Generalized multistep Steffensen iterative method. Solving the model of a photomultiplier device

Eva G. Villalba[a], José. L. Hueso[a] and Eulalia Martínez[a]

[a]Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València. València, Spain.

ARTICLE HISTORY
Compiled February 22, 2023

Abstract

It is well known that the Steffensen-type methods approximate the derivative appearing in Newton's scheme by means of the first-order divided difference operator. The generalized multistep Steffensen iterative method consists of composing the method with itself m times. Specifically, the divided difference is held constant for every m steps before it is updated. In this work, we introduce a modification to this method, in order to accelerate the convergence order. In the proposed, scheme we compute the divided differences in first and second step and use the divided difference from the second step in the following $m-1$ steps. We perform an exhaustive study of the computational efficiency of this scheme and also introduce memory to this method to speed up convergence without performing new functional evaluations. Finally, some numerical examples are studied to verify the usefulness of these algorithms.

KEYWORDS
Iterative processes, convergence order, computational efficiency index, divided differences, derivative-free iterative processes. Dynamics.

AMS CLASSIFICATION
45G10, 47H17, 65J15.

1. Introduction

Within the field of numerical analysis, the solution of nonlinear equations and systems of equations is one of the most relevant and most studied aspects. This is because other large blocks of this field, like ordinary differential equations, partial derivative equations or nonlinear integral equations, can be reduced in some part of the process to finding the solution of a nonlinear system of the form $F(x) = 0$. Generally, it is very difficult to find the exact solution of this type of problem, so iterative methods to approximate their solution have been an important field of research in recent years.

Newton's method is the best known iterative methods to approximate the solution of $F(x) = 0$, [18]. This method is a simple, efficient and optimal method. However, it can be applied only to differentiable operators due to the presence of derivatives in its iterative scheme.

When derivatives are difficult to obtain, either due to the computational cost of $F'(x)$ or due to the operator being non-differentiable, derivative-free methods are considered, among which Steffensen's method is the first option ([1, 5, 8, 13]). In the multidimensional case, this method

CONTACT A. N. Author. Email: eumarti@mat.upv.es

approximates the derivative present in Newton's scheme by means of the first-order divided difference operator $[x; x + F(x); F]$.

Definition 1.1. We denote by $\mathcal{L}\ (X, Y)$ the space of bounded linear operator of X to Y. An operator $[x, y; F] \in \mathcal{L}\ (X, Y)$ is called first order divided difference for the function $F : \Omega \subset X \to Y$ at points x and y $(x \neq y)$ if

$$[x, y; F](x - y) = F(x) - F(y).$$

Amat et al. in [4] improved the applicability of Steffensen's method, ensuring its second order of convergence, presenting the following generalization of Steffensen's method:

$$\begin{cases} x_0 \in \Omega \\ u_k = x_k - aF(x_k), \qquad a \in \mathbb{R}^+, \\ v_k = x_k + bF(x_k), \qquad b \in \mathbb{R}^+, k \in \mathbb{N}, \\ x_{k+1} = x_k - [v_k, u_k; F]^{-1}F(x_k), \qquad k \geq 0. \end{cases} \tag{1}$$

where a and b are arbitrary constants such that $a \neq 0$ or $b \neq 0$ and $a \neq b$. In [4] it is shown that this generalization preserves the efficiency and the quadratic order of convergence presented by Steffensen's method. However, increasing the convergence speed of this scheme involves performing more functional evaluations of the nonlinear operator and more calculations.

As a consequence, many articles focus on the construction of derivative-free multistep iterative schemes, which manage to increase the order of convergence of methods such as Steffensen or (1), without greatly increasing the number of operations performed in each step.

[3] and [15] a multistep iterative process based on the composition of method(1) with itself m times are studied, so that in each of the m steps the divided difference already evaluated in the first one is used. We center in the iterative method defined in [15] that has the following algorithm:

$$S1_m : \quad \begin{cases} x_0 \in \Omega, \\ x_k^{(1)} = x_k^{(0)} - [v_k, u_k; F]^{-1}F(x_k^{(0)}), \\ \quad \vdots \\ x_k^{(m-1)} = x_k^{(m-2)} - [v_k, u_k; F]^{-1}F(x_k^{(m-2)}), \\ x_{k+1}^{(m)} = x_k^{(m-1)} - [v_k, u_k; F]^{-1}F(x_k^{(m-1)}), \quad k \geqslant 0, \end{cases} \tag{2}$$

where $u_k = x_k - aF(x_k)$, $v_k = x_k + bF(x_k)$, $x_k = x_k^{(0)}$, a and b are arbitrary constants such that $a \neq 0$ or $b \neq 0$, $a \neq b$ and $x_{k+1} = x_k^{(m)}$, with $m \geq 1$. It is well known that this iterative method reaches convergence order $m + 1$.

In order to accelerate the convergence order of this method and, in turn, improve its efficiency, in this article we build a new derivative-free multistep iterative scheme and study its speed of convergence; this is done in Section 2. To compare these methods with other multistep schemes, we will not only pay attention to the order of convergence but also study its efficiency by measuring it under different criteria; thus, in Section 3 we deal with the computational efficiency of different methods. Next, in Section 4 we use some processes seen in [15] to introduce memory to this method, so that its convergence speed is improved without performing new functional evaluations. In Section 5 some numerical examples are studied to verify the usefulness of these algorithms. Finally, Section 6 is devoted to draw some conclusions and final remarks.

2. Generalized multistep Steffensen's methods by freezing the second divided difference

In this Section we present a new multistep method based on the Steffensen's method, which manages to achieve a higher order of convergence than the generalized Steffensen method given in (1). The idea of this process starts from composing the Steffensen's method m times but, in this case, using two divided differences in each iteration. The algorithm consists in performing two steps of the generalized Steffensen's method updating the divided difference operator in both of them and from the third step on, we will "freeze" the divided difference operator used in the second step to calculate the rest of the steps:

$$S2_m : \quad \begin{cases} x_0 \in \Omega, \\ x_k^{(1)} = x_k^{(0)} - [r_k, s_k; F]^{-1} F(x_k^{(0)}), \\ x_k^{(2)} = x_k^{(1)} - [v_k, u_k; F]^{-1} F(x_k^{(1)}), \\ \quad \vdots \\ x_k^{(m-1)} = x_k^{(m-2)} - [v_k, u_k; F]^{-1} F(x_k^{(m-2)}), \\ x_{k+1}^{(m)} = x_k^{(m-1)} - [v_k, u_k; F]^{-1} F(x_k^{(m-1)}), \quad k \geq 0, \end{cases} \tag{3}$$

where $s_k = x_k^{(0)} - aF(x_k^{(0)})$, $r_k = x_k^{(0)} + bF(x_k^{(0)})$, $u_k = x_k^{(1)} - cF(x_k^{(1)})$, $v_k = x_k^{(1)} + dF(x_k^{(1)})$; a, b, c and d are nonzero arbitrary constants, $a \neq b$, $c \neq d$ and $x_{k+1} = x_k^{(m)}$, with $m \geq 2$.

We denote this method by $S2_m$. It is obvious that $S2_m$ will have a higher computational cost than the generalized Steffensen 's method $S1_m$ for the same number of steps as the divided difference operator is evaluated twice. However, we can reach the same order of convergence than this method with less steps. So, we are interested in studying which method is more convenient in terms of efficiency as a function of the number of steps m, the size n of the system to be solved, of the order p and the number of iterations k that each one needs to perform to approximate the solution of a problem.

First of all, in the following result we will prove that the method (3) has convergence order $2m$, where m is the number of steps of the method ($m \geq 2$). For this purpose, we consider the characterization of divided difference operator introduced in [17], given by

$$[x, x+h; F] = \int_0^1 F'(x+th)\,\mathrm{d}t, \qquad (x,h) \in \mathbb{R}^m \times \mathbb{R}^m. \tag{4}$$

Now, integrating the Taylor's expansion of $F'(x+th)$ around x we have:

$$[x, x+h; F] = F'(x) + \frac{1}{2}F''(x)h + \frac{1}{6}F'''(x)h^2 + O(h^3). \tag{5}$$

Then, we establish the following result:

Theorem 2.1. *Let $F : \Omega \subset \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be a smooth function in the neighborhood of x^*, that is a nonempty open convex set $\Omega \subset \mathbb{R}^n$ such that $F(x^*) = 0$. Then the m-step iterative method given by (3) with a, b, c and d non-negative arbitrary constants and x_0 a close approximation to x^*, converges to x^* with convergence order $2m$.*

Proof. We will do the demonstration by induction on the number of steps $m \geq 1$.

- For $m = 1$, we are studying the order of convergence of a one-step method, which is the generalization of Steffensen's method.

  Let $e_k^{(0)} = x_k^{(0)} - x^*$ the error in the k-th approximation to the solution x^* of the system $F(x) = 0$.

Define e_1 and e_2 as follows

$$\begin{aligned} e_1 =& s_k - x^* = x_k^{(0)} - aF(x_k^{(0)}) - x^* = e_k^{(0)} - aF(x_k^{(0)}), \\ e_2 =& r_k - x^* = x_k^{(0)} + bF(x_k^{(0)}) - x^* = e_k^{(0)} + bF(x_k^{(0)}). \end{aligned} \tag{6}$$

We write the operator $[r_k, s_k; F]$ of method (3) using the Taylor series expansion of an operator of the form $[x+h, x; F]$, considering $x = r_k$, $x+h = s_k$ and $h = s_k - r_k = e_1 - e_2$. Thus we obtain

$$[r_k, s_k; F] = F'(r_k) + \frac{1}{2}F''(r_k)(e_1 - e_2) + \frac{1}{6}F'''(r_k)(e_1 - e_2)^2 + O((e_1 - e_2)^3) \tag{7}$$

where

$$\begin{aligned} F'(r_k) &= F'(x_k^{(0)} + bF(x_k^{(0)})) = F'(x_k^{(0)}) + bF''(x_k^{(0)})F(x_k^{(0)}) + \frac{1}{2}b^2F'''(x_k^{(0)})F(x_k^{(0)}) + \ldots, \\ F''(r_k) &= F''(x_k^{(0)}) + bF'''(x_k^{(0)})F(x_k^{(0)}) + \frac{1}{2}b^2F^{(iv)}(x_k^{(0)})F(x_k^{(0)}) + \ldots. \end{aligned} \tag{8}$$

Now, to be able to substitute in the expression (8), let's consider the Taylor expansion of $F(x_k^{(0)})$ around x^*:

$$F(x_k^{(0)}) = \Gamma\left(e_k + A_2e_k^2 + A_3e_k^3 + A_4e_k^4 + A_5e_k^5 + A_6e_k^6 + O\left(e_k^7\right)\right), \tag{9}$$

where $\Gamma = F'(x^*)$, $e_k = e_k^{(0)} = x_k^{(0)} - x^*$ and $A_j = \frac{1}{j!}F'(x^*)^{-1}F^{(j)}(x^*) \in L_j(\mathbb{R}^n, \mathbb{R}^n)$ with $L_j(\mathbb{R}^n, \mathbb{R}^n)$ the set of j-linear functions of bounded functions, $j = 1, 2, 3, \ldots$.

Then, we substitute the derivatives of $F(x_k^{(0)})$ in an environment of x^* in the two equations of (8) and, taking into account that $e_1 = e_k^{(0)} - aF(x_k^{(0)})$ and $e_2 = e_k^{(0)} + bF(x_k^{(0)})$, we substitute the resulting expressions in (7) to obtain the error equation for the divided difference operator:

$$[r_k, s_k; F] = \Gamma\left(I + (-a\Gamma + b\Gamma + 2I)A_2e_k + \left(\Gamma A_2^2(b-a) - 3(a\Gamma - I)(b\Gamma + I)A_3\right)e_k^2 + O\left(e_k^3\right)\right). \tag{10}$$

Then, we can obtain the error equation for its inverse operator, in a similar way as in [12] (see equations (2) and (3)). So by performing a step of (3) using divided differences and denoting I the identity matrix of order n, we get:

$$\begin{aligned} e_{k+1} = e_k^{(1)} &= x_k^{(1)} - x^* = x_k^{(0)} - x^* - [r_k, s_k; F]^{-1}F(x_k^{(0)}) \\ &= ((b-a)\Gamma + I)A_2e_k^2 + O(e_k^3). \end{aligned} \tag{11}$$

In this case, the method has order $p = 2$.

- For $m = 2$ we are facing a two-step method. Calculating the expression of the second divided difference of the schema (3) and the expansion of $F(x_k^{(1)})$, we will obtain the equation for the error of the method (3) for $m = 2$.

  Let $e_k^{(1)} = x_k^{(1)} - x^*$ be the error in the k-th approximation to the solution x^* of the system $F(x) = 0$ of the first step of the generalized 2-step Steffensen method with frozen second divided difference. Define e_1 and e_2 as follows

$$\begin{aligned} e_1 =& u_k - x^* = x_k^{(1)} - cF(x_k^{(1)}) - x^* = e_k^{(1)} - cF(x_k^{(1)}), \\ e_2 =& v_k - x^* = x_k^{(1)} + dF(x_k^{(1)}) - x^* = e_k^{(1)} + dF(x_k^{(1)}). \end{aligned} \tag{12}$$

We write the operator $[v_k, u_k; F]$ of the method (3) using the Taylor series expansion of an operator of the form $[x + h, x; F]$, considering, in this case, $x = v_k$, $x + h = u_k$ and $h = u_k - v_k = e_1 - e_2$:

$$[v_k, u_k; F] = F'(v_k) + \frac{1}{2}F''(v_k)(e_1 - e_2) + \frac{1}{6}F'''(v_k)(e_1 - e_2)^2 + O((e_1 - e_2)^3) \tag{13}$$

where

$$\begin{aligned} &F'(v_k) = F'(x_k^{(1)} + dF(x_k^{(1)})) = F'(x_k^{(1)}) + dF''(x_k^{(1)})F(x_k^{(1)}) + \frac{1}{2}d^2F'''(x_k^{(1)})F(x_k^{(1)}) + \dots, \\ &F''(v_k) = F''(x_k^{(1)}) + dF'''(x_k^{(1)})F(x_k^{(1)}) + \dots. \end{aligned} \tag{14}$$

To be able to substitute in the expression (14), we need, as in the first step, the Taylor expansion of $F(x_k^{(1)})$ around x^* and that of its derivatives. Substituting them in (14) and, in turn, using these expressions in (13), we get the expression of $[v_k, u_k, F]$.

$$\begin{aligned} [v_k, u_k; F] = \Gamma(((b - a)\Gamma + I)((d - c)\Gamma + 2I)A_2^2 e_k^2 \\ + ((d - c)\Gamma + 2I)((3(b - a(b\Gamma + I))\Gamma + 2I)A_3 \\ - \left(a^2\Gamma^2 - 2a\Gamma(b\Gamma + I) + b^2\Gamma^2 + 2b\Gamma + 2I\right) A_2^2)A_2 e_k^3 + O(e_k^4)). \end{aligned} \tag{15}$$

where, to simplify the notation, we have denoted $e_k = e_k^{(1)} = x_k^{(1)} - x^*$.

Then, we take the inverse of (15), in a similar way as in [12] (see equations (2) and (3)), and, using the Taylor series expansion of $F(x_k^{(1)})$, we obtain the error equation of the method (3) for $m = 2$:

$$\begin{aligned} e_{k+1} = e_k^{(2)} = x_k^{(2)} - x^* = x_k^{(1)} - x^* - [v_k, u_k; F]^{-1}F(x_k^{(1)}) \\ = ((b - a)\Gamma + I)^2((d - c)\Gamma + I)A_2^3 e_k^4 + O(e_k^5). \end{aligned} \tag{16}$$

That is, for $m = 2$, we obtain that the theoretical order of convergence of the method is $p = 2m = 4$.

- Assuming that the theorem holds up to $m - 1$, that is, that all the methods of j steps have order $p = 2j$ for $1 \leq j \leq m - 1$, we are going to prove that this theorem also holds for m. To do this, we use the error equation of the $(m - 1)$-steps of the form method (3) with order $p = 2(m - 1)$:

$$\begin{aligned} e_{k+1} &= e_k^{(m-1)} = x_k^{(m-1)} - x^* = x_k^{(m-2)} - x^* - [v_k, u_k; F]^{-1}F(x_k^{(m-2)}) \\ &= -((b - a)\Gamma + I)^{m-1}((d - c)\Gamma + 2I)^{m-3}((c - d)\Gamma - I)^{m-4}A_2^{2(m-1)-1}e_k^{2(m-1)} \\ &+ O(e_k^{2m-1}), \end{aligned} \tag{17}$$

where $e_k = e_k^{(m-2)}$ and $m \geq 4$.

As step m uses $[v_k, u_k; F]$, already developed in (15), and $F(x_k^{(m-1)})$ for the calculation of $x_k^{(m)}$, we only need to perform the Taylor series expansion of $F(x_k^{(m-1)})$ around x^*:

$$F(x_k^{(m-1)}) = (-1)^{m-2}((b-a)\Gamma+I)^{m-1}((c-d)\Gamma-2I)^{m-3}((c-d)\Gamma-I)A_2^{2(m-1)-1}e^{2(m-1)}.$$

Substituting this development and the inverse expression (15) in the m-th step of the method (3), we obtain the following error equation for the generalized m-step Steffensen's method with frozen divided second difference:

$$\begin{aligned} e_{k+1} = e_k^{(m)} = x_k^{(m)} - x^* &= x_k^{(m-1)} - x^* - [v_k, u_k; F]^{-1}F(x_k^{(m-1)}) \\ &= -((b-a)\Gamma+I)^{m}((d-c)\Gamma+2I)^{m-2}((c-d)\Gamma-I)^{m-3}A_2^{2m-1}e_k^{2m} + O(e_k^{2m+1}), \end{aligned} \tag{18}$$

where $m \geq 3$ is the number of steps we perform in the method. Then, observing the error equation, we deduce that for m steps the result is also verified since the order of convergence is $p = 2m$. Therefore, the theorem is proved for all number of steps m.

□

3. Computational efficiency

To complete the study of the new multistep iterative method for nonlinear systems of equations, denoted by $S2_m$ presented in the previous section, we analyze the computational cost of the methods in order to compare their efficiency with the family of methods given by $S1_m$. To do this, we use the computational efficiency index (*CEI*), defined in [2],

$$CEI(\mu_0, \mu_1, n) = p^{\frac{1}{C(\mu_0,\mu_1,n)}}, \tag{19}$$

where p is the order of convergence, n is the dimension of the problem and C is the computational cost per iteration. For a system of n nonlinear equations and n unknowns, the expression for C is

$$C(\mu_0, \mu_1, n) = a_0(n)\mu_0 + a_1(n)\mu_1 + p(n), \tag{20}$$

where $a_0(n)$ and $a_1(n)$ is the number of functional evaluations of $F(x)$ and $F'(x)$, respectively, per iteration, $p(n)$ is the number of products/quotients needed per iteration and μ_0 and μ_1 are the ratios between the products/quotients and the functional evaluations of F and $F'(x)$, respectively.

On the other hand, in order to study the efficiency by taking into account the number of iterations performed by each method, we will also use the total computational efficiency index, defined previously in [10] by the following expression:

$$TCEI(m, n, k, \mu_0, \mu_1) = p^{\frac{1}{kC(\mu_0,\mu_1,n)}}, \tag{21}$$

where k is the number of iterations performed by an iterative method in order to reach a established stopping criteria, p the convergence order and C is the computational cost given in (20). Note that this index depends on the system to be solved, the required tolerance and the initial estimate chosen.

We notice that an iteration on a system of size n requires n functional evaluations of a vector function $F(x)$, n^2 functional evaluations for obtaining $F'(x)$, $n^2 - n$ functional evaluations and n^2 quotients in the computation of a first-order divided difference operator of the form $[x, y; F]$, $\frac{1}{3}n^3 - \frac{1}{3}n$ products/quotients in decomposition LU and n^2 products/quotients in the solution of 2 triangular systems used to solve a linear system. Note that if we need to solve m linear systems with the same coefficient matrix the computational cost is $\frac{1}{3}n^3 + mn^2 - \frac{1}{3}n$ operations since only one LU decomposition is performed and the two triangular systems are solved m times.

In Table 5 we can see the expression for the computational cost of iterative methods $S1_m$ and $S2_m$. Observe that, as Steffensen-like methods do not use derivatives, the ratio μ_1 does not appear.

Table 1. Computacional cost

Method	$C(\mu_0, \mu_1, n)$	Notation
$S1_m$	$(n^2 + (m+1)n)\mu_0 + \frac{1}{3}(n^3 - n) + (m+1)n^2 + 2n$	N1
$S2_m$	$(2n^2 + (m+2)n)\mu_0 + \frac{2}{3}(n^3 - n) + (m+2)n^2 + 4n$	N2

3.1. Comparing methods

We are interested in comparing the computational efficiency between the multistep iterative methods $S1_m$ and $S2_m$ whose iteration functions are given by (2) and (3). So, we notice that although $S1_m$ has convergence order $m+1$, we can obtain convergence order equal to or even higher than m with $S2 : m'$ with $m' < m$. In other words, $S2 : m'$ reaches the same order as $S1_m$ with about half the steps, specifically $m' \geq \frac{m+1}{2}$.

Table 2. Relation between number of steps and convergence order for methods $S1_m$ and $S2_m$

m	$S1_m$	$S2_m$	$S2_{m-1}$	$S2_{m-2}$	$S2_{m-3}$	$S2_{m-4}$
2	3	4				
3	4	6	4			
4	5	8	6			
5	6	10	8	6		
6	7	12	10	8		
7	8	14	12	10	8	
8	9	16	14	12	10	
9	10	18	16	14	12	10
10	11	20	18	16	14	12

In Table 2 we can see for a number of steps from 2 to 10, the different possibilities for obtaining same or higher order with $S2_m$ but with less steps. For instance while with $m = 7$, $S1_m$ reach order 8, we can perform $m - 3 = 4$ steps with $S2_m$ for reaching the same order. In other words, for $m = 7$ we can perform until 3 less steps for getting order higher with $S2_m$ than with $S1_m$. This is very important in order to compare the efficiency index of the families of iterative method defined in the previous section, because we take into account the total computational efficiency and we can assume that as the order of convergence of a method increases, the total number of iterations decreases.

We perform an exhaustive study for comparing the efficiency of both families of methods in the following term and by taking the ratio $\mu_0 = 2$.

Theorem 3.1. *Let be m the number of steps we perform with the iterative method $S1_m$ for solving a nonlinear system of size $n \leq 46$ and performing k iterations. Then, if an iterative method of family $S2_m$ performs a maximum of $k-1$ iterations it is verified that:*

(1) If $k \leq 5$, $n \leq 15$ and $m \geq 5$ then $TCEI_{S2_m}(m, n, k-1) > TCEI_{S1_m}(m, n, k)$
(2) If $k \leq 4$, $n \leq 30$ and $m \geq 5$ then $TCEI_{S2_m}(m, n, k-1) > TCEI_{S1_m}(m, n, k)$
(3) If $k \leq 4$, $n \leq 40$ and $m \geq 7$ then $TCEI_{S2_m}(m, n, k-1) > TCEI_{S1_m}(m, n, k)$
(4) If $k \leq 4$, $n \leq 15$ and $m \geq 5$ then $TCEI_{S2_m}(m-1, n, k-1) > TCEI_{S1_m}(m, n, k)$
(5) If $k \leq 4$, $n \leq 30$ and $m \geq 7$ then $TCEI_{S2_m}(m-1, n, k-1) > TCEI_{S1_m}(m, n, k)$
(6) If $k \leq 4$, $n \leq 46$ and $m \geq 10$ then $TCEI_{S2_m}(m-1, n, k-1) > TCEI_{S1_m}(m, n, k)$

Proof. By applying the definition of total computational efficiency index, (21), we have that if $N1$ ans $N2$ are the total number of operations performed per iteration by $S1_m$ and $S2_m$ respectively, see Table 5, while k and $k-j$ are the iterations performed by these methods, we have that $S2_m$ it is more efficient than $S1_m$ if

$$\Big(2m\Big)^{\frac{1}{(k-j)N2}} > \Big(m+1\Big)^{\frac{1}{kN1}}$$

by taking logarithms it follows

$$\frac{1}{(k-j)N2}log(2m) > \frac{1}{kN1}log(m+1)$$

that is,

$$log(2m) > \frac{(k-j)}{k}\frac{N2}{N1}log(m+1) \tag{22}$$

we take the exponential function in both sides and by denoting the second part of (22) by function $A(m,n,k)$ we consider the difference

$$E(m,n,k) = 2m - exp(A(m,n,k))$$

that is decreasing in terms of n. We observe Figure 1, on the right, where we have represented $E(5,n,k)$ in function of k with $j=1$ and for different values of $n \leq 15$. So, we can appreciate that in all cases the function is positive for $k \leq 5$, that means that the corresponding iterative method derived from family $S2_m$ is more efficient than the one corresponding to $S1_m$ if performs at least one less iteration, that is very probable because the order is higher than the corresponding method of $S1_m$.

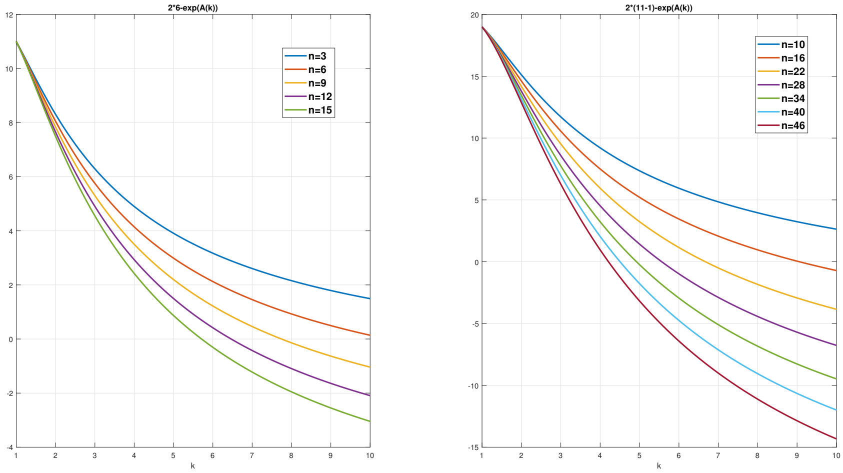


Figure 1. Limit values of m for having $S2_m$ more efficient than $S1_m$ associated to case (1) of theorem 2, right and case (7) on the left side.

Then, in case (1) we fix $n=15$, $m=5$ and $j=1$ so it follows that $E(5,15,k) > 0$ and the function remains positive until the root of the nonlinear equation $E(5,15,k) = 10 - exp(A(5,15,k)) = 0$, that it is $k = 5.1784$, so (1) is proved for $m=5$.

Obviously $E(m,n,k)$ is an increasing function in m so by performing the same process with $m =$

$6, \ldots, 10$ we find always a root greater than five, then this case is proved.

In case (2) we take $n = 30$, now the equation to solve is $E(5, 30, k) = 10 - exp(A(5, 30, k)) = 0$ whose root is 4.3121. Next, for $n = 40$, $m = 7$, the value 4.2867 gave us the limit for case (3). Finally in case (4) the value is 4.0847. Now, in cases (5) to (7) notice that the equation needed to solve is

$$log(2(m-1)) > \frac{(k-1)}{k}\frac{N2}{N1}log(m+1) \tag{23}$$

or equivalently

$$E(m, n, k) = 2(m-1) - exp(A(m, n, k)).$$

With a similar reasoning to the previous cases and taking the appropriate parameters we can obtain the limit of iterations performed by $S2_M$ in order to be more efficient than $S1_M$. These values are 4.31, 4.1181 and 4.0383, this last value can be observed in the left part of Figure 1.

□

Theorem 3.2. *Let be m the number of steps we perform with the iterative method $S1_m$ for solving a nonlinear system of size $n \leq 1000$ and performing k iterations, then if we find an iterative method of family $S2_m$ performing a maximum of $k-2$ iterations it is verified that:*

(1) If $k \leq 7$, $n \leq 50$ and $m \geq 5$ then $TCEI_{S2_m}(m, n, k-2) > TCEI_{S1_m}(m, n, k)$
(2) If $k \leq 7$, $n \leq 45$ and $m \geq 7$ then $TCEI_{S2_m}(m-1, n, k-2) > TCEI_{S1_m}(m, n, k)$
(3) If $k \leq 6$, $n \leq 150$ and $m \geq 4$ then $TCEI_{S2_m}(m, n, k-2) > TCEI_{S1_m}(m, n, k)$
(4) If $k \leq 5$, $\forall n \in N$ and $m \geq 4$ then $TCEI_{S2_m}(m, n, k-2) > TCEI_{S1_m}(m, n, k)$
(5) If $k \leq 5$, $n \leq 150$ and $m \geq 5$ then $TCEI_{S2_m}(m-1, n, k-2) > TCEI_{S1_m}(m, n, k)$
(6) If $k \leq 5$, $n \leq 1000$ and $m \geq 7$ then $TCEI_{S2_m}(m-1, n, k-2) > TCEI_{S1_m}(m, n, k)$

Proof. The proof is similar to the corresponding of previous theorem so it is omitted. The values limited for each case are $7.0754, 7.0214, 6.0569, 5.6, 5.1589$ and 5.0575. Cases (2) and (7) can be seen in the left and right sides of Figure 2, respectively.

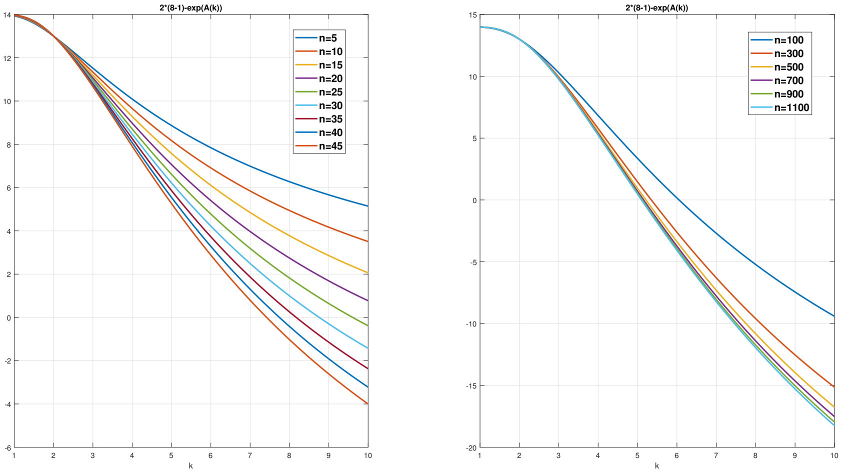


Figure 2. Limit values of m for having $S2_m$ more efficient than $S1_m$ associated to cases (2) and (6) of Theorem 3.

□

4. Iterative schemes with memory

This section focuses on the construction of two memory multistep methods based on the non-memory generalized m-step Steffensen method with frozen divided second difference (3). To do this, we will introduce memory as they do in [15] with the $S1_m$ method.

In Section 2 we have studied the generalized multistep Steffensen's method freezing the divided difference of the second step and we have deduced that, for m steps, this method has convergence order $2m$ and that its error equation is of the form:

$$\begin{aligned} e_{k+1} = e_k^{(m)} = x_k^{(m)} - x^* &= x_k^{(m-1)} - x^* - [v_k, u_k; F]^{-1} F(x_k^{(m-1)}) \\ &= -(I - a\Gamma + b\Gamma)^m (2I - c\Gamma + d\Gamma)^{m-2} (-I + c\Gamma - d\Gamma)^{m-3} A_2^{2m-1} e_k^{2m} + O(e_k^{2m+1}), \end{aligned} \tag{24}$$

where $m \geq 3$ is the number of steps we carry out in (3).

We want to increase the speed of convergence of this method without performing new functional evaluations. To do this, we are going to study which are the appropriate expressions for the accelerating parameters a, b, c and d. As we can see in the error equation (24), we have many options to increase the order of convergence of the method to $2m + 1$.

To achieve this order, we need to cancel one of the three terms of (24). However, it is not possible to make any of them totally zero since we do not know $\Gamma = F'(x^*)$. Therefore, we are going to try to reduce the influence of these factors on the error equation as much as possible by defining the parameters a, b, c and d.

In this case we will try to minimize the impact of more than one term at a time to accelerate the convergence as much as possible. This is possible because the constants a and b allow the convergence of the first step to be accelerated while the constants c and d accelerate that of the rest of the steps, because the divided difference of the second step is the one used in the rest of the steps. Although there are several ways to add memory, in this article we will only present one way to increase the convergence order of(3).

From the error equation (24), it is clear that we will obtain convergence order $2m + 1$ if we find a solution of the matrix equation for $b - a$ such that:

$$I - a\Gamma + b\Gamma = I + (b - a)[F'(x^*)] = 0. \tag{25}$$

We will also get order $2m + 1$ if we find a solution of the matrix equation for $d - c$ such that:

$$2I - c\Gamma + d\Gamma = 2I + (d - c)[F'(x^*)] = 0. \tag{26}$$

As we have explained before, we will not be able to find parameter values that cancel (25) and (26) and achieve $2m + 1$ convergence order.

In order to accelerate the convergence of the method (3) as much as possible, we use an appropriate approximation B_k of $b - a$ and a C_k of $d - c$ to decrease the first and second term of the error equation. Thus, using data from the current and previous iterations to approximate $[F'(x^*)]^{-1}$ in (25) and (26), we assume that the parametric matrices B_k and C_k have the following form:

$$\begin{aligned} C_k = b - a = -[r_{k-1}, s_{k-1}; F]^{-1} \approx -[F'(x^*)]^{-1}, \quad \text{for} \quad k \geq 1, \\ B_k = d - c = -2[v_{k-1}, u_{k-1}; F]^{-1} \approx -2[F'(x^*)]^{-1}, \quad \text{for} \quad k \geq 1. \end{aligned} \tag{27}$$

We propose an iterative method with memory from the method (3) defined by the following scheme:

$$S2_mM \begin{cases} x_0 \in \Omega, \\ x_k^{(1)} = x_k^{(0)} - Q_k^{-1}F(x_k^{(0)}), \\ x_k^{(2)} = x_k^{(1)} - P_k^{-1}F(x_k^{(1)}), \\ \quad\vdots \\ x_k^{(m-1)} = x_k^{(m-2)} - P_k^{-1}F(x_k^{(m-2)}), \\ x_{k+1}^{(m)} = x_k^{(m-1)} - P_k^{-1}F(x_k^{(m-1)}), \quad k \geq 0, \end{cases} \tag{28}$$

where $Q_k = [r_k, s_k; F]$, $s_k = x_k^{(0)} - \gamma_1 C_k F(x_k^{(0)})$, $r_k = x_k^{(0)} + \delta_1 C_k F(x_k^{(0)})$, $P_k = [u_k, v_k; F]$, $u_k = x_k^{(1)} - 2\gamma_2 B_k F(x_k^{(1)})$ and $v_k = x_k^{(1)} + 2\delta_2 B_k F(x_k^{(1)})$. Here, γ_1 and δ_1 are arbitrary constants that must satisfy the condition $\delta_1 - \gamma_1 = I$ as in the first section. In turn, γ_2 and δ_2 are constants that must also satisfy the condition $\delta_2 - \gamma_2 = I$.

We know that this method with memory increases to some extent the convergence order, $2m$ of method (3), but we don't know what the convergence order will be. Therefore, we present the following result that will allow us to know the order of convergence of (28) depending on the number of steps m.

Theorem 4.1. *Let $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ a sufficiently differentiable function in a neighbourhood $\Omega \subset \mathbb{R}^n$ of a point x^* such that $F(x^*) = 0$. We assume that $F'(x)$ is a continuous and non-singular at x^* and that the arrays B_k and C_k in the iterative method (28) are recursively calculated by the expressions (27). Then the method given by (28) converges to x^* with convergence order $m + \sqrt{m^2 + 2m - 2}$ for $m \geq 1$.*

Proof. Let $\{x_k\}$ be the sequence of approximations which converges to the solution x^* with R-order $\geq r$ and $e_k = x_k - x^*$, then we can write

$$e_{k+1} \sim D_{k,r} e_k^r \tag{29}$$

where $\{D_{k,r}\}$ is a sequence that tends to the constant asymptotic error D_r of the method when $k \to \infty$. The notation $s \sim t$ symbolizes that they have magnitudes of the same order.

Therefore, we will have

$$e_{k+1} \sim D_{k,r}(D_{k-1,r} e_{k-1}^r)^r = D_{k,r} D_{k-1,r}^r e_{k-1}^{r^2}. \tag{30}$$

Using the expansion of the inverse of (10) for $k-1$, we get:

$$[v_{k-1}, u_{k-1}; F]^{-1} = [I - (2A_2 + (b-a)\Gamma A_2)e_{k-1} + O(e_{k-1}^2)]\Gamma^{-1}.$$

Thus we obtain the expressions of the C_k and B_k arrays of the iterative method (28):

$$\begin{aligned} b - a = C_k &= -[r_{k-1}, s_{k-1}; F]^{-1} = -[I - (2A_2 + C_{k-1}\Gamma A_2)e_{k-1} + O(e_{k-1}^2)]\Gamma^{-1}, \\ d - c = B_k &= -2[v_{k-1}, u_{k-1}; F]^{-1} = -2[I - (2A_2 + B_{k-1}\Gamma A_2)e_{k-1} + O(e_{k-1}^2)]\Gamma^{-1}. \end{aligned} \tag{31}$$

From these developments, we obtain:

$$\begin{aligned} I + \Gamma C_k &= (2I + Ck - 1\Gamma)A_2 e_{k-1} + O(e_{k-1}^2) \sim e_{k-1}, \\ 2I + B_k\Gamma &= 2(2I + B_{k-1}\Gamma)A_2 e_{k-1} + O(e_{k-1}^2) \sim e_{k-1}. \end{aligned} \tag{32}$$

We rewrite the error equation (24) using the accelerator parameters,

$$e_{k+1} = (I + C_k\Gamma)^m(2I + B_k\Gamma)^{m-2}(I + B_k\Gamma)^{m-3}A_2^{2m-1}e_k^{2m} + O(e_k^{2m+1}), \tag{33}$$

and we substitute (32) in (33) to obtain:

$$e_{k+1} \sim e_{k-1}^m e_{k-1}^{m-2} e_k^{2m} \sim e_{k-1}^m e_{k-1}^{m-2}(e_{k-1}^r)^{2m} \sim e_{k-1}^{2mr+2m-2}. \tag{34}$$

We know from the equation (30) that $e_{k+1} \sim e_{k-1}^{r^2}$. Therefore, comparing the exponents of (30) and (34), we obtain

$$r^2 = 2mr + 2m - 2. \tag{35}$$

We solve the quadratic equation (35) and obtain the positive real root $m + \sqrt{m^2 + 2m - 2}$, which is the lower bound of the R-order of convergence of the method (28).

□

5. Numerical results

In this section we will use the above proposed iterative methods $S1_m$, $S2_m$ and $S2_mM$ to approximate the solution of several nonlinear systems of equations in order to confirm the theoretical results and verify their efficiency. Furthermore, we will use the numerical examples to compare these three methods to the two memory versions of the $S1_m$ method, presented in [15], which we will denote as $S1_mM_1$ and $S2_mM_2$.

To better understand all the methods that we will use, we present the following table with some of their characteristics:

Table 3. Theoretical order and computational cost of the methods

Method	$Order$	$C(\mu_0, \mu_1, n)$
$S1_m$	$m + 1$	$(n^2 + (m+1)n)\mu_0 + \frac{1}{3}(n^3 - n) + (m+1)n^2 + 2n$
$S1_mM_1$	$\frac{(m+1)+\sqrt{(m+1)^2+4}}{2}$	$(n^2 + (m+1)n)\mu_0 + \frac{1}{3}(n^3 - n) + (m+2)n^2 + 2n$
$S1_mM_2$	$\frac{(m+1)+\sqrt{(m+1)^2+4(m-1)}}{2}$	$(n^2 + (m+1)n)\mu_0 + \frac{1}{3}(n^3 - n) + (m+2)n^2 + 2n$
$S2_m$	$2m$	$(2n^2 + (m+2)n)\mu_0 + \frac{2}{3}(n^3 - n) + (m+2)n^2 + 4n$
$S2_mM$	$m + \sqrt{m^2 + 2m - 2}$	$(2n^2 + (m+2)n)\mu_0 + \frac{2}{3}(n^3 - n) + (m+4)n^2 + 4n$

where n is the size of the system to be solved, m the number of steps per iteration of the method and μ_0 the ratio between functional evaluations and products.

For the computational calculations, the software used has been RMatlab 2019b and the processor used has been $Intel(R)\ Core(TM)\ i7 - 9700\ CPU@3.00GHz$ with $32GB$ of RAM.

In the three examples studied, we compare the approximation obtained, the norm of the distance between the last two iterations, the number of iterations needed to satisfy the tolerance required in each example, the CEI and $TCEI$ of $S1_m$ and $S2_m$, assuming a ratio $\mu_0 = 2$, the computational time and the approximate computational convergence order $ACOC$, [7], which has the following expression:

$$ACOC = \rho \approx \frac{\ln\left(\|x_{k+1} - x_k\|/\|x_k - x_{k-1}\|\right)}{\ln\left(\|x_k - x_{k-1}\|/\|x_{k-1} - x_{k-2}\|\right)}. \tag{36}$$

Example 1: Model of a photomultiplier device

A photomultiplier tube (PMT) is an electronic device that receives light and transforms it into (amplified) electric current, allowing to detect light intensities as small as a photon. The light arrives at the cathode which emits electrons that are multiplied in cascade by different electrodes called dynodes. Finally, the electrons are captured by the last electrode, the anode, generating a measurable current. PMT devices are widely used in different applications, in particular in medicine, they are utilised in diagnosis (PET/CT scanners, gamma camera, Anger camera for SPECT scanners, real time PCRs) as well as for therapy monitorisation (Compton camera, planar image devices) [9].

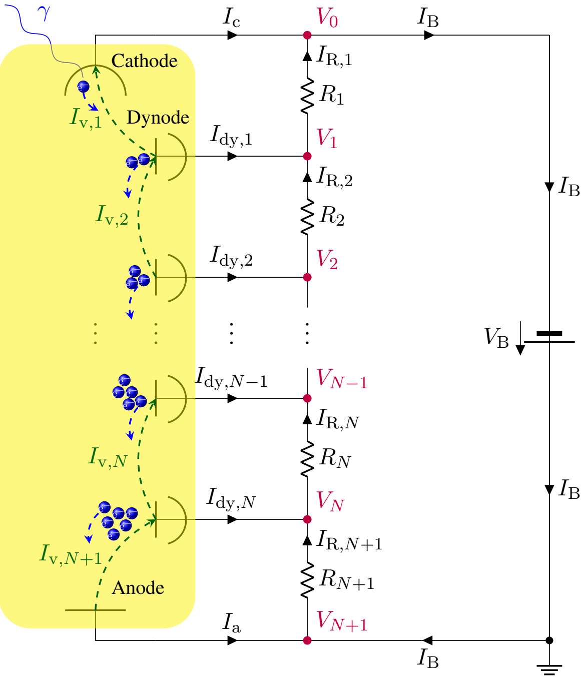


Figure 3. Schematic of an N-stage PMT connected to a voltage bias V_B in negative polarity mode and a chain of resistors R_i with $i \in [1, N+1]$. The vacuum tube (in yellow) consists of one photocathode ($i = 0$), N dynodes ($i = 1, \ldots, N$), and one anode ($i = N+1$). The electrode voltages $V(i)$ are shown in purple for $i \in [0, N+1]$. The power supply bias current is I_B. The electrode currents are I_c, $I_{\text{dy},i}, i \in [1, N]$, and I_a, whereas the current flowing through the resistors is labeled as $I_{\text{R},i}, i \in [1, N+1]$. A photon ($\gamma$), shown in blue, generates a photoelectron in the cathode. The electrons (small balls) flow in vacuum in the direction of the dashed blue arrow. The dashed green arrows show the sign convention of the definition of the inter-electrode vacuum currents $I_{\text{v},i}$. Adapted from [11].

Figure 3 shows a simple model of a PMT device. Using Kirchhoff's current law and Ohm's law, the equations governing the PMT model can be formulated [11] as follows for $i = 1 \ldots N+1$:

$$I_{\text{R},i} + I_{\text{v},i} = I_\text{B}, \tag{37}$$

$$V_i - V_{i-1} = I_{\text{R},i} \cdot R_i, \tag{38}$$

$$I_{\text{v},i} = I_\text{k}\eta_i \prod_{j=1}^{i-1} k_j [V_j - V_{j-1}]^{\alpha_j}, \tag{39}$$

where (39) for $i = 1$ is defined instead as just $I_{\text{v},1} = \eta_1 I_\text{k}$.

In order to determine the state of the PMT we have to solve a system of N non-linear equations in terms of the electrode voltages V_i $(i = 1, \dots, N)$:

$$\frac{V_i - V_{i-1}}{R_i} = \frac{V_{i+1} - V_i}{R_{i+1}} + I_{\mathrm{v},i+1} - I_{\mathrm{v},i} \qquad i = 1, \dots, N \tag{40}$$

where $V_0 = -V_b$ and $V_{N+1} = 0$.

The model shown in Figure 3 can be solved applying the presented iterative methods to the former equations with $N = 8$, $\eta_i = 1$ for $i = 1, \dots, N+1$, $R_i = 330\,\mathrm{k}\Omega$ for $i = 1, \dots, N$, $R_{N+1} = 160\,\mathrm{k}\Omega$, $\alpha = 0.881$ and $k = 0.0936$.

This system of non-linear equations is solved considering the inner node voltages $V_i,\ i = 1, \dots, N$ as unknowns. As initial estimation we take, for the unknowns V_i, N values linearly interpolated between $-V_b$ and 0, excluding both extremes.

We solve the systems for two different parameter values $I_k = 10\,\mathrm{pA}$ and $I_k = 100\,\mathrm{pA}$, with tolerance 10^{-3}. We have taken $m = 1$, $a = c = 1.1$, $\gamma = 1.1 \cdot I$, $b = d = 2.1$ and $\delta = 2.1 \cdot I$ in the corresponding methods. The results are shown in Table 4.

Table 4. Results of the PMT model for different parameter values

	$I_k = 10\,\mathrm{pA}$			$I_k = 100\,\mathrm{pA}$		
Method	$ACOC$	$iter$	$\|f(x_k)\|$	$ACOC$	$iter$	$\|f(x_k)\|$
$S1_m$	1.5738	3	$1.0115e-11$	2.0141	4	$2.6454e-13$
$S1_mM_1$	2.5380	3	$1.3487e-18$	2.8882	4	$8.2955e-19$
$S1_mM_2$	2.2092	3	$1.0272e-18$	2.1271	5	$8.9457e-19$
$S2_m$	1.5738	3	$5.8366e-11$	2.0141	4	$3.3774e-11$
$S2_mM$	2.2889	3	$5.2732e-19$	2.1687	4	$1.5049e-18$

As I_k increases, so does the system nonlinearity, and the methods need more iterations to converge, but the error in terms of the norm of the residual and the ACOC have similar values.

Example 2: A differentiable operator

We consider the following system of *n* equations and *n* unknowns:

$$\begin{cases} x_i \sin(x_{i+1}) - 1 = 0, & 1 \leq i \leq n-1, \\ x_n \sin(x_1) - 1 = 0. \end{cases} \tag{41}$$

For this example we will use variable precision arithmetic with 850 digits iterating from two initial estimates $x^{(0)} = [1.1, \dots, 1.1]^t$ and $x_2^{(0)} = [1.4, \dots, 1.4]^t$ up to a distance between consecutive iterations less than the tolerance 10^{-200}. We have also solved Example (41) for $n = 30$ and $n = 50$ by varying the number of steps of all the methods from $m = 5$ to $m = 4$ to numerically check some of the conclusions of the Theorem 3.1 and Theorem 3.2. The numerical results are shown in Table 5.

As we know, most of these methods are generic and, therefore, have constants that we must determine. In this example, for $S1_m$ we have chosen $a = 1.1$ and $b = 2.1$, for $S2_m$ $a = c = 1.1$ and $c = d = 2.1$ and for $S1_mM_1$, $S1_mM_2$ and $S2_mM$ we have worked with $\gamma_1 = \gamma_2 = 1.1 \cdot I$ and $\delta_1 = \delta_2 = 2.1 \cdot I$ such that they verify $\delta_1 - \gamma_1 = \delta_2 - \gamma_2 = I$.

In all cases, we obtain as an approximation of the solution of the equation (41) the following vector $x_{k+1}^{(m)} = [1.11415714087193, \dots, 1.11415714087193, \dots]^T$.

For this example we see that the numerical results for each method are quite similar. The greatest difference observed between the methods is reflected in the ACOC of the method $S1_m$ and its variants

Table 5. Example 2 results for different sizes and steps

Method	n	m	$ACOC$	$iter$	$\Vert x_{k+1}^{(m)} - x_k^{(m)} \Vert$	CEI	$TCEI$	$time$
$S1_m$	30	5	5.999999	4	$3.229847e-648$	1.00010787	1.00002696	25.3906
$S1_mM_1$			6.161404	4	$5.126769e-789$	1.00010385	1.00002596	26.3594
$S1_mM_2$			6.701992	4	$4.682497e-817$	1.00010782	1.00002695	25.4219
$S2_m$			9.913360	3	$4.972061e-302$	1.00008102	1.00002707	43.8438
$S2_mM$			10.995597	3	$6.536994e-358$	1.00007933	1.00002619	43.5000
$S1_m$	50	4	5.000000	6	$1.614601e-926$	1.00002693	1.00000448	242.4063
$S1_mM_1$			5.192271	5	$2.414904e-1020$	1.00002646	1.00000529	212.7031
$S1_mM_2$			5.646294	5	$2.479962e-881$	1.00002750	1.00000550	206.9688
$S2_m$			8.000000	4	$3.4147252e-625$	1.00002396	1.00000599	305.2500
$S2_mM$			8.999502	4	$3.216070e-1037$	1.00001925	1.00000481	314.0781

with memory and the method $S2_m$ and its version with memory. On the other hand, we observe that the numerical results coincide with the theoretical ones since the $TCEI$ of $S2_m$ is greater than that of $S1_m$ and, in addition, the first method is more efficient than the other methods from the perspective of this efficiency index. We must also add that, in general, in $S1_m$ and $S1_mM$ fewer iterations are required than in the rest. Finally, as expected, we see that the methods that perform a greater number of iterations make less error in the approximation.

Example 3: A non-differentiable operator

In this example we solve a nonlinear integral equations that appears in some applied problems in electrostatic and electro magnetic problems, among many others situations ([6], [20]). We consider the nonlinear integral equation of Hammerstein type given by

$$[\mathcal{H}(x)](s) = x(s) - f(s) - \int_a^b G(s,t)(\, x(t)^3 + 5|x(t)|)\, dt, \quad s \in [a,b]. \tag{42}$$

where $a, b \in \mathbb{R}$, G is the Green function, $f(s) = 1, \forall s \in [0,1]$ and x is the solution to be obtained.
We transform the problem $\mathcal{H}(x) = 0$, where $\mathcal{H} : \Omega \subset \mathcal{C}[a,b] \to \mathcal{C}[a,b]$ into a nonlinear system of equations. To begin with, we approximate the given integral by a quadrature formula with the corresponding weighs, t_j and nodes $q_j, j = 1, 2, ..., n$. So, we have the following nonlinear system:

$$x_j = f_j + \sum_{i=1}^{n} p_{ji}\, (\lambda x(t)^3 + \sigma|x(t)|), \quad j = 1, 2, \ldots, n, \tag{43}$$

where

$$p_{ji} = q_i G(t_j, t_i) = \begin{cases} q_i \frac{(b-t_j)(t_i-a)}{b-a}, & i \leq j, \\ q_i \frac{(b-t_i)(t_j-a)}{b-a}, & i > j. \end{cases}$$

The interest of this example is to show the applicability of the methods seen in this article to a non-differentiable operator.

For this example we will use variable precision arithmetic with 400 digits iterating from an initial estimate $x^{(0)} = [-2.8, \ldots, -2.8]^t$ up to a distance between consecutive iterations less than the tolerance 10^{-150}. We have solved the Example (43) for a system size $n = 8$ and with a number of steps $m = 4$. The numerical results are shown in Table 6.
In this example we have used the same method constants as in the previous. We have obtained as an

Table 6.

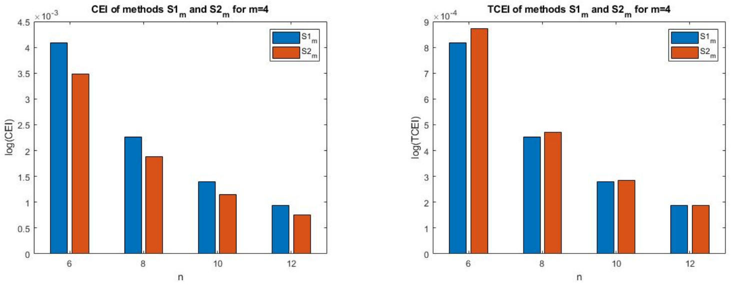
Method	$ACOC$	$iter$	$\|x_{k+1} - x_k\|$	$time$
$S1_m$	4.998270	5	$7.127831e - 333$	245.4375
$S1_m M_1$	5.187298	4	$1.034017e - 168$	195.2656
$S1_m M_2$	5.617349	4	$3.414139e - 175$	190.7188
$S2_m$	7.999571	4	$8.067708e - 223$	354.9844
$S2_m M$	8.609689	4	$4.298286e - 406$	348.5156

approximation to the solution of the equation (43) the following vector

$$x_{k+1}^{(m)} = [-3.307603, -3.029411, -2.890455, -2.759186, -2.611678, -2.611678, -2.759186, \ldots]^T.$$

In this second example we see numerical results similar to the previous one. Again the numerical method with the highest ACOC is $S2_m M$ followed by its non-memory version $S2_m$. In addition, both methods present a smaller error in the approximation if we take into account that both perform $k = 4$ iterations. In this example, to compare the CEI and $TCEI$ of the $S1_m$ and $S2_m$ methods, we have drawn the graphs 4.

In these figures we can see the importance of comparing the efficiency of the methods using the CEI or the $TCEI$. The second index is the one that seems most appropriate to talk about efficiency as it takes into account the number of iterations performed by each method and k is of great importance when calculating the computational cost. Therefore, for a system size n between 6 and 12, we can see in the graph 4 that the method $S2_m$ is more efficient than $S1_m$.



(a) *CEI* of methods $S1_m$ and $S2_m$ (b) *TCEI* of methods $S1_m$ and $S2_m$

Figure 4. Efficiendy of $S1_m$ and $S2_m$.

6. Dynamical study

In this section we analyze the behaviour of the methods $S1_m$ and $S2_m$ from the point of view of their dynamics. The idea is that the application of the methods to simple equations will provide insights on their behaviour in general situations. This approach has been widely used in the literature since the onset of complex dynamics.

Consider an function $G : \overline{\mathbb{C}} \longrightarrow \overline{\mathbb{C}}$ in the complex plane. The orbit of a point $z \in \mathbb{C}$ is the sequence of images $\{z_n\}_{n=0}^{\infty} = \{z, \ G(z), \ G^2(z), \ldots\}$. The points of the complex plane are classified according to the behaviour of their orbits. If the orbit converges to a limit z^*, then $G(z^*) = z^*$, so that z^* is a fixed point of G. A fixed point z^* of G is attracting if $|G(z^*)| < 1$, repelling if $|G(z^*)| > 1$ and super-attracting if $G'(z^*) = 0$. The points whose orbit converges to a fixed point form the attraction basin of this fixed point. The dynamic plane is obtained by coloring differently each basin. We have considered that an orbit $\{z_n\}$ converges if the distance between two consecutive iterates is less than 10^{-6} before

200 iterations and that diverges to infinity if $|z_n| > 10^6$ for some n. Other cases can include slow convergence or divergence, but they are anyway undesired dynamics. We have not observed periodicity in the considered algorithms.

Let us begin with the standard second degree polynomial $f(z) = z^2 - 1$ in the complex plane $\mathbb{C}$, whose roots are ± 1. The application of the generalized multistep Steffensen method $S1_m$ with $m = 1$, which coincides with the generalized Steffensen method, to $f(z)$ results in the rational iteration function

$$G(z) = \frac{(a-b)z^3 - z^2 - (a-b)z - 1}{(a-b)z^2 - 2z - a + b}$$

which depends only on the parameters difference $a - b$. The fixed point equation $G(z) - z = 0$ is

$$\frac{z^2 - 1}{(a-b)z^2 - 2z - a + b} = 0,$$

so that, its only fixed points are $z = \pm 1$, independently of the parameter values.

If $a = b$, $G(z)$ is the iteration function of Newton's method, whose dynamics is well known. In this case, the divided differences are the central ones, which exactly approximate the derivative for second degree polynomials. The attraction basins are the semiplanes $Re(z) < 0$ and $Re(z) > 0$, and the imaginary axis is the Julia set. The dynamics is richer when $a \neq b$.

The iteration function of $S1_m$ with $m = 2$ is slightly more involved in the general case, but still it depends only on the difference $e = a - b$:

$$G(z) = \frac{e^3z^7 - 4e^2z^6 - (3e^3 - 6e)z^5 + (6e^2 - 3)z^4 + (3e^3)z^3 - 6z^2 - (e^3 + 6e)z - 2e^2 + 1}{e^3z^6 - 6e^2z^5 - (3e^3 - 12e)z^4 + (12e^2 - 8)z^3 + (3e^3 - 12e)z^2 - 6e^2z - e^3}.$$

For $a - b = 1$, it results

$$G(z) = \frac{z^7 - 4z^6 + 3z^5 + 3z^4 + 3z^3 - 6z^2 - 7z - 1}{z^6 - 6z^5 + 9z^4 + 4z^3 - 9z^2 - 6z - 1}.$$

There are 6 fixed points, the two super-attracting roots and four more repelling points.

For the simpler case $a = b$, the iteration function

$$G(z) = \frac{3z^4 + 6z^2 - 1}{8z^3}$$

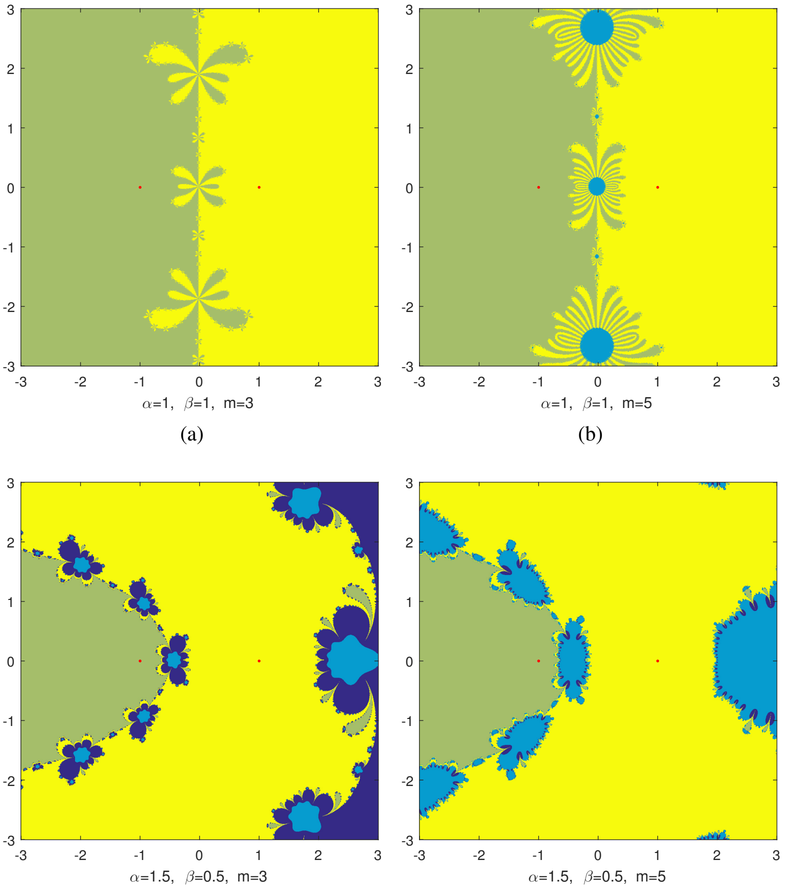
has four fixed points, ± 1 and $\pm\sqrt{5}/5$. The fist two fixed points, the roots of f, are super-attracting, $G'(z) = 0$, whereas the others are repelling, $|G'(z)| = 6 > 1$, so that they do not affect the convergence of the method.

For $m = 2$, the iteration function of $S1_m$ with $a - b = 1$ applied to $z^2 - 1$ presents 14 fixed points, two of them, the roots, are super-attracting and the rest repelling.

For $a = b$, the iteration function results

$$G(z) = \frac{39z^8 + 124z^6 - 46z^4 + 12z^2 - 1}{128z^7}.$$

It has 8 fixed points, the roots, which are super-attracting, and 6 more, which are repelling. These examples show that the generalized multistep Steffensen method has good dynamical behaviour, because the only attracting fixed points are the roots of the polynomial. Figure 2 shows that the dynamics is richer for $a \neq b$, so that, the other case is preferable from the point of view of stability.



(a)

(b)

(c)

(d)

Figure 5. Dynamical planes for the multistep generalized Steffensen method $S1_m$ for $z^2 - 1$ with different parameter values and number of steps. The red dots are the roots of the polynomial and their basins are coloured in green and yellow. The points whose orbits diverge to infinity are in deep blue and the light blue represents the points whose orbit does not fulfill the convergence criteria.

Let us now analyze the dynamics of method $S2_m$ for the quadratic polynomial $z^2 - 1$. For the shake of simplicity, we consider $a = c$ and $b = d$. As before, the iteration function $G(z)$ depends only on the difference $a - b$, although it is more complex than for $S2_m$. For example, for $m = 2$, $G(z) = p(z)/q(z)$, where $p(z)$ is a polynomial of 9-th degree and $q(z)$ of 8-th degree.

Taking $a - b = 1$, it results

$$G(z) = \frac{z^9 - 4z^8 + 3z^7 + 4z^6 + 3z^5 - 18z^4 - 3z^3 + 16z^2 + 12z + 2}{z^8 - 6z^7 + 11z^6 - 15z^4 - 6z^3 + 17z^2 + 12z + 2}.$$

It has 6 fixed points, the roots of $z^2 - 1$, which are super-attracting, and another 4, which are repelling.

On the other hand, if we take $a = b$, the iteration function simplifies to

$$G(z) = \frac{2z^5 + 12z^3 + 2z}{8z^4 + 8z^2},$$

whose fixed points are ± 1, super-attracting, and $\pm\sqrt{-3}/3$, repelling.

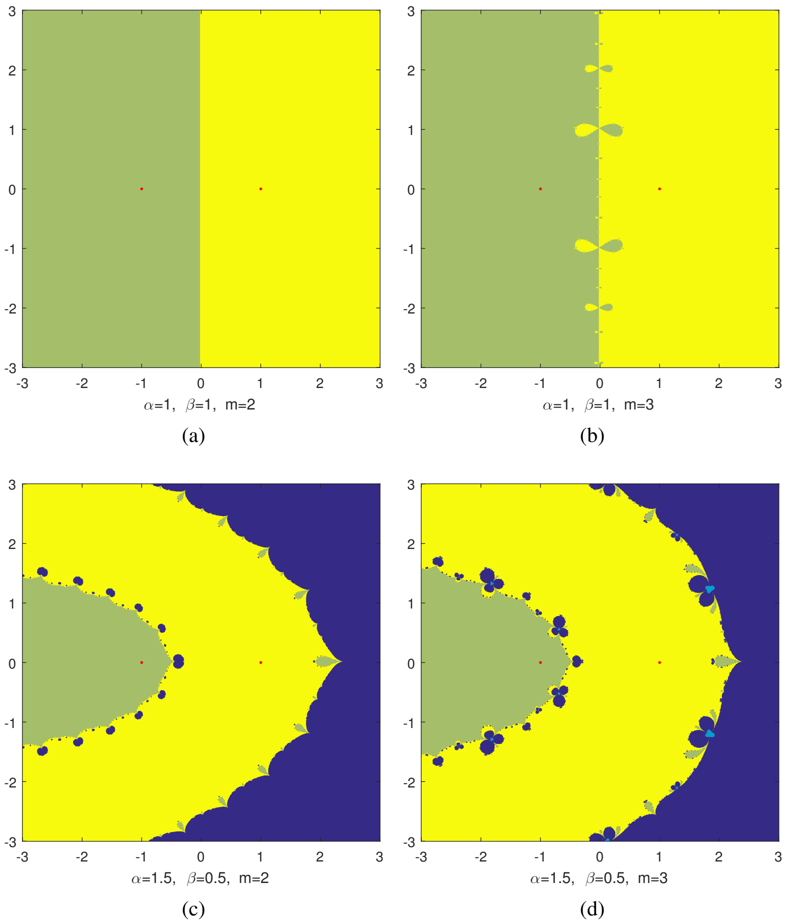


Figure 6. Dynamical planes for the multistep generalized Steffensen method with frozen second difference $S2_m$ for $z^2 - 1$ with different parameter values and number of steps. The red dots are the roots of the polynomial and their basins are coloured in green and yellow. The points whose orbits diverge to infinity are in deep blue and the light blue represents the points whose orbit does not fulfill the convergence criteria.

Taking $m = 3$ in $S2_m$, the iteration function $G(z)$ is the quotient of polynomials of degrees 21 and 20. In the case of $a - b = 1$, there are 18 repelling fixed points, besides the 2 super-attracting roots.

For $a = b$, $G(z)$ is the quotient of polynomials of degrees 11 and 10, and there are only 8 fixed points, including the 2 super-attracting roots. The remaining fixed points are repelling.

Figure 6 shows that the different instances of $S2_m$ behave better than their $S1_m$ counterparts that achieve the same convergence order.

A similar dynamical study has been performed using the third degree polynomial $f(z) = z^3$. The results are graphically presented in figures 7 and 8.

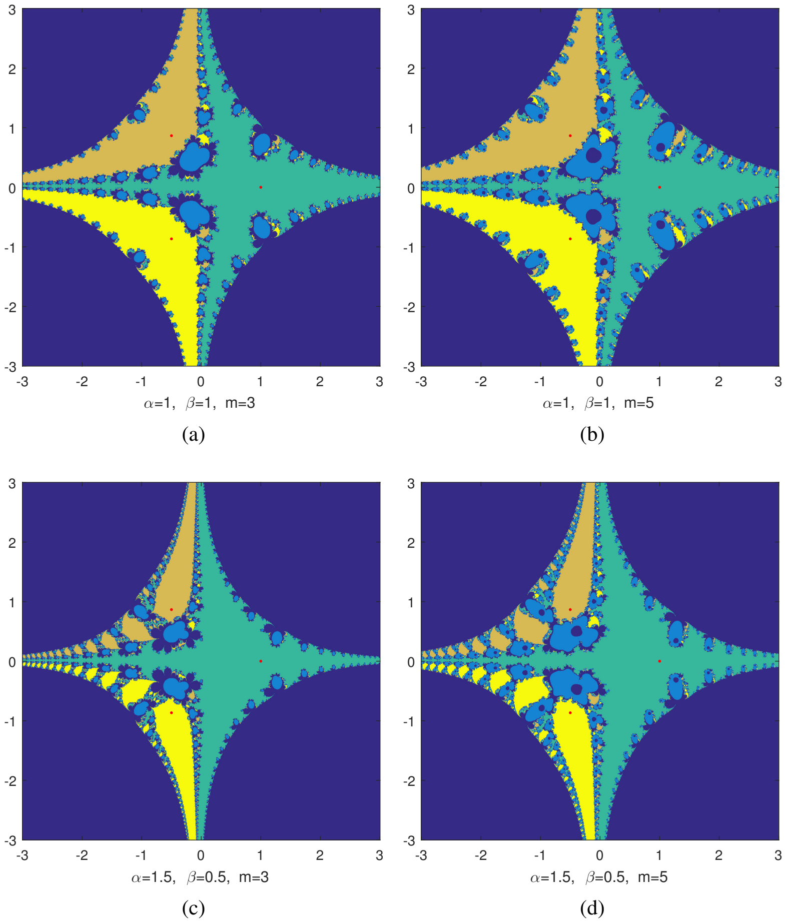
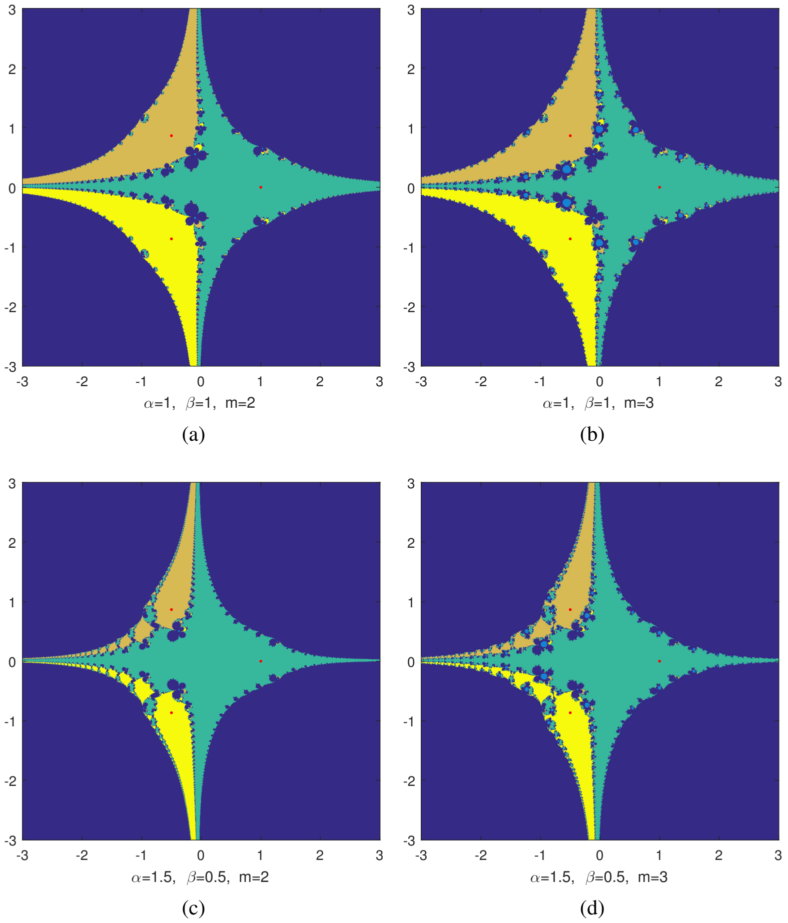


Figure 7. Dynamical planes for the multistep generalized Steffensen method $S1_m$ for $z^3 - 1$ with different parameter values and number of steps.



(a) (b)

(c) (d)

Figure 8. Dynamical planes for the multistep generalized Steffensen method with frozen second difference $S2_m$ for $z^2 - 1$ with different parameter values and number of steps.

7. Conclusions

In this paper we present a new family of derivative-free iterative methods with m steps per iteration. The constructed scheme not only allows us to generalize the family of iterative methods published in [15], but also increases the order of convergence from $m+1$ to $2m$. As we can see, the order of convergence has increased considerably, so that the difference between the two families is notable for large values of m. This fact has also been reflected in the study of efficiency, since the computational cost increment is minor, taking into account the great improvement in the order of convergence.

Due to our interest in further accelerating the convergence of some methods in the family, we have introduced memory in some appropriate cases. This study leads us to build a method with memory that converges with very high order for large m, but it takes too long to converge. Despite the execution time, we have observed that with this technique we improve the results obtained by $S2_m$.

Finally, we have carried out a dynamic study to show the good behavior of the methods introduced in the paper.

This article and its results have inspired new ideas to deepen the study of the family of iterative methods $S2_m$. The first idea deals about putting memory into $S1_m$ and $S2_m$ using Kurchatov's divided differences. On the other hand, in order to study this family from other points of view, we are carrying out a local convergence analysis of this method in order to theoretically determine which methods have better accessibility.

Funding

This research was partially supported by Ministerio de Economía y Competitividad under grant PGC2018-095896-B-C22.

References

[1] V. Alarcón, S. Amat, S. Busquier, and D. J. López, *Steffensen's type method in Banach spaces with applications on boundary-value problems*, J . Comput. Appl. Math. 216 (2008), pp. 243–250.

[2] S. Amat, S. Busquier, A. Grau, and M. Grau-Sánchez, *Maximum efficiency for a family of Newton-like methods with frozen derivatives and some applications*, Appl. Math. Comput. 219 (2013), pp. 7954–7963.

[3] S. Amat, S. Busquier, M. Grau-Sánchez, M.A. Hernández-Verón. *On the Efficiency of a Family of Steffensen-Like Methods with Frozen Divided Difference*, Computational Methods in Applied Mathematics, vol. 17, no. 2, 2017, pp. 187-199.

[4] S. Amat, J.A. Ezquerro, and M. A. Hernández, *On a Steffensen-like method for solving nonlinear equations*, Calcolo 53 (2016), pp. 171–188.

[5] I.K. Argyros, *A new convergence theorem for Steffensen's method on Banach spaces and applications*, Southwest J. Pure Appl. Math. 1 (1997), pp. 23–29.

[6] D.D. Bruns and J.E. Bailey, *Nonlinear feedback control for operating a nonisothermal CSTR near an unstable steady state.* Chem. Eng. Sci. 32 (1977), pp. 257–264.

[7] A. Cordero and J.R. Torregrosa, *Variants of Newton's method using fifth-order quadrature formulas*, Applied Mathematics and Computation 190 (2007), pp. 686–698.

[8] J.A. Ezquerro, M.A. Hernández, N. Romero, and A.I. Velasco, *On Steffensen's method on Banach spaces*, J. Comput. Appl. Math. 249 (2013), pp. 9–23.

[9] Hamamatsu, *Photomultiplier tubes. Basics and applications. 4th edition.* https://www.hamamatsu.com/resources/pdf/etd/PMT_handbook_v4E.pdf, retrieved in january 2022.

[10] C.L. Howk, J.L. Hueso, Eulalia Martínez, and C. Teruel, *A class of efficient high-order iterative methods*

with memory for nonlinear equations and their dynamics, Mathematical Methods in the Applied Sciences 41 (2018).

[11] F. Hueso-González, D. Ginestar, J.L. Hueso, and J. Riera, *Comments on "SPICE Model of Photomultiplier Tube Under Different Bias Conditions"*, IEEE Sens J 21, (2021), pp. 17395–17402.

[12] A. Cordero , J.L. Hueso, E. Martínez, and J.R. Torregrosa, A modified Newton-Jarratt's composition, Numerical Algorithms 55 (2010), pp. 87–99.

[13] H.T. Kung and J.F. Traub, *Optimal order of one-point and multi-point iteration*, Journal of the Association for Computing Machinery 21 (1974), pp. 643–651.

[14] M.A. Hernández-Verón, Eulalia Martínez, and C. Teruel, *Semilocal convergence of a k-step iterative process and its application for solving a special kind of conservative problems*, Numer Algor 76 (2017), pp. 309–331.

[15] M. Narang, S. Bhatia, A. Saleh Alshomrani, and V. Kanwar, *General efficient class of Steffensen type methods with memory for solving systems of nonlinear equations*, Journal of Computational and Applied Mathematics 352 (2019), pp. 23–39.

[16] J.M. Ortega, *The Newton-Kantorovich theorem*, Amer. Math. Monthly 75 (1968), pp. 658–660.

[17] J.M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[18] A.M. Ostrowski, *Solutions of Equations and System of Equations*. Academic Press, New York, 1960.

[19] J.F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

[20] A.M. Wazwaz, *Applications of Integral Equations*, Linear and Nonlinear Integral Equations, Springer: Berlin/Heidelberg, Germany, 2011.