Highly efficient iterative algorithms for solving nonlinear systems with arbitrary order of convergence $p+3$, $p \geq 5$ [☆]

Alicia Cordero[a,*], Cristina Jordán[a], Esther Sanabria-Codesal[b], Juan R. Torregrosa[a]

[a] *Instituto de Matemáticas Multidisciplinar, Universitat Politècnica de València, València, Spain*
[b] *Departamento de Matemática Aplicada, Universitat Politècnica de València, València, Spain*

Abstract

It is known that the concept of optimality is not defined for multidimensional iterative methods for solving nonlinear systems of equations. However, usually optimal fourth-order schemes (extended to the case of several variables) are employed as starting steps in order to design higher order methods for this kind of problems. In this paper, we use a non-optimal (in scalar case) iterative procedure that is specially efficient for solving nonlinear systems, as the initial steps of an eighth-order scheme that improves the computational efficiency indices of the existing methods, as far as the authors know. Moreover, the method can be modified by adding similar steps, increasing the order of convergence three times per step added.

This kind of procedures can be used for solving big-sized problems, such as those obtained by applying finite differences for approximating the solution of diffusion problem, heat conduction equations, etc. Numerical comparisons are made with the same existing methods, on standard nonlinear systems and Fisher's equation by transforming it in a nonlinear system by using finite differences. From these numerical examples, we confirm the theoretical results and show the performance of the proposed schemes.

Keywords: Nonlinear systems; iterative method; convergence; efficiency index; Fisher's equation.

1. Introduction

Nonlinearity is ubiquitous in physical phenomena as fluid and plasma mechanics, gas dynamics, elasticity, relativity, chemical reactions, combustion, ecology, biomechanics, economics modeling problems, transport theory and many other problems that are modeled by nonlinear equations. So, the design of fixed point iterative methods for solving systems of nonlinear equations is a challenging task in Numerical Analysis.

The proliferation of iterative methods for solving nonlinear equations has been spectacular in the last years (we can see a good overview in [2, 16]). Some of these methods can be transferred directly to the context of nonlinear systems, keeping the order of convergence, but others, at least apparently, cannot be extended to multidimensional case (although it can be done by using divided differences operator, as it was done in [1, 6, 8]). Other times, the procedures are designed specifically for multidimensional problems, as it is the case.

We will focus our efforts in finding the solution $\bar{x}$ of a nonlinear system $F(x) = 0$, wherein $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ is a sufficiently Fréchet differentiable function in an open convex set D. Although the most used method for finding the solution $\bar{x} \in D$ is Newton's scheme,

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}),\ k = 0, 1, 2, \dots,$$

where $F'(x^{(k)})$ is the Jacobian matrix of function F evaluated in the kth iteration, in recent years many iterative methods have been designed for solving multidimensional nonlinear problems as [5, 7, 11, 12, 13, 17] and the references therein.

In what follows we present some recently known methods of eighth-order of convergence that will be used in the comparison with our proposed scheme, in the computational efficiency and in the numerical tests. The first one is due to

[☆] This research was partially supported by Ministerio de Economía y Competitividad MTM2014-52016-C2-2-P, MTM2015-64013-P and Generalitat Valenciana PROMETEO/2016/089.

[*] Corresponding author
Email addresses: `acordero@mat.upv.es` (Alicia Cordero), `cjordan@mat.upv.es` (Cristina Jordán), `esanabri@mat.upv.es` (Esther Sanabria-Codesal), `jrtorre@mat.upv.es` (Juan R. Torregrosa)

Xiao and Yin, who in [19] described the following four-step eighth-order scheme, that we denote by XY8

$$\begin{aligned}
y^{(k)} &= x^{(k)} - \frac{2}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}),\\
z^{(k)} &= x^{(k)} - \left(-I + \frac{9}{4}[F'(y^{(k)})]^{-1}F'(x^{(k)}) + \frac{3}{4}[F'(x^{(k)})]^{-1}F'(y^{(k)})\right)[F'(x^{(k)})]^{-1}F(x^{(k)}),\\
w^{(k)} &= z^{(k)} + \left(3[F'(y^{(k)})]^{-1} - [F'(x^{(k)})]^{-1}\right)F(z^{(k)}),\\
x^{(k+1)} &= w^{(k)} + \left(3[F'(y^{(k)})]^{-1} - [F'(x^{(k)})]^{-1}\right)F(w^{(k)}).
\end{aligned} \tag{1}$$

Also, Xiao and Yin used a sixth-order method from Sharma and Arora's [17] to generate an eighth-order scheme, that is going to be used in this paper for comparison purposes. In the following, it is denoted by SA8.

$$\begin{aligned}
y^{(k)} &= x^{(k)} - \frac{2}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}),\\
z^{(k)} &= x^{(k)} - \left(\frac{23}{8}I + [F'(x^{(k)})]^{-1}F'(y^{(k)})\left(-3I + \frac{9}{8}[F'(x^{(k)})]^{-1}F'(y^{(k)})\right)\right)[F'(x^{(k)})]^{-1}F(x^{(k)}),\\
w^{(k)} &= z^{(k)} + \left(\frac{5}{2}I - \frac{3}{2}[F'(x^{(k)})]^{-1} - [F'(y^{(k)})]^{-1}\right)[F'(x^{(k)})]^{-1}F(z^{(k)}),\\
x^{(k+1)} &= w^{(k)} + \frac{1}{2}\left(3[F'(y^{(k)})]^{-1} - [F'(x^{(k)})]^{-1}\right)F(w^{(k)}).
\end{aligned} \tag{2}$$

The eighth-order scheme designed by Soleymani et al. [18], that we denote by SLB8, will be also used in this manuscript. It is a four-step Jarratt-type method whose iterative expression is

$$\begin{aligned}
y^{(k)} &= x^{(k)} - \frac{2}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}),\\
z^{(k)} &= x^{(k)} - \frac{1}{2}\left[3F'(y^{(k)}) - F'(x^{(k)})\right]^{-1}\left[3F'(y^{(k)}) + F'(x^{(k)})\right][F'(x^{(k)})]^{-1}F(x^{(k)}),\\
w^{(k)} &= z^{(k)} - \left(\frac{1}{2}\left[3F'(y^{(k)}) - F'(x^{(k)})\right]^{-1}\left[3F'(y^{(k)}) + F'(x^{(k)})\right]\right)^2[F'(x^{(k)})]^{-1}F(z^{(k)}),\\
x^{(k+1)} &= w^{(k)} - \left(\frac{1}{2}\left[3F'(y^{(k)}) - F'(x^{(k)})\right]^{-1}\left[3F'(y^{(k)}) + F'(x^{(k)})\right]\right)^2[F'(x^{(k)})]^{-1}F(w^{(k)}).
\end{aligned} \tag{3}$$

In order to compare the different methods, under the point of view of the computational cost, we use two different tools: Ostrowski in [15] defined (for the scalar case) the efficiency index as $I = p^{1/d}$, where p is the order of convergence and d the number of functional evaluations per iteration. Let us remark that, as we study the multidimensional case, for evaluating function F we need n scalar functional evaluations (the coordinate functions of F), whilst for evaluating Jacobian F' it is necessary to evaluate n^2 functions (all the entries of matrix F'). On the other hand, all the iterative methods for solving nonlinear systems require one or more matrix inversion, that is, one or more linear systems must be solved. So, the number of operations needed for solving a linear system plays in this context an important role. For this reason, the authors introduced in [7] the computational efficiency index, CI, which combine the efficiency index defined by Ostrowski and the number of products-quotients required per iteration. This index was defined as $CI = p^{1/(d+op)}$, where *op* is the number of products-quotients per iteration.

We recall that the number of products and quotients required for solving a linear system by Gaussian elimination is $\frac{1}{3}n^3 + n^2 - \frac{1}{3}n$, where n is the size of the system. In addition, for solving q linear systems with the same matrix of coefficients, by using LU decomposition we need $\frac{1}{3}n^3 + qn^2 - \frac{1}{3}n$ products-quotients. By using this information, in Section 3 we compare the computational efficiency indices of the different methods used in this manuscript.

The main objective of this paper is to develop high-order iterative methods in such a way that they involve as lower computational cost as possible, as well as they have good stability properties. From a highly-efficient method of order five, we design an eighth-order scheme increasing the order of convergence up to eight, by using a combination between Jacobian matrices and divided differences of first order. The order of convergence of the resulting procedure can be increased in three units by adding a new step with the same structure as the last one. This idea is generalized for obtaining an iterative method of arbitrary order increasing in three units the order each time that we add a new step with the same structure as the previous one. In each new step we only need a new functional evaluation.

In order to analyze the convergence properties of the different new schemes that will be introduced in this paper, we need to recall several concepts and tools, some of them introduced by the authors in [7].

1.1. Basic definitions

Let $\{x^{(k)}\}_{k\geq 0}$ be a sequence in $\mathbb{R}^n$ which converges to $\bar{x}$. Then, the convergence is called of order p, $p \geq 1$, if there exists $M > 0$ ($0 < M < 1$ if $p = 1$) and k_0 such that

$$\|x^{(k+1)} - \bar{x}\| \leq M\|x^{(k)} - \bar{x}\|^p, \ \ \forall k \geq k_0,$$

or

$$\|e^{(k+1)}\| \leq M\|e^{(k)}\|^p, \ \ \forall k \geq k_0,$$

where $e^{(k)} = x^{(k)} - \bar{x}$.

The following notation was introduced in [7], but we present it for completeness. Let $F : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be sufficiently Fréchet differentiable in D. The qth derivative of F at $u \in \mathbb{R}^n$, $q \geq 1$, is the q-linear function $F^{(q)}(u) : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$ such that $F^{(q)}(u)(v_1, \ldots, v_q) \in \mathbb{R}^n$. It is easy to observe that

1. $F^{(q)}(u)(v_1, \ldots, v_{q-1}, \cdot) \in \mathcal{L}(\mathbb{R}^n)$
2. $F^{(q)}(u)(v_{\sigma(1)}, \ldots, v_{\sigma(q)}) = F^{(q)}(u)(v_1, \ldots, v_q)$, for all permutation σ of $\{1, 2, \ldots, q\}$

From the above properties we can use the following notation:

(a) $F^{(q)}(u)(v_1, \ldots, v_q) = F^{(q)}(u)v_1 \ldots v_q$

(b) $F^{(q)}(u)v^{q-1}F^{(p)}v^p = F^{(q)}(u)F^{(p)}(u)v^{q+p-1}$

On the other hand, for $\bar{x} + h \in \mathbb{R}^n$ lying in a neighborhood of a solution $\bar{x}$ of $F(x) = 0$, we can apply Taylor's expansion and assuming that the Jacobian matrix $F'(\bar{x})$ is nonsingular, we have

$$F(\bar{x} + h) = F'(\bar{x}) \left[h + \sum_{q=2}^{p-1} C_q h^q \right] + \mathcal{O}(h^p), \tag{4}$$

where $C_q = (1/q!)[F'(\bar{x})]^{-1}F^{(q)}(\bar{x})$, $q \geq 2$. We observe that $C_q h^q \in \mathbb{R}^n$ since $F^{(q)}(\bar{x}) \in \mathcal{L}(\mathbb{R}^n \times \cdots \times \mathbb{R}^n, \mathbb{R}^n)$ and $[F'(\bar{x})]^{-1} \in \mathcal{L}(\mathbb{R}^n)$.

In addition, we can express F' as

$$F'(\bar{x} + h) = F'(\bar{x}) \left[I + \sum_{q=2}^{p-1} qC_q h^{q-1} \right] + \mathcal{O}(h^{p-1}), \tag{5}$$

where I is the identity matrix. Therefore, $qC_q h^{q-1} \in \mathcal{L}(\mathbb{R}^n)$. From (5), we obtain

$$[F'(\bar{x} + h)]^{-1} = \left[I + X_2 h + X_3 h^2 + X_4 h^4 + \cdots \right] [F'(\bar{x})]^{-1} + \mathcal{O}(h^{p-1}), \tag{6}$$

where

$$\begin{aligned} X_2 &= -2C_2, \\ X_3 &= 4C_2^2 - 3C_3, \\ X_4 &= -8C_2^3 + 6C_2C_3 + 6C_3C_2 - 4C_4, \\ &\vdots \end{aligned}$$

The equation

$$e^{(k+1)} = L{e^{(k)}}^p + \mathcal{O}({e^{(k)}}^{p+1}),$$

where L is a p-linear function $L \in \mathcal{L}(\mathbb{R}^n \times \cdots \times \mathbb{R}^n, \mathbb{R}^n)$, is called *error equation* and p is the *order of convergence*. Observe that ${e^{(k)}}^p$ is $(e^{(k)}, e^{(k)}, \cdots, e^{(k)})$.

On the other hand, the divided difference operator of F on $\mathbb{R}^n$ is a mapping $[\cdot, \cdot; F] : \Omega \times \Omega \subset \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathcal{L}(\mathbb{R}^n)$ (see [14]) such that

$$[x, y; F](x - y) = F(x) - F(y), \ \text{ for any } x, y \in \Omega.$$

In the proof of the following result, we will use Genochi-Hermite formula (see [14])

$$[x, y; F] = \int_0^1 F'(x + t(x - y))dt, \ \text{for all } (x, y) \in \Omega \times \Omega$$

and, by developing $F'(x+th)$ in Taylor series around x, we obtain

$$\int_0^1 F'(x+th)dt = F'(x) + \frac{1}{2}F''(x)h + \frac{1}{6}F'''(x)h^2 + \mathcal{O}(h^3).$$

Denoting by $e = x - \bar{x}$ and assuming that $F'(\bar{x})$ is nonsingular, we have:

$$\begin{aligned} F(x) &= F'(\bar{x})(e + C_2e^2 + C_3e^3 + C_4e^4 + C_5e^5) + \mathcal{O}(e^6), \\ F'(x) &= F'(\bar{x})(I + 2C_2e + 3C_3e^2 + 4C_4e^3 + 5C_5e^4) + \mathcal{O}(e^5), \\ F''(x) &= F'(\bar{x})(2C_2 + 6C_3e + 12C_4e^2 + 20C_5e^3) + \mathcal{O}(e^4), \\ F'''(x) &= F'(\bar{x})(6C_3 + 24C_4e + 60C_5) + \mathcal{O}(e^2), \end{aligned}$$

where $C_q = \dfrac{1}{q!}[F'(\bar{x})]^{-1}F^{(q)}(\bar{x})$, $q \geq 2$. Replacing these developments in the formula of Genocchi-Hermite and denoting the second point of the divided difference by $y = x + h$ and the error at the first step by $e_y = y - \bar{x}$, we have

$$[x, y; F] = F'(\bar{x})[I + C_2(e_y + e) + C_3e^2] + \mathcal{O}(e^3).$$

In particular, if y is an approximation of the solution provided by the Newton's method, i.e. $h = x - y = [F'(x)]^{-1}F(x)$, we obtain

$$[x, y; F] = F'(\bar{x})[I + C_2e + (C_2^2 + C_3)e^2] + \mathcal{O}(e^3).$$

We summarize the contents of this paper. In Section 2, we describe the new eighth-order iterative method for solving nonlinear systems and show a procedure for constructing schemes with arbitrary order of convergence $p + 3$ from a previous algorithms with order of convergence $p \geq 5$, by using the same structure. The efficiency and computational efficiency indices of the new and other known schemes are analyzed in Section 3. Some numerical tests on Fisher's equation and also on some academical problems are made in Section 4, in order to confirm the theoretical results and to show the performance of the presented schemes. The paper finishes with some conclusions and the references used.

2. Development and convergence of the method

By adding a new step to the iterative method described in [4], we construct the following four-step scheme with eight-order of convergence:

$$\begin{aligned} y^{(k)} &= x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} &= y^{(k)} - 5\,[F'(x^{(k)})]^{-1}F(y^{(k)}), \\ w^{(k)} &= z^{(k)} - \frac{1}{5}\,[F'(x^{(k)})]^{-1}(-16\,F(y^{(k)}) + F(z^{(k)})), \\ x^{(k+1)} &= w^{(k)} - G(t^{(k)})[F'(x^{(k)})]^{-1}F(w^{(k)}), \end{aligned} \tag{7}$$

where G is a weight function that should be chosen in order to obtain the eighth-order of convergence, being

$$t^{(k)} = I - 5F'(x^{(k)})]^{-1}[y^{(k)}, z^{(k)}; F].$$

The following result establishes the convergence of iterative method (7).

Theorem 1. *Let $F : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be sufficiently differentiable at each point of an open neighborhood D of $\bar{x} \in \mathbb{R}^n$, that is a solution of the system $F(x) = 0$ and the initial estimation $x^{(0)}$ is close enough to $\bar{x}$. Let us suppose that $F'(x)$ is continuous and nonsingular in $\bar{x}$. Then, sequence $\{x^{(k)}\}_{k\geq 0}$ obtained from expression (7) converges to $\bar{x}$ with order 8 when $G(0) = \frac{49}{25}$, $G'(0) = \frac{7}{25}$ and $G''(0) = \frac{1}{50}$, being in this case the error equation*

$$\begin{aligned} e^{(k+1)} = \Big[&210C_2^7 - 30C_2^4C_3C_2 + 90C_2^3C_3C_2^2 - 21C_2C_3C_2^4 + 3C_2C_3C_2C_3C_2 - 9C_2C_3^2C_2^2 + 63\,C_3C_2^5 \\ &- 9C_3C_2^2C_3C_2 + 27C_3C_2C_3C_2^2)\Big]{e^{(k)}}^8 + \mathcal{O}({e^{(k)}}^9), \end{aligned}$$

where $C_q = \dfrac{1}{q!}[F'(\bar{x})]^{-1}F^{(q)}(\bar{x})$, $q = 2, 3, \ldots$

Proof. By using Taylor expansion of $F(x^{(k)})$ and $F'(x^{(k)})$ around $\bar{x}$,

$$F(x^{(k)}) = F'(\bar{x})\Big[e^{(k)} + C_2{e^{(k)}}^2 + C_3{e^{(k)}}^3 + C_4{e^{(k)}}^4 + C_5{e^{(k)}}^5\Big] + \mathcal{O}({e^{(k)}}^6),$$

$$F'(x^{(k)}) = F'(\bar{x})\Big[I + 2C_2e^{(k)} + 3C_3{e^{(k)}}^2 + 4C_4{e^{(k)}}^3 + 5C_5{e^{(k)}}^4\Big] + \mathcal{O}({e^{(k)}}^5).$$

From the above expression, we have

$$[F'(x^{(k)})]^{-1} = \Big[I + X_2e^{(k)} + X_3{e^{(k)}}^2 + X_4{e^{(k)}}^3 + X_5{e^{(k)}}^4\Big][F'(\overline{x})]^{-1} + \mathcal{O}({e^{(k)}}^5),$$

where

$$\begin{aligned}
X_2 &= -2C_2,\\
X_3 &= -3C_3 + 4C_2^2,\\
X_4 &= -4C_4 + 6C_2C_3 + 6C_3C_2 - 8C_2^3,\\
X_5 &= -5C_5 + 8C_2C_4 - 12C_2^2C_3 + 9C_3^2 + 8C_4C_2 - 12C_2C_3C_2 + 16C_2^4 - 12C_3C_2^2.
\end{aligned}$$

Then,

$$\begin{aligned}
\Big[F'(x^{(k)})\Big]^{-1} F(x^{(k)}) &= e^{(k)} - C_2{e^{(k)}}^2 + 2(C_2^2 - C_3){e^{(k)}}^3 + (4C_2C_3 + 3C_3C_2 - 4C_2^3 - 3C_4){e^{(k)}}^4 +\\
&\quad (-4C_5 + 6C_2C_4 - 8C_2^2C_3 + 6C_3^2 + 4C_4C_2 - 6C_2C_3C_2 + 8C_2^4 - 6C_3C_2^2){e^{(k)}}^5 + \mathcal{O}({e^{(k)}}^6).
\end{aligned}$$

So,

$$\begin{aligned}
y^{(k)} - \bar{x} &= C_2{e^{(k)}}^2 - 2(C_2^2 - C_3){e^{(k)}}^3 - (4C_2C_3 + 3C_3C_2 - 4C_2^3 - 3C_4){e^{(k)}}^4\\
&\quad - (-4C_5 + 6C_2C_4 - 8C_2^2C_3 + 6C_3^2 + 4C_4C_2 - 6C_2C_3C_2 + 8C_2^4 - 6C_3C_2^2){e^{(k)}}^5 + \mathcal{O}({e^{(k)}}^6),\\
(y^{(k)} - \bar{x})^2 &= C_2^2{e^{(k)}}^4 + (2C_2C_3 + 2C_3C_2 - 4C_2^3){e^{(k)}}^5 + \mathcal{O}({e^{(k)}}^6),
\end{aligned}$$

and

$$\begin{aligned}
F(y^{(k)}) &= F'(\bar{x})\Big[(y^{(k)} - \bar{x}) + C_2(y^{(k)} - \bar{x})^2\Big] + \mathcal{O}((y^{(k)} - \bar{x})^3)\\
&= F'(\bar{x})\Big[C_2{e^{(k)}}^2 + 2(C_3 - C_2^2){e^{(k)}}^3 + (3C_4 + 5C_2^3 - 3C_3C_2 - 4C_2C_3){e^{(k)}}^4\\
&\quad + (4C_5 - 6C_2C_4 + 10C_2^2C_3 - 6C_3^2 - 4C_4C_2 + 8C_2C_3C_2 - 12C_2^4 + 6C_3C_2^2){e^{(k)}}^5\Big] + \mathcal{O}({e^{(k)}}^6).
\end{aligned}$$

In order to obtain the Taylor expansion of

$$z^{(k)} = y^{(k)} - 5\,[F'(x^{(k)})]^{-1}F(y^{(k)}),$$

we calculate

$$\begin{aligned}
\Big[F'(x^{(k)})\Big]^{-1} F(y^{(k)}) &= -C_2{e^{(k)}}^2 + (-4C_2^2 + 2C_3){e^{(k)}}^3 + (-8C_2C_3 - 6C_3C_2 + 13C_2^3 + 3C_4){e^{(k)}}^4\\
&\quad + (4C_5 - 12C_2C_4 - 8C_4C_2 + 26C_2^2C_3 + 18C_3C_2^2 - 12C_3^2 + 20C_2C_3C_2 - 38C_2^4){e^{(k)}}^5 + \mathcal{O}({e^{(k)}}^6).
\end{aligned}$$

In a similar way as before, we get

$$\begin{aligned}
z^{(k)} - \bar{x} &= -4C_2{e^{(k)}}^2 + (18C_2^2 - 8C_3){e^{(k)}}^3 + (36C_2C_3 + 27C_3C_2 - 61C_2^3 - 12C_4){e^{(k)}}^4\\
&\quad - (-16C_5 + 54C_2C_4 + 36C_4C_2 - 122C_2^2C_3 - 84C_3C_2^2 - 94C_2C_3C_2 + 54C_3^2 + 182C_2^4){e^{(k)}}^5 + \mathcal{O}({e^{(k)}}^6),\\
(z^{(k)} - \bar{x})^2 &= 16C_2^2{e^{(k)}}^4 + (32C_2C_3 + 32C_3C_2 - 144C_2^3){e^{(k)}}^5 + \mathcal{O}({e^{(k)}}^6),
\end{aligned}$$

and

$$\begin{aligned}
F(z^{(k)}) &= F'(\bar{x})\Big[(z^{(k)} - \bar{x}) + C_2(z^{(k)} - \bar{x})^2\Big] + \mathcal{O}((z^{(k)} - \bar{x})^3)\\
&= F'(\bar{x})\Big[-4C_2{e^{(k)}}^2 + (18C_2^2 - 8C_3){e^{(k)}}^3 + (36C_2C_3 + 27C_3C_2 - 45C_2^3 - 12C_4){e^{(k)}}^4\\
&\quad + (-16C_5 + 54C_2C_4 + 36C_4C_2 - 90C_2^2C_3 - 84C_3C_2^2 - 62C_2C_3C_2 + 54C_3^2 + 38C_2^4){e^{(k)}}^5\Big] + \mathcal{O}({e^{(k)}}^6).
\end{aligned}$$

Taking into account the definition of $w^{(k)}$,

$$\begin{aligned} w^{(k)} - \bar{x} &= z^{(k)} - \bar{x} - \frac{1}{5}\,[F'(x^{(k)})]^{-1}(-16\,F(y^{(k)}) + F(z^{(k)})) \\ &= y^{(k)} - \bar{x} - \frac{1}{5}\,[F'(x^{(k)})]^{-1}(9\,F(y^{(k)}) + F(z^{(k)})), \end{aligned}$$

from

$$\begin{aligned} \Big[F'(x^{(k)})\Big]^{-1}(9\,F(y^{(k)}) + F(z^{(k)})) &= 5C_2{e^{(k)}}^2 + (-10C_2^2 + 10C_3){e^{(k)}}^3 \\ &+ (-20C_2C_3 - 15C_3C_2 + 20C_2^3 + 15C_4){e^{(k)}}^4 \\ &+ (20C_5 - 30C_2C_4 - 20C_4C_2 + 40C_2^2C_3 + 40C_2C_3C_2 - 110C_2^4 - 30C_3^2){e^{(k)}}^5 \\ &+ \mathcal{O}({e^{(k)}}^6), \end{aligned}$$

we get

$$w^{(k)} - \bar{x} = (14C_2^4 + 6C_3C_2^2 - 2C_2C_3C_2){e^{(k)}}^5 + \mathcal{O}({e^{(k)}}^6),$$

and

$$\begin{aligned} F(w^{(k)}) &= F'(\bar{x})(w^{(k)} - \bar{x}) + \mathcal{O}((w^{(k)} - \bar{x})^8) \\ &= F'(\bar{x})\Big[(14C_2^4 + 6C_3C_2^2 - 2C_2C_3C_2){e^{(k)}}^5\Big] + \mathcal{O}({e^{(k)}}^6). \end{aligned}$$

In order to obtain the weight function G, we calculate

$$\Big[y^{(k)}, z^{(k)}; F\Big] = F'(y^{(k)}) + F''(y^{(k)})(z^{(k)} - y^{(k)}) + \mathcal{O}((z^{(k)} - y^{(k)})^2),$$

where

$$F(y^{(k)}) = F'(\bar{x})(y^{(k)} - \bar{x}) + \frac{F''(\bar{x})}{2}(y^{(k)} - \bar{x})^2 + \mathcal{O}((y^{(k)} - \bar{x})^3),$$

$$F'(y^{(k)}) = F'(\bar{x})\Big[I + 2C_2(y^{(k)} - \bar{x}) + 3C_3(y^{(k)} - \bar{x})^2\Big] + \mathcal{O}((y^{(k)} - \bar{x})^3),$$

$$F''(y^{(k)}) = F'(\bar{x})\Big[2C_2 + 6C_3(y^{(k)} - \bar{x}) + 12C_4(y^{(k)} - \bar{x})^2\Big] + \mathcal{O}((y^{(k)} - \bar{x})^3).$$

Then,

$$\Big[y^{(k)}, z^{(k)}; F\Big] = F'(\bar{x})\Big[I - 3C_2^2{e^{(k)}}^2 + (16C_2^3 - 6C_2C_3){e^{(k)}}^3\Big] + \mathcal{O}({e^{(k)}}^4)$$

and expanding

$$t_k = I - 5\,\Big[F'(x^{(k)})\Big]^{-1}\Big[y^{(k)}, z^{(k)}; F\Big] = -4I + 10C_2e^{(k)} + (15C_3 - 5C_2^2){e^{(k)}}^2 + (20C_4 - 70C_2^3 + -30C_3C_2){e^{(k)}}^3 + \mathcal{O}({e^{(k)}}^4),$$

we get

$$\begin{aligned} x^{(k+1)} - \bar{x} &= (w^{(k)} - \bar{x}) - \Big(G(0)I + G'(0)t_k + \frac{1}{2}\,G''(0)t_k^2\Big)\Big[F'(x^{(k)})\Big]^{-1}F(w^{(k)}) \\ &= (w^{(k)} - \bar{x}) - \Big(G(0)I + G'(0)t_k + \frac{1}{2}\,G''(0)t_k^2\Big)\Big[F'(x^{(k)})\Big]^{-1}F'(\bar{x})(w^{(k)} - \bar{x}) \\ &= \Big[\Big((1 - G(0) + 4G'(0) - 8G''(0)\Big)I + \Big(2G(0) - 18G'(0) + 56G''(0)\Big)C_2e^{(k)} \\ &+ \Big((-4G(0) + 41G'(0) - 182G''(0))C_2^2 + (3G(0) - 27G'(0) + 84G''(0))C_3\Big){e^{(k)}}^2 \\ &+ \Big((8G(0) - 12G'(0) + 134G''(0))C_2^3 + (-6G(0) + 54G'(0) - 243G''(0))C_2C_3 \\ &+ (-6G(0) + 84G'(0) - 363G''(0))C_3C_2 + (4G(0) - 36G'(0) + 112G''(0))C_4\Big){e^{(k)}}^3 + \mathcal{O}({e^{(k)}}^4)\Big](w^{(k)} - \bar{x}). \end{aligned}$$

And then, by solving the homogeneous system of equations obtained from the coefficients of $I, e^{(k)}, {e^{(k)}}^2$ we get $G(0)=\frac{49}{25}$, $G'(0)=\frac{7}{25}$ and $G''(0)=\frac{1}{50}$. Since the order of $(w^{(k)}-\bar{x})$ is $p\geq 5$, we have the desired order with error equation

$$e^{(k+1)}=\Big[210C_2^7-30C_2^4C_3C_2+90C_2^3C_3C_2^2-21C_2C_3C_2^4+3C_2C_3C_2C_3C_2-9C_2C_3^2C_2^2+63\,C_3C_2^5 \\ -9C_3C_2^2C_3C_2+27C_3C_2C_3C_2^2)\Big]{e^{(k)}}^8+\mathcal{O}({e^{(k)}}^9).$$

■

In the previous results, some conditions on the weight function have been imposed to assure the desired order of convergence. This gives us the possibility of designing different schemes depending on the $G(t^{(k)})$ used, as for example

$$G(t^{(k)})=\frac{49}{25}I+\frac{7}{25}t^{(k)}+\frac{1}{100}{t^{(k)}}^2,$$

or

$$G(t^{(k)})=\Big[25(28I-t^{(k)})\Big]^{-1}\Big(49(28I+3t^{(k)})\Big).$$

In the second case, a new inverse appears and it yields to an additional linear system to be solved in practice. So, we use the first one in the rest of the manuscript, denoting the resulting scheme by M8.

In a similar way, this structure can be extended in order to construct an iterative scheme of arbitrary order.

Theorem 2. *Let $u^{(k)}=\phi(x^{(k)})$ be the iterative expression of a method of order $p\geq 8$, with asymptotic error constant M, where the four first steps are those of (7). By adding a new steps in the form*

$$x^{(k+1)}=u^{(k)}-G(t^{(k)})\,[F'(x^{(k)})]^{-1}F(u^{(k)}),$$

the order of the new method is $p+3$ if $G(0)=\frac{49}{25}$, $G'(0)=\frac{7}{25}$ and $G''(0)=\frac{1}{50}$.

Proof. By hypothesis,

$$F(u^{(k)})=M{e^{(k)}}^p+\mathcal{O}({e^{(k)}}^{p+1}).$$

Then, taking into account the conditions on function G, following a similar procedure as in the previous theorem,

$$\begin{aligned}x^{(k+1)}-\bar{x}&=(u^{(k)}-\bar{x})-\Big(G(0)I+G'(0)t_k+\frac{1}{2}\,G''(0)t_k^2\Big)\Big[F'(x^{(k)})\Big]^{-1}F(u^{(k)})\\ &=\Big[(8G(0)-12G'(0)+134G''(0))C_2^3+(-6G(0)+54G'(0)-243G''(0))C_2C_3\\ &\quad+(-6G(0)+84G'(0)-363G''(0))C_3C_2+(4G(0)-36G'(0)+112G''(0))C_4\Big){e^{(k)}}^3+\mathcal{O}({e^{(k)}}^4)\Big]M{e^{(k)}}^p\\ &=\Big[15C_2^3+-\frac{3}{2}C_2C_3+\frac{9}{2}C_3C_2\Big){e^{(k)}}^3+\mathcal{O}({e^{(k)}}^4)\Big]M{e^{(k)}}^p\end{aligned}$$

and then, the order is $p+3$. ■

3. Computational efficiency

Now, we are going to use two indices for comparing the different iterative schemes for solving nonlinear systems: the multidimensional extension of the efficiency index defined by Ostrowski as $I=p^{1/d}$ and the computational efficiency index CI defined as $CI=p^{1/(d+op)}$, where p is the order of convergence, d is the number of functional evaluations per iteration and op is the number of products-quotients per iteration.

In Table 1, the efficiency indices I of methods M8, XY8, SA8 and SLB8 are presented. The number of Jacobian evaluations and the number of functional evaluations in all these schemes are different, but the order of convergence is the same. In order to calculate the efficiency index I, it must be taken into account that the number of functional evaluations of one F, F' and first order divided difference $[\cdot,\cdot;F]$ at certain iterates are n, n^2 and $n(n-1)$, respectively. We can observe that the classical efficiency index is the same for all these methods.

On the other hand, in order to compute an inverse linear operator we solve a $n\times n$ linear system where we have to do $\frac{1}{3}n^3+n^2-\frac{1}{3}n$ products-quotients for obtaining LU decomposition and solving two triangular linear systems. In addition, we need n^2 products for matrix-vector multiplication.

Method	n. F	n. F'	n. $[\cdot,\cdot;F]$	Funct. evaluations	I
M8	4	1	1	$2n^2+3n$	$8^{\frac{1}{2n^2+3n}}$
XY8	3	2	0	$2n^2+3n$	$8^{\frac{1}{2n^2+3n}}$
SA8	3	2	0	$2n^2+3n$	$8^{\frac{1}{2n^2+3n}}$
SLB8	3	2	0	$2n^2+3n$	$8^{\frac{1}{2n^2+3n}}$

Table 1: Efficiency index for different schemes

Taking into account the previous considerations, we calculate CI of method M8. For each iteration, we need to evaluate function F four times, once Jacobian F' and once the divided difference, so $2n^2+3n$ functional evaluations are needed. In addition, we must solve six linear systems with $F'(x^{(k)})$ as coefficients matrix (that is $\frac{1}{3}n^3+6n^2-\frac{1}{3}n$ products-quotients) and two matrix-vector products ($2n^2$ products-quotients). Therefore, the value of index CI for method M8 on a nonlinear system of size $n\times n$ is

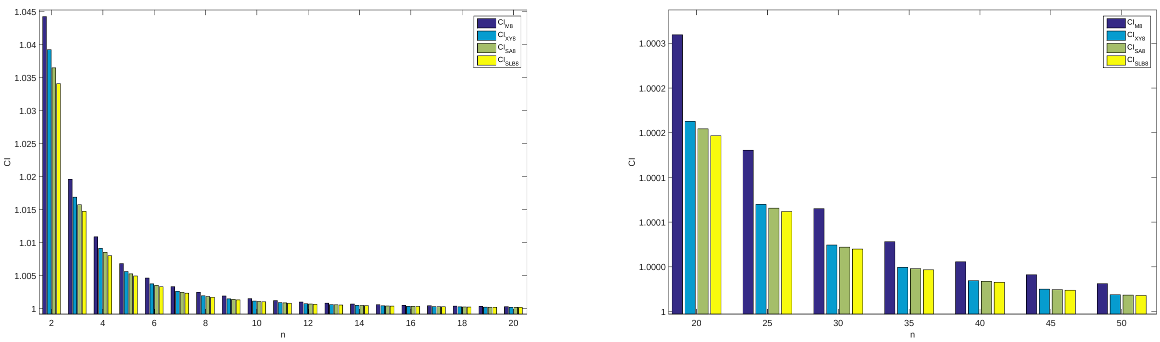
$$CI_{M8}=8^{\frac{1}{\frac{1}{3}n^3+10n^2+\frac{8}{3}n}}.$$

In Table 2 we show index CI of schemes M8, XY8, SA8 and SLB8. In it, NFE is the number of functional evaluations, $NLS1$ denotes the number of linear systems with the matrix of coefficients $F'(x^{(k)})$ to be solved, $NLS2$ is the number of linear systems with other matrix of coefficient that are solved and $M\times V$ denotes the number of matrix-vector products.

Method	NFE	$NLS1$	$NLS2$	$M\times V$	CI
M8	$2n^2+3n$	6	0	2	$8^{\frac{1}{\frac{1}{3}n^3+10n^2+\frac{8}{3}n}}$
XY8	$2n^2+3n$	4	3	2	$8^{\frac{1}{\frac{2}{3}n^3+11n^2+\frac{7}{3}n}}$
SA8	$2n^2+2n$	6	1	3	$8^{\frac{1}{\frac{2}{3}n^3+12n^2+\frac{7}{3}n}}$
SLB8	$2n^2+3n$	3	3	5	$8^{\frac{1}{\frac{2}{3}n^3+13n^2+\frac{7}{3}n}}$

Table 2: Functional evaluations and products-quotients of the methods

Let us observe that, although the classical index is the same in all these cases, it is not the case of the computational efficiency index since the number of inverse linear operators is different for each scheme. In Figure 1, the computational efficiency index for those methods and systems of size from 2 to 50 is shown. We can observe that, for any size of the system, the best index corresponds to proposed method M8, due to the factor of the dominating term, that is $\frac{1}{3}n^3$ in M8, in comparison to $\frac{2}{3}n^3$ in the rest of schemes.



(a) Size from 2 to 20

(b) Size from 20 to 50

Figure 1: Indices CI for different sizes of the system

4. Numerical results

In this section, we firstly check the applicability of our proposed method by analyzing its performance on Fisher's partial differential equation. Afterwards, we apply this and other known methods of the same order in order to compare their behavior on some academic problems.

Example 1. *Fisher's equation,*

$$u_t = Du_{xx} + ru(1 - \frac{u}{k}), \;\; x \in [a, b] \;\; t \geq 0,$$

was initially proposed by Fisher in [10] as a model of the diffusion process in population dynamics, being $D > 0$ the diffusion constant, r the growth rate of the species and k the carrying capacity. More recently, this equation has shown to be useful for a great amount of problems as wave propagation, economy or genetics.

In this example, we analyze a particular case of Fisher's equation, corresponding to $r = k = 1$, null boundary conditions and $u(x, 0) = sech^2(7x)$ (see for example [3]). By applying an implicit finite differences scheme, we transform Fisher's problem in a family of nonlinear systems to be solved. They provide the estimated solution for each instant t_j from the approximation calculated at t_{j-1}.

We choose the spacial step $h = 8/nx$ and the timing step $k = T_{max}/nt$, being nx and nt the amount of subintervals for variables x and t, respectively and T_{max} is the final instant of the numerical study. So, we get a mesh whose domain is $[-4, 4] \times [0, T_{max}]$ with $(nx + 1) \times (nt + 1)$ equi-spaced nodes (x_i, t_j).

By using the estimations of the derivatives

$$u_t(x, t) \approx \frac{u(x, t) - u(x, t - k)}{k} \text{ and } u_{xx}(x, t) \approx \frac{u(x + h, t) - 2u(x, t) + u(x - h, t)}{h^2}$$

and denoting by $u_{i,j}$ the estimation of the solution at (x_i, t_j), we construct the nonlinear system

$$ku_{i+1,j} + (kh^2 - 2k - h^2)u_{i,j} - kh^2u_{i,j}^2 + ku_{i-1,j} = -h^2u_{i,j-1},$$

for $i = 1, 2, \ldots, nx - 1$ and $j = 1, 2, \ldots, nt$.

Therefore, for a fixed and arbitrary index j, we use the proposed eighth-order scheme M8 to solve a nonlinear system of size $(nx - 1) \times (nx - 1)$, getting the estimation of the solution at instant t_j, $u_{i,j}$, by using meanwhile the calculated estimation at the previous instant $u_{i,j-1}$. This is made for different values of the final time T_{max}, employing different values of $nx = 20$ and nt. In this iterative process, double precision arithmetics is used and the tolerance of the error is 10^{-8}. After solving each nonlinear system, we consider as initial guess for the solution of the next system, the solution at t_{j-1}. In Table 3 we show the mean of the number of iterations employed (iter) for solving the nonlinear systems for each instant, by using several T_{max} and also the estimation of the error (norm of the solution of the last system) $\|F(x^{(k+1)})\|$ and the cpu-time for calculations, in seconds. The processor of the machine used is Intel(R) Xeon(R) CPU E5-2420 v2 @ 2.20GHz, with 64 GB of RAM.

T_{max}	nx	nt	iter	$\|F(x^{(k+1)})\|$	cpu-time
0.6	20	10	1	1.24e-13	0.027
0.6	200	10	1	3.74e-15	0.164
1	20	10	1	6.45e-12	0.027
1	200	10	1	6.50e-14	0.162
6	20	10	2	1.76e-16	0.033
6	200	10	2	2.29e-16	0.286
20	20	20	nc	-	-
20	20	80	1.0625	5.02e-17	0.064
20	200	10	1.0250	1.01e-16	0.781

Table 3: Numerical results for Fisher's equation

It can be observed in Figure 2 that at the first instants diffusion is greater than reaction, and the curve decreases. Then, reaction increases and the system reaches the equilibrium state. In Figure 3, the long-term behavior can be observed for $T_{max} = 20$ seconds and a large enough number of temporal subintervals $nt = 80$.

In the rest of the section, we compare the new method M8 with Newton's procedure and eighth-order schemes XY8, SA8 and SLB8. The numerical results are shown in Tables 4 to 7. All experiments have been carried out on Matlab $R2014b$

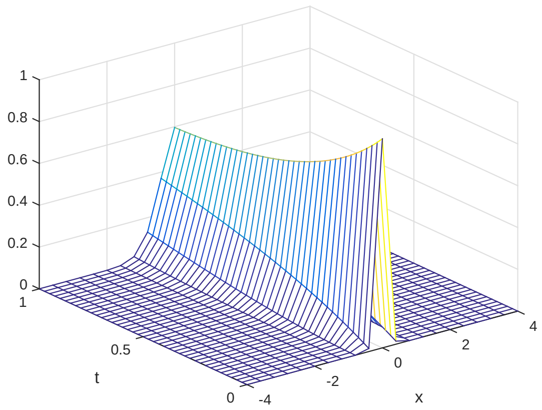


Figure 2: Approximated solution for $nx = 20$, $nt = 40$ and $T_{max} = 1$

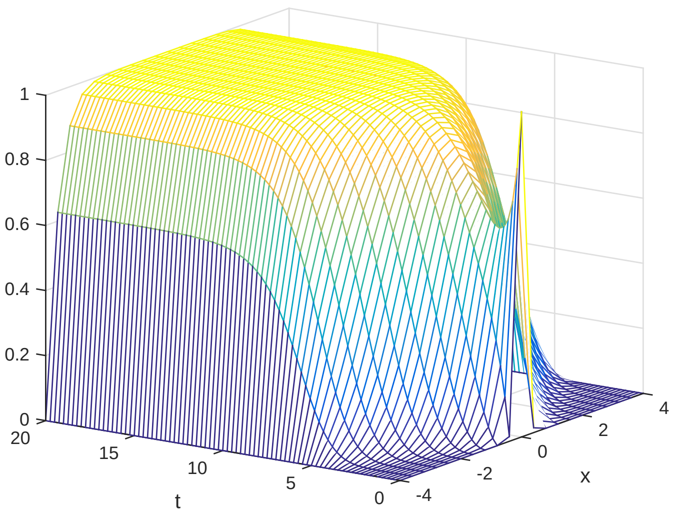


Figure 3: Approximated solution for $nx = 20$, $nt = 80$ and $T_{max} = 20$

using variable precision arithmetics with 4000 digits of mantissa and the stopping criterium is $\|x^{(k+1)} - x^{(k)}\| < 10^{-500}$ or $\|F(x^{(k+1)})\| < 10^{-500}$. To check the theoretical order of convergence p, we calculate the approximated computational order of convergence (ACOC) introduced in [9] as

$$p \approx ACOC = \frac{\ln\left(\|x^{(k+1)} - x^{(k)}\| / \|x^{(k)} - x^{(k-1)}\|\right)}{\ln\left(\|x^{(k)} - x^{(k-1)}\| / \|x^{(k-1)} - x^{(k-2)}\|\right)}.$$

Let us remark that, if the entries of vector ACOC do not stabilize their values along the iterative process, it is marked as '-'.

Example 2. *The first standard nonlinear system is described as:*

$$x_i^2 x_{i+1} - 1 = 0, \;\; i = 1, 2, \ldots, n-1,$$
$$x_n^2 x_1 - 1 = 0.$$

whose solution is $\bar{x} = (1, 1, \ldots, 1)^T$.

We use $n = 9$ and two initial estimations, $x^{(0)} = (1.25, 1.25, \ldots, 1.25)^T$ and $x^{(0)} = (-1, -1, \ldots, -1)^T$. The obtained results can be observed in Tables 4 and 5, where the number of iterations needed, the estimations of the error in the last iteration $\|x^{(k+1)} - x^{(k)}\|$ and $\|F(x^{(k+1)})\|$ and the approximated computational order of convergence ACOC, are shown.

	Newton	M8	XY8	SA8	SLB8
iter	10	4	4	4	4
$\|x^{(k+1)} - x^{(k)}\|$	1.99e-344	2.97e-212	2.33e-270	1.00e-238	9.85e-361
$\|F(x^{(k+1)})\|$	3.96e-688	2.04e-1693	2.17e-2159	1.13e-1905	3.60e-2884
ACOC	2.0000	7.9999	8.0000	8.0000	8.0000

Table 4: Numerical results for Example 2, $x^{(0)} = (1.25, 1.25, \ldots, 1.25)^T$

	Newton	M8	XY8	SA8	SLB8
iter	14	6	5	4	nc
$\|x^{(k+1)} - x^{(k)}\|$	4.02e-280	2.66e-231	1.65e-120	3.74e-63	-
$\|F(x^{(k+1)})\|$	1.62e-559	8.50e-1846	1.36e-960	4.18e-501	-
ACOC	2.0000	8.0000	7.9976	8.0386	-

Table 5: Numerical results for Example 2, $x^{(0)} = (-1, -1, \ldots, -1)^T$

The obtained results show good convergence properties, even using bad initial estimations, as well as an excellent precision in the calculated approximation.

Example 3. *The third nonlinear system is defined by:*

$$x_i - \cos\left(2x_i - \sum_{j=1}^{4} x_j\right) = 0, \quad i = 1, 2, \ldots, n.$$

We use $n = 20$ and initial estimations $x^{(0)} = (1, 1, \ldots, 1)^T$ and $x^{(0)} = (-0.1, -0.1, \ldots, -0.1)^T$ for our numerical tests, being the solution in this case $\bar{x} \approx (0.5149, 0.5149, \ldots, 0.5149)^T$. In Tables 6 and 7 we show again the number of iterations needed (iter), the estimations of the error in the last iteration $\|x^{(k+1)} - x^{(k)}\|$ and $\|F(x^{(k+1)})\|$ satisfying the same stopping criterium as before and the approximated computational order of convergence (ACOC).

	Newton	M8	XY8	SA8	SLB8
iter	9	4	4	4	4
$\|x^{(k+1)} - x^{(k)}\|$	1.93e-277	3.38e-346	9.22e-310	2.80e-305	3.44e-316
$\|F(x^{(k+1)})\|$	8.60e-555	2.09e-2770	8.75e-2478	1.37e-2441	9.43e-2530
ACOC	2.0000	8.0000	8.0000	8.0000	-

Table 6: Numerical results for Example 3, $x^{(0)} = (1, 1, \ldots, 1)^T$

	Newton	M8	XY8	SA8	SLB8
iter	56	4	nc	nc	nc
$\|x^{(k+1)} - x^{(k)}\|$	2.21e-328	3.12e-70	-	-	-
$\|F(x^{(k+1)})\|$	1.13e-656	1.09e-562	-	-	-
ACOC	2.0000	7.7892	-	-	-

Table 7: Numerical results for Example 3, $x^{(0)} = (-0.1, -0.1, \ldots, -0.1)^T$

It is observed in Tables 6 and 7 that, for this problem, the proposed method M8 shows to be more stable than the rest of schemes of the same order of convergence, as it converges even for initial estimations far from the solution, as Newton's procedure does, but in a number of iterations very reduced.

5. Conclusions

In this manuscript we have constructed a highly-efficient iterative method of order eight for solving nonlinear systems. Indeed, higher order methods can be constructed by adding new steps with the same structure involving one new functional evaluation (per step) and increasing the order of convergence in three units per step. Its applicability and stability have been checked by means of an applied problem, Fisher's equation, showing excellent results and also have been compared with recent high order methods on academical examples with competitive results.

6. References

[1] M. Abad, A. Cordero, J.R. Torregrosa, A family of seventh-order schemes for solving nonlinear systems. Bull. Math. Soc. Sci. Math. Roumanie 57(105) (2014) 133–145.

[2] S. Amat, S. Busquier, Advances in Iterative Methods for Nonlinear Equations, SEMA SIMAI Springer Series 10, Springer International Publishing, Switzerland 2016.

[3] K. Al-Khaled, K. Numerical study of Fisher's reaction-diffusion equation by Sinc collocation method, Comput. Appl. Math. 137 (2001) 245–255.

[4] V. Arroyo, A. Cordero, J.R. Torregrosa, Approximation of artificial satellites' preliminary orbits: The efficiency challeng, Math. Comput. Model. 54 (2011) 1802–1807.

[5] F. Awawdeh, On new iterative method for solving systems of nonlinear equations, Numer. Algor. 54 (2010) 395–409.

[6] D. Budzko, A. Cordero, J.R. Torregrosa, A new family of iterative methods widening areas of convergence, Appl. Math. Comput. 252 (2015) 405–417.

[7] A. Cordero, J.L. Hueso, E. Martínez, J.R. Torregrosa, A modified Newton-Jarratt's composition, Numer. Algor. 55 (2010) 87–99.

[8] A. Cordero, J.G. Maimó, J.R. Torregrosa, M.P. Vassileva, Solving nonlinear problems by Ostrowski–Chun type parametric families, Math. Chem. 53 (2015) 430–449.

[9] A. Cordero, J.R. Torregrosa, Variants of Newton's method using fifth-order quadrature formulas, Appl. Math. Comput. 190 (2007) 686–698.

[10] R.A. Fisher, The wave of advance of advantageous genes, Ann. Eugenics 7 (1937) 353-369.

[11] C. Grosan, A. Abraham, A new approach for solving nonlinear equations systems, IEEE Trans. Syst. Man Cybernet Part A: Syst. Humans 38 (2008) 698–714.

[12] J.L. Hueso, E. Martínez, C. Teruel, Convergence, effiency and dinamimics of new fourth and sixth order families of iterative methods for nonlinear systems, Comput. Appl. Math. 275 (2015) 412–420.

[13] H. Montazeri, F. Soleymani, S. Shateyi, S.S. Motsa, On a new method for computing the numerical solution of systems of nonlinear equations, Appl. Math. vol. 2012 (2012) ID. 751975, 15 pages.

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

[15] A.M. Ostrowski, Solution of equations and systems of equations, Prentice-Hall, Englewood Cliffs, New York, 1964.

[16] M. Petković, B. Neta, L. Petković, J. Džunić, Multipoint methods for solving nonlinear equations, Academic Press, Amsterdam, 2013.

[17] J.R. Sharma, H. Arora, Efficient Jarratt-like methods for solving systems of nonlinear equations, Calcolo 51 (2014) 193–210.

[18] F. Soleymani, T. Lofti, P. Bakhtiari, A multistep class of iterative methods for nonlinear systems, Optim. Letters 8 (2014) 1001–1015.

[19] X.Y. Xiao, H.W. Yin, Increasing the order of convergence for iterative methods to solve nonlinear systems, Calcolo 53(3) (2016) 285–300.