the number of products and quotients per iteration. Under the point of view of this index, the relationship between the schemes that we use in this section is

$$\begin{aligned}
 CI_{\text{CTM}}(4^{1/8}) &= CI_{\text{SM}}(2^{1/4}) > CI_{\text{M7}}(7^{1/12}) \\
 &= CI_{\text{HFM}} > CI_{\text{DHM}}(3^{1/7}) \\
 &> CI_{\text{LZM}}(4^{1/10}) > CI_{\text{NM}}(7^{1/16}) > CI_{\text{SKM}}(7^{1/18}).
 \end{aligned} \tag{3.1}$$

Nowadays, high-order methods are important because numerical applications use high precision in their computations; for this reason, numerical computations have been carried out using variable precision arithmetic in Matlab 7.12 (R2011a) with 500 significant digits. The computer specifications are Intel(R) Core(TM) i5-2500 CPU @ 3.30 GHz with 16.00 GB of RAM. The stopping criterion used is $|x_{k+1} - x_k| < 10^{-150}$ or $|f(x_k)| < 10^{-150}$. The information shown in Tables 1 and 2 is, for every method, the number of iterations needed to reach the required tolerance (if the method does not converge, it will be denoted

Table 1: Numerical results from f_1 to f_{10} .

		SM	DHM	LZM	CTM	M7
$f_1, x_0 = 0.9$	$ x_{k+1} - x_k $	$3.9289e - 112$	$1.7589e - 055$	$3.7228e - 122$	$1.5049e - 124$	$1.9456e - 023$
	$ f(x_{k+1}) $	$4.4514e - 223$	$2.8819e - 164$	0	0	$1.8101e - 159$
	ACOC	2.0000	3.0000	4.0000	4.0000	6.6629
	iter	9	7	5	5	3
	e-time	0.1217	0.1891	0.1139	0.1033	0.0927
$f_2, x_0 = 1.2$	$ x_{k+1} - x_k $	$1.4587e - 149$	$7.6358e - 103$	$3.4035e - 138$	$2.6499e - 141$	$3.1050e - 029$
	$ f(x_{k+1}) $	$2.0878e - 298$	$2.5787e - 306$	0	0	$1.0495e - 202$
	ACOC	2.0000	3.0000	4.0000	4.0000	6.8723
	iter	9	11	5	5	3
	e-time	0.1282	0.2791	0.1161	0.1092	0.1075
$f_3, x_0 = 2.1$	$ x_{k+1} - x_k $	$8.3630e - 085$	$1.9786e - 059$	$1.0746e - 143$	$1.4483e - 112$	$5.6495e - 024$
	$ f(x_{k+1}) $	$1.7410e - 169$	$1.8682e - 177$	0	0	$3.7489e - 167$
	ACOC	2.0000	3.0000	4.0000	4.0000	7.0731
	iter	8	6	5	5	3
	e-time	0.0746	0.0894	0.0874	0.0867	0.0468
$f_4, x_0 = 2.2$	$ x_{k+1} - x_k $	$1.9109e - 116$	$1.8976e - 072$	$3.3922e - 110$	$1.0118e - 116$	$3.4709e - 027$
	$ f(x_{k+1}) $	$4.3820e - 231$	$1.5033e - 214$	0	0	$5.1781e - 184$
	ACOC	2.0000	3.0000	4.0000	4.0000	6.8325
	iter	10	6	5	5	3
	e-time	0.1176	0.1443	0.1246	0.0914	0.0930
$f_5, x_0 = 2.3$	$ x_{k+1} - x_k $	$7.8747e - 085$	$1.2204e - 123$	$9.1432e - 142$	$8.5347e - 144$	$1.2638e - 030$
	$ f(x_{k+1}) $	$5.9818e - 167$	0	0	0	$6.8463e - 207$
	ACOC	2.0000	3.0000	4.0000	4.0000	6.8181
	iter	10	7	5	5	3
	e-time	0.1168	0.1353	0.0945	0.1059	0.0596
$f_6, x_0 = 2$	$ x_{k+1} - x_k $	$1.4558e - 087$	—	$2.1767e - 109$	$5.9067e - 112$	$5.4741e - 023$
	$ f(x_{k+1}) $	$5.7398e - 174$	—	0	0	$9.2491e - 157$
	ACOC	2.0000	—	4.0000	4.0000	5.9331
	iter	8	nc	5	5	3
	e-time	0.1245	—	0.1576	0.1418	0.1226
$f_7, x_0 = 0.5$	$ x_{k+1} - x_k $	$5.1639e - 127$	$3.2959e - 079$	$1.5312e - 050$	$3.3808e - 073$	$4.7872e - 034$
	$ f(x_{k+1}) $	$9.3020e - 253$	$1.1763e - 235$	$4.6052e - 199$	$7.2079e - 290$	$9.9787e - 234$
	ACOC	2.0000	3.0000	3.9999	4.0000	6.8055
	iter	11	6	5	5	3
	e-time	0.1671	0.1696	0.1426	0.1349	0.0732
$f_8, x_0 = 1.5$	$ x_{k+1} - x_k $	$1.0817e - 142$	$8.4333e - 122$	$6.9628e - 136$	$2.1376e - 137$	$1.1249e - 030$
	$ f(x_{k+1}) $	$1.6591e - 282$	0	0	0	$7.6946e - 207$
	ACOC	2.0000	3.0000	4.0000	4.0000	6.7788
	iter	11	7	5	5	3
	e-time	0.1886	0.2356	0.1797	0.1463	0.1170

Table 1: Continued.

$f_9, x_0 = 0.8$	$ x_{k+1} - x_k $	$2.2055e-129$	$4.6810e-128$	$2.9693e-139$	$7.1679e-140$	$6.1073e-028$
	$ f(x_{k+1}) $	$9.0498e-257$	0	0	0	$1.6582e-191$
	ACOC	2.0000	3.0000	4.0000	4.0000	6.7613
	iter	15	6	7	8	4
	<i>e</i> -time	0.2368	0.1680	0.2684	0.1902	0.1232
$f_{10}, x_0 = 0.6$	$ x_{k+1} - x_k $	$2.4132e-081$	$1.3419e-130$	$6.2415e-141$	$1.0766e-031$	$2.7207e-019$
	$ f(x_{k+1}) $	$2.8106e-242$	0	0	$1.9282e-155$	$2.1785e-167$
	ACOC	3.0000	5.0000	5.0000	4.9922	8.7406
	iter	7	5	5	4	3
	<i>e</i> -time	0.0955	0.1471	0.0746	0.0733	0.0749

by "nc"), the last value of $|x_{k+1} - x_k|$ and $|f(x_{k+1})|$, and the approximated computational order of convergence (ACOC) ρ , defined by the authors in [12]:

$$\rho = \frac{\ln(|x_{k+1} - x_k|/|x_k - x_{k-1}|)}{\ln(|x_k - x_{k-1}|/|x_{k-1} - x_{k-2}|)}. \quad (3.2)$$

By means of (3.2), a vector is obtained by using the different iterations calculated in the process. The value of ρ that appears in Tables 1 to 4 is the last coordinate of this vector when the variation between its components is small. Let us note that when the approximated convergence order is not stable (if the difference between two consecutive values is bigger than one unit), we will denote it by '—'.

On the other hand, in Tables 1 and 2, the mean elapsed time, calculated by means of the command "cputime" of Matlab (*e*-time), after 100 performances of the program, appears. It can be observed that, in most cases, the elapsed time taken by M7 to obtain the solution is lower than the corresponding ones of the other methods involved. In terms of computational effort, the efficiency of the proposed method is not lower than that of the optimal fourth-order methods. These elapsed times are in concordance with the computational efficiency index of each method.

Numerical results in Table 1 confirm the theoretical statements developed in this paper, showing that the estimated order of convergence coincides with the theoretical one, except in case f_{10} : the second derivative of this nonlinear function at the solution is zero, so the order of convergence increases: from second to third in SM, from third (and fourth) to fifth in DHM (and LZM or CTM), and from seventh to ninth in M7. Nevertheless, in this case the *e*-time from the new method M7 does not improve the other ones. In fact, in case f_{10} the best time is obtained by Steffensen's method, followed by CTM and DHM. In general, there can be stated that the new high-order scheme improves the results obtained by other known methods, even optimal fourth-order methods such as LZM and CTM.

In Table 2, we compare the new method M7 with other known seventh-order schemes described in the introduction, HFM, SKM, and NM. It can be observed that M7 performs better than SKM and NM, but HFM is more precise than the rest of methods.

Now, we are going to make some numerical tests in order to check how the methods SM and M7 behave in nonsmooth cases. Moreover, we apply the α_k -procedure to both methods to avoid some stability problems. In these cases, numerical computations have been carried out using simple precision arithmetic, so $\text{tol}_c = 10^{-16}$, and the stopping criterion used

Table 2: Numerical results from f_1 to f_{10} and seventh-order schemes.

		HFM	SKM	NM	M7
$f_1, x_0 = 0.9$	$ x_{k+1} - x_k $	$9.2403e - 050$		$7.8995e - 104$	$1.9456e - 023$
	$ f(x_{k+1}) $	0		0	$1.8101e - 159$
	ACOC	—		7.0023	6.6629
	iter	6	≥ 100	4	3
	e-time	0.1126		0.1404	0.0927
$f_2, x_0 = 1.2$	$ x_{k+1} - x_k $	$1.1602e - 045$	$1.0203e - 036$	$6.3747e - 058$	$3.1050e - 029$
	$ f(x_{k+1}) $	0	0	0	$1.0495e - 202$
	ACOC	6.7965	7.2403	6.7108	6.8723
	iter	3	4	3	3
	e-time	0.0604	0.1445	0.1100	0.1075
$f_3, x_0 = 2.1$	$ x_{k+1} - x_k $	$1.8810e - 040$	$9.8612e - 037$	$1.4468e - 052$	$5.6495e - 024$
	$ f(x_{k+1}) $	0	0	0	$3.7489e - 167$
	ACOC	6.3228	6.2401	6.5225	7.0731
	iter	3	4	3	3
	e-time	0.0315	0.0969	0.0674	0.0468
$f_4, x_0 = 2.2$	$ x_{k+1} - x_k $	$5.1064e - 034$	—	$9.9985e - 042$	$3.4709e - 027$
	$ f(x_{k+1}) $	0	—	0	$5.1781e - 184$
	ACOC	6.8073	—	6.9177	6.8325
	iter	3	n.c.	3	3
	e-time	0.0532	—	0.0842	0.0930
$f_5, x_0 = 2.3$	$ x_{k+1} - x_k $	$2.6845e - 053$	$3.2107e - 025$	$1.2399e - 062$	$1.2638e - 030$
	$ f(x_{k+1}) $	$3.1147e - 207$	$9.9146e - 167$	$3.1147e - 207$	$6.8463e - 207$
	ACOC	6.9527	7.1651	6.9798	6.8181
	iter	3	5	3	3
	e-time	0.0406	0.0894	0.0961	0.0596
$f_6, x_0 = 2$	$ x_{k+1} - x_k $	$8.6332e - 124$		$1.0547e - 075$	$5.4741e - 023$
	$ f(x_{k+1}) $	0		0	$9.2491e - 157$
	ACOC	6.9990		6.9948	5.9331
	iter	4	≥ 100	4	3
	e-time	0.0911		0.1409	0.1226
$f_7, x_0 = 0.5$	$ x_{k+1} - x_k $	$8.9879e - 076$	$6.4335e - 045$	$1.6607e - 028$	$4.7872e - 034$
	$ f(x_{k+1}) $	$3.8934e - 208$	$1.9467e - 208$	$1.1707e - 195$	$3.8434e - 208$
	ACOC	7.0115	6.8734	7.2394	6.8055
	iter	4	3	3	3
	e-time	0.0747	0.0757	0.1214	0.0732
$f_8, x_0 = 1.5$	$ x_{k+1} - x_k $	$2.1424e - 053$		$8.0693e - 063$	$1.1249e - 030$
	$ f(x_{k+1}) $	0		0	$9.3442e - 207$
	ACOC	6.9528		6.9783	6.7788
	iter	3	≥ 100	3	3
	e-time	0.0730		0.0967	0.1170

Table 2: Continued.

$f_9, x_0 = 0.8$	$ x_{k+1} - x_k $	$1.6408e - 027$	$1.3528e - 128$	$4.3711e - 039$	$6.1073e - 028$
	$ f(x_{k+1}) $	$1.5509e - 190$	$2.4334e - 208$	$1.2167e - 208$	$1.6582e - 191$
	ACOC	7.3243	6.9999	7.1316	6.7613
	iter	3	4	3	4
	e -time	0.0552	0.1158	0.1112	0.1232
$f_{10}, x_0 = 0.6$	$ x_{k+1} - x_k $	$2.0089e - 035$	—	$1.1046e - 051$	$2.7207e - 019$
	$ f(x_{k+1}) $	0	—	0	$2.1785e - 167$
	ACOC	10.9352	—	10.9217	8.7406
	iter	3	n.c.	3	3
	e -time	0.0367	—	0.0922	0.0749

has been $|x_{k+1} - x_k| < tol_u = 10^{-11}$ or $|f(x_{k+1})| < tol_u = 10^{-11}$. From a sufficiently small α_0 , we use the following algorithm to compute the different α_k :

$$\alpha_{k+1} = \begin{cases} \alpha_k^2 & \text{if } |\alpha_k^2|f(x_k)|f(x_k)| \geq tol_c, \\ \frac{tol_c}{\|f(x_k)|f(x_k)|} & \text{elsewhere.} \end{cases} \quad (3.3)$$

The first test has been made on the function:

$$f_{11}(x) = \begin{cases} x(x+1) & \text{if } x < 0, \\ -2x(x-1) & \text{if } x \geq 0, \end{cases} \quad (3.4)$$

that can be found in [13]. We use three initial estimations in order to approximate the three different roots of the equation, $\{0, 1, -1\}$. In Table 3, we show for each initial estimations and every method, the exact absolute error at first and last iterations, the absolute difference between the two last iterations $|x_{k+1} - x_k|$, the value of f in the last iteration $|f(x_{k+1})|$, and the ACOC. From Table 3 it can be inferred that the order of convergence of M7 method decreases to five and stability problems appear when it is applied to nonsmooth equations. Nevertheless, it usually performs better than or equal to Steffensen's method and its modifications by the α_k procedure. Indeed, when this strategy is applied on the seventh-order method (M7mod), the stability of the method is improved and it results in more precise estimations with less iterations. In this example, the ACOC is not stable in some cases.

Finally, we consider the nonsmooth function

$$f_{12}(x) = |x^2 - 9|. \quad (3.5)$$

The numerical experiments made on this function are summarized in Table 4. In this case, the advantages of the modified methods over original Steffensen's method and M7 method are more evident when the initial estimation is far from the zero of the function. When the initial estimation is good enough, it is clear that the behavior of M7mod improves lower-order methods, in terms of precision and number of iterations.

Table 3: Numerical results for function $f_{11}(x)$.

$x_0 = 0.1$	SM		STM ($tol_u = 10^{-11}$)		M7		M7mod	
	iter	error	iter	error	iter	error	iter	error
$\alpha = 0$	1	$4.52e-2$	1	$1.50e-2$	1	$1.50e-3$	1	$2.28e-4$
	2	$4.60e-3$	2	$2.32e-4$	2	$3.49e-11$	2	0
	⋮		3	$5.38e-8$	3	0		
	4	$8.32e-9$	4	0				
	5	0						
$ x_{k+1} - x_k $	$8.32e-9$		$5.38e-8$		$3.49e-11$		$2.28e-4$	
$ f(x_{k+1}) $	$2.08e-16$		$5.80e-15$		$3.76e-37$		$4.08e-15$	
ACOC	2.0919		2.0000		4.2042		—	
$x_0 = 3$	iter	error	iter	error	iter	error	iter	error
$\alpha = 1$	1	0.29	1	0.80	1	$3.45e-2$	1	$6.22e-2$
	2	0.15	2	0.25	2	$6.91e-11$	2	$2.35e-9$
	⋮		⋮		3	NaN	3	0
	5	$7.98e-8$	6	$5.36e-12$				
	6	$1.0e-14$	7	0				
$ x_{k+1} - x_k $	$7.98e-8$		$5.36e-12$		NaN		$2.35e-9$	
$ f(x_{k+1}) $	$1.27e-14$		0		NaN		0	
ACOC	1.9997		2.0000		—		4.9686	
$x_0 = -20$	iter	error	iter	error	iter	error	iter	error
$\alpha = -1$	1	18.44	1	9.26	1	7.17	1	2.31
	2	17.88	2	4.39	2	2.38	2	$8.82e-2$
	⋮		⋮		⋮		3	$2.29e-10$
	30	$5.65e-5$	8	$1.98e-6$	4	$2.72e-8$	4	0
	31	$1.80e-13$	9	$3.93e-12$	5	NaN		
$ x_{k+1} - x_k $	$5.65e-5$		$1.98e-6$		NaN		$2.29e-8$	
$ f(x_{k+1}) $	$1.80e-13$		$3.93e-12$		NaN		0	
ACOC	—		1.9991		—		—	

4. Conclusions

A new seventh-order family of derivative-free iterative methods for solving nonlinear equations has been presented. As only four functional evaluations are required per iteration, the efficiency index of each member of this family is equal to $7^{1/4} = 1.6266$. In addition, these methods use a small amount of products and quotients and are derivative-free, which allow us to apply them also to nonsmooth equations with positive and promising results.

The generalization of these methods to nonlinear systems $F(x) = 0$ is similar to the classical Steffensen’s method (see [1]):

$$\begin{aligned}
 \tilde{x}_k &= x_k + \alpha_k \|F(x_k)\| F(x_k), \\
 x_{k+1} &= x_k - [x_k, \tilde{x}_k; F]^{-1} F(x_k),
 \end{aligned}
 \tag{4.1}$$

Table 4: Numerical results for function $f_{12}(x)$.

$x_0 = 2.8$	SM		STM ($\text{tol}_u = 10^{-11}$)		M7		M7mod	
	iter	error	iter	error	iter	error	iter	error
$\alpha = 3$	1	0.44	1	$7.14e-3$	1	0.17	1	$7.54e-5$
	2	0.89	2	$8.39e-6$	2	$1.69e-11$	2	$1.0e-14$
	\vdots		3	$1.17e-11$	3	0		
	24	$6.0e-10$	4	0				
	25	0						
$ x_{k+1} - x_k $	$6.0e-10$		$1.17e-11$		$1.69e-8$		$7.54e-5$	
$ f(x_{k+1}) $	0		0		0		$6.93e-14$	
ACOC	2.0009		1.9984		—		—	
$x_0 = -2.8$	iter		error		iter		error	
	$> 10^4$		1	$7.14e-3$	1	$5.66e-3$	1	$7.54e-5$
$\alpha = -3$			2	$8.39e-6$	2	$4.41e-3$	2	$1.0e-14$
			3	$1.17e-11$	\vdots			
			4	0	4	$2.74e-10$		
					5	NaN		
$ x_{k+1} - x_k $			$1.17e-11$		NaN		$7.54e-5$	
$ f(x_{k+1}) $			0		NaN		$6.93e-14$	
ACOC			1.9984		—		—	
$x_0 = -10$	iter		error		iter		error	
	$> 10^4$		1	2.45	1	0.70	1	$7.98e-2$
$\alpha = -3$			2	0.55	2	0.73	2	$3.2e-13$
			\vdots		\vdots			
			5	$2.05e-8$	7	$2.23e-9$		
			6	0	8	NaN		
$ x_{k+1} - x_k $			$2.05e-8$		NaN		$7.98e-2$	
$ f(x_{k+1}) $			0		NaN		$1.95e-12$	
ACOC			1.9368		—		—	

where $[u, v; F] : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a linear operator such that $[u, v; F](u - v) = F(u) - F(v)$, which is called divided difference.

Acknowledgments

The authors would like to thank the referees for their valuable comments and for their suggestions to improve the readability of the paper. This research was supported by Ministerio de Ciencia y Tecnología MTM2011-28636-C02-02 and by Vicerrectorado de Investigación, Universitat Politècnica de València PAID-06-2010-2285.

References

- [1] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, NY, USA, 1970.
- [2] A. M. Ostrowski, *Solution of Equations and Systems of Equations*, Pure and Applied Mathematics, Vol. 9, Academic Press, New York, NY, USA, 1966.
- [3] H. T. Kung and J. F. Traub, "Optimal order of one-point and multipoint iteration," *Journal of the Association for Computing Machinery*, vol. 21, pp. 643–651, 1974.
- [4] Z. Liu, Q. Zheng, and P. Zhao, "A variant of Steffensen's method of fourth-order convergence and its applications," *Applied Mathematics and Computation*, vol. 216, no. 7, pp. 1978–1983, 2010.
- [5] M. Dehghan and M. Hajarian, "Some derivative free quadratic and cubic convergence iterative formulas for solving nonlinear equations," *Computational & Applied Mathematics*, vol. 29, no. 1, pp. 19–30, 2010.
- [6] A. Cordero and J. R. Torregrosa, "A class of Steffensen type methods with optimal order of convergence," *Applied Mathematics and Computation*, vol. 217, no. 19, pp. 7653–7659, 2011.
- [7] Y. Hu and L. Fang, "A seventh-order convergence Newton-type method for solving nonlinear equations," in *2nd International Conference on Computational Intelligence and Natural Computing*, 2010.
- [8] M. A. Noor, W. A. Khan, K. I. Noor, and E. Al-said, "High-order iterative method free from second derivative for solving nonlinear equations," *International Journal of the Physical Science*, vol. 6, no. 8, pp. 1887–1893, 2011.
- [9] F. Soleymani and S. K. Khattri, "Finding simple roots by seventh and eighth order derivative-free methods," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 6, pp. 45–52, 2012.
- [10] S. Amat and S. Busquier, "On a Steffensen's type method and its behavior for semismooth equations," *Applied Mathematics and Computation*, vol. 177, no. 2, pp. 819–823, 2006.
- [11] A. Cordero, J. L. Hueso, E. Martínez, and J. R. Torregrosa, "A modified Newton-Jarratt's composition," *Numerical Algorithms*, vol. 55, no. 1, pp. 87–99, 2010.
- [12] A. Cordero and J. R. Torregrosa, "Variants of Newton's method using fifth-order quadrature formulas," *Applied Mathematics and Computation*, vol. 190, no. 1, pp. 686–698, 2007.
- [13] S. Amat and S. Busquier, "On a higher order secant method," *Applied Mathematics and Computation*, vol. 141, no. 2-3, pp. 321–329, 2003.