



# Efficient parametric family of fourth-order Jacobian-free iterative vectorial schemes

Alicia Cordero<sup>1</sup> · Renso V. Rojas-Hiciano<sup>2,3</sup> · Juan R. Torregrosa<sup>1</sup> · Paula Triguero-Navarro<sup>4</sup>

Received: 13 November 2023 / Accepted: 31 January 2024  
© The Author(s) 2024

## Abstract

In this work, a multiparametric family of iterative vectorial fourth-order methods free of Jacobian matrices is proposed. A convergence analysis of this family is carried out as well as a study of its efficiency. Several numerical experiments are made in order to compare the behaviour of the proposed family with other competitive methods of the literature.

**Keywords** Iterative scheme · Non-linear systems · Jacobian-free · Efficiency index

## 1 Introduction

Newton's method, see [1], is one of the best known methods and has the following structure for vectorial non-linear problems:

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

where  $F : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a non-linear vectorial function describing the non-linear system  $F(x) = 0$ .

This method has quadratic convergence, and it is computationally very efficient. Many methods have been designed in an attempt to improve Newton's convergence and its computational efficiency, see for instance [2].

On the other hand, Newton's method is the first optimal method for vectorial problems according to the following conjecture for vectorial methods proposed in [3].

**Conjecture 1** The order of convergence of any iterative method, without memory, for solving non-linear systems cannot exceed  $2^{k_1+k_2-2}$ , where  $k_1$  is the number of

---

✉ Alicia Cordero  
acordero@mat.upv.es

Extended author information available on the last page of the article

evaluations of the Jacobian matrix and  $k_2$  the number of evaluations of the non-linear function per iteration, and  $k_1 \leq k_2$ . When the scheme reaches this upper bound, we say that it is optimal.

In the literature, up to our knowledge, only one other optimal method for vectorial problems has been proposed according to the above conjecture and it is the family proposed in [4]. In it, the authors proposed a parametric family of optimal methods of order 4 which has the following structure:

$$\begin{aligned} y^{(k)} &= x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}), \quad k = 0, 1, \dots, \\ x^{(k+1)} &= y^{(k)} - [F'(x^{(k)})]^{-1} (p_k F(y^{(k)}) + q_k F(x^{(k)})), \end{aligned} \tag{2}$$

where  $\lambda$  and  $\psi$  are free real parameters,

$$v_k = \frac{\|F(y^{(k)})\|_2^2}{\|F(x^{(k)})\|_2^2}, \quad K_k = \frac{1}{1 + \lambda v_k}, \quad p_k = K_k(1 + \psi v_k), \quad q_k = 2K_k v_k.$$

The idea of using such a definition of the variable  $v_k$  comes from the article [5].

The iterative class (2) has a Jacobian matrix in its iterative expression, so it is not a Jacobian-free family. The aim of this work is to modify this family to obtain a new class of Jacobian-free schemes, which maintains the convergence order. In addition, we intend to make the modifications without considerably increasing the computational cost in order to obtain an efficient and competitive family compared to methods with the same characteristics.

In this article, a Jacobian-free multiparametric class of iterative methods for solving non-linear systems is presented. In Sect. 2, the parametric family is presented and an analysis of the convergence order is performed. Section 3 presents an analysis of the efficiency of the proposed family and compares it with other known methods in the literature with similar characteristics. In Sect. 4, several numerical experiments are carried out to see the behaviour of the family and to compare it with other iterative methods. The article concludes with some conclusions and references used in it.

## 2 Design of the parametric family

The only part we need to replace to obtain a derivative-free method is the Jacobian matrix  $F'(x^{(k)})$  of the family (2). For that reason, we are going to keep the previous structure by replacing the Jacobian matrix with a divided difference operator. If we replace the matrix by a forward or backward divided difference operator, the order of convergence is not maintained, so we choose the following symmetrical divided difference operator  $[x^{(k)} + rF(x^{(k)}), x^{(k)} - rF(x^{(k)}); F]$ , which has a parameter  $r$  that can be any real number different from 0.

The parametric family obtained is denoted by *CRTT* and has the following iterative structure

$$\begin{aligned}
 y^{(k)} &= x^{(k)} - [x^{(k)} + rF(x^{(k)}), x^{(k)} - rF(x^{(k)}); F]^{-1} F(x^{(k)}), \\
 x^{(k+1)} &= y^{(k)} - [x^{(k)} + rF(x^{(k)}), x^{(k)} - rF(x^{(k)}); F]^{-1} (p_k F(y^{(k)}) + q_k F(x^{(k)})), \tag{3}
 \end{aligned}$$

where  $r \in \mathbb{R}, r \neq 0$  and

$$\begin{aligned}
 v_k &= \frac{F(y^{(k)})^T F(y^{(k)})}{F(x^{(k)})^T F(x^{(k)})}, & K_k &= \frac{1}{1 + \lambda v_k}, \\
 p_k &= K_k(1 + \psi v_k), & q_k &= 2K_k v_k.
 \end{aligned}$$

We prove that the parametric family *CRTT* has fourth order of convergence for any value of real parameters  $r, \lambda$  and  $\psi$ .

**Theorem 2** *Let us consider  $F : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$  a differentiable enough function defined in a neighbourhood  $D$  of  $\alpha$ , such that  $F(\alpha) = 0$ . Let us also assume that  $F'(\alpha)$  is non-singular. Therefore, being  $x^{(0)}$  an initial guess close enough to  $\alpha$ , sequence  $\{x^{(k)}\}$  defined by *CRTT* converges to  $\alpha$  with order 4, for any non-zero value of parameter  $r$  and for any values of  $\lambda$  and  $\psi$ .*

**Proof** We first obtain the Taylor development of  $F(x^{(k)}), F'(x^{(k)}), F''(x^{(k)})$  and  $F'''(x^{(k)})$  around  $\alpha$ , where  $e_k = x^{(k)} - \alpha$  and  $C_i = \frac{1}{i!}[F'(\alpha)]^{-1} F^{(i)}(\alpha), i = 2, 3, \dots$

$$\begin{aligned}
 F(x^{(k)}) &= F'(\alpha)(e_k + C_2 e_k^2 + C_3 e_k^3) + O(e_k^4), \\
 F'(x^{(k)}) &= F'(\alpha)(I + 2C_2 e_k + 3C_3 e_k^2) + O(e_k^3), \\
 F''(x^{(k)}) &= F'(\alpha)(2C_2 + 6C_3 e_k) + O(e_k^2), \\
 F'''(x^{(k)}) &= F'(\alpha)6C_3 + O(e_k).
 \end{aligned}$$

From [6], it is obtained the following expansion using the Genocchi-Hermite formula:

$$\begin{aligned}
 [x^{(k)} + rF(x^{(k)}), x^{(k)} - rF(x^{(k)}); F] &= F'(x^{(k)}) + \frac{1}{6} F'''(x^{(k)})(rF(x^{(k)}))^2 + O(rF(x^{(k)}))^3 \\
 &= F'(\alpha)(I + 2C_2 e_k + 3C_3 e_k^2) + \frac{1}{6} F'(\alpha)(6C_3)r^2 F'(\alpha)^2 (e_k^2) + \\
 &\quad + O(e_k^3) \\
 &= F'(\alpha) \left( I + 2C_2 e_k + C_3 \left( 3I + r^2 F'(\alpha)^2 \right) e_k^2 \right) + O(e_k^3).
 \end{aligned}$$

It can be simply checked that the inverse operator of  $[x^{(k)} + rF(x^{(k)}), x^{(k)} - rF(x^{(k)}); F]$  has the following expression

$$[x^{(k)} + rF(x^{(k)}), x^{(k)} - rF(x^{(k)}); F]^{-1} = \left( I - 2C_2e_k + \left( 4C_2^2 - C_3 \left( 3I + r^2F'(\alpha)^2 \right) \right) e_k^2 \right) F'(\alpha)^{-1} + O\left(e_k^3\right). \tag{4}$$

From expression (4) and the Taylor development of  $F(x^{(k)})$ , we obtain that  $e_{y,k} = y^{(k)} - \alpha$  satisfies the following equality

$$\begin{aligned} e_{y,k} &= x^{(k)} - \alpha - [x^{(k)} + rF(x^{(k)}), x^{(k)} - rF(x^{(k)}); F]^{-1} F(x^{(k)}) \\ &= e_k - \left( I - 2C_2e_k + \left( 4C_2^2 - C_3 \left( 3I + r^2F'(\alpha)^2 \right) \right) e_k^2 \right) \left( e_k + C_2e_k^2 + C_3e_k^3 \right) + O\left(e_k^4\right) \\ &= e_k - e_k - C_2e_k^2 - C_3e_k^3 + 2C_2e_k^2 + 2C_2^2e_k^3 - \left( 4C_2^2 - C_3 \left( 3I + r^2F'(\alpha)^2 \right) \right) e_k^3 + O\left(e_k^4\right) \\ &= C_2e_k^2 + \left( -2C_2^2 + C_3 \left( 2I + r^2F'(\alpha)^2 \right) \right) e_k^3 + O\left(e_k^4\right). \end{aligned}$$

To simplify the notation, we define  $Y_3$  as follows

$$Y_3 = -2C_2^2 + C_3 \left( 2I + r^2F'(\alpha)^2 \right).$$

For any  $x \in \mathbb{R}^n$ , we have that  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where  $f_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  for  $i = 1, 2, \dots, n$ .

Also, it is verified that

$$\frac{F(y^{(k)})^T F(y^{(k)})}{F(x^{(k)})^T F(x^{(k)})} = \frac{\sum_{i=1}^n f_i^2(y^{(k)})}{\sum_{i=1}^n f_i^2(x^{(k)})}, \tag{5}$$

where the developments of  $f_i(x^{(k)})$  and  $f_i(y^{(k)})$  around  $\alpha$  for  $i = 1, 2, \dots, n$  are

$$\begin{aligned} f_i(x^{(k)}) &= f'_i(\alpha)e_k + \frac{1}{2}f''_i(\alpha)e_k^2 + O\left(e_k^3\right), \\ f_i(y^{(k)}) &= f'_i(\alpha)e_{y,k} + \frac{1}{2}f''_i(\alpha)e_{y,k}^2 + O\left(e_{y,k}^3\right), \end{aligned}$$

being  $f'_i(x) = \left( \frac{\partial f_i}{\partial x_1}, \frac{\partial f_i}{\partial x_2}, \dots, \frac{\partial f_i}{\partial x_n} \right)$  and  $f''_i(x)$  is the Hessian matrix with components  $\frac{\partial^2 f_i}{\partial x_j \partial x_k}$ , for  $j, k \in \{1, 2, \dots, n\}$ .

To simplify the notation, we can rewrite it in the following way

$$\begin{aligned}
 f_i(x^{(k)}) &= R_i e_k + H_i e_k^2 + O(e_k^3), \\
 f_i(y^{(k)}) &= R_i e_{y,k} + H_i e_{y,k}^2 + O(e_{y,k}^3),
 \end{aligned}$$

with  $R_i = f'_i(\alpha)$  and  $H_i = \frac{1}{2} f''_i(\alpha)$ .

In order to obtain  $v_k$ , we compute now  $f_i^2(x^{(k)})$  and  $f_i^2(y^{(k)})$ , for  $i = 1, 2, \dots, n$ :

$$\begin{aligned}
 f_i^2(x^{(k)}) &= f_i(x^{(k)})^T f_i(x^{(k)}) \\
 &= R_i^T R_i e_k^2 + (R_i^T H_i + H_i^T R_i) e_k^3 + O(e_k^4), \\
 f_i^2(y^{(k)}) &= f_i(y^{(k)})^T f_i(y^{(k)}) \\
 &= R_i^T R_i e_{y,k}^2 + (R_i^T H_i + H_i^T R_i) e_{y,k}^3 + O(e_{y,k}^4).
 \end{aligned} \tag{6}$$

By denoting  $P_i = R_i^T R_i$  and  $Q_i = R_i^T H_i + H_i^T R_i$ , the relations (6) can be rewritten as

$$\begin{aligned}
 f_i^2(x^{(k)}) &= P_i e_k^2 + Q_i e_k^3 + O(e_k^4), \\
 f_i^2(y^{(k)}) &= P_i e_{y,k}^2 + Q_i e_{y,k}^3 + O(e_{y,k}^4).
 \end{aligned} \tag{7}$$

Since  $e_{y,k} = C_2 e_k^2 + Y_3 e_k^3 + O(e_k^4)$ , therefore  $e_{y,k}^2 = C_2^2 e_k^4 + (C_2 Y_3 + Y_3 C_2) e_k^5 + O(e_k^6)$ . Substituting this relation in (7), we obtain

$$\begin{aligned}
 f_i^2(y^{(k)}) &= P_i e_{y,k}^2 + Q_i e_{y,k}^3 + O(e_{y,k}^4) \\
 &= P_i (C_2^2 e_k^4 + (C_2 Y_3 + Y_3 C_2) e_k^5) + O(e_k^6).
 \end{aligned} \tag{8}$$

If we denote  $P = \sum_{i=1}^n P_i$  and  $Q = \sum_{i=1}^n Q_i$ , and substituting the results of (7) and (8) in (5), we obtain

$$\begin{aligned}
 v_k &= \frac{P e_{y,k}^2 + Q e_{y,k}^3 + O(e_{y,k}^4)}{P e_k^2 + Q e_k^3 + O(e_k^4)} \\
 &= \frac{P(C_2^2 e_k^4 + (C_2 Y_3 + Y_3 C_2) e_k^5) + O(e_k^6)}{P e_k^2 + Q e_k^3 + O(e_k^4)} \\
 &= \frac{P(C_2^2 e_k^2 + (C_2 Y_3 + Y_3 C_2) e_k^3) + O(e_k^4)}{P + Q e_k + O(e_k^2)}.
 \end{aligned} \tag{9}$$

Since

$$[P + Qe_k + O(e_k^2)]^{-1} = P^{-1} - P^{-1}Qe_k + O(e_k^2),$$

we obtain that

$$\begin{aligned} v_k &= (P^{-1} - P^{-1}Qe_k)P(C_2^2e_k^2 + (C_2Y_3 + Y_3C_2)e_k^3) + O(e_k^4) \\ &= C_2^2e_k^2 + (C_2Y_3 + Y_3C_2)e_k^3 - P^{-1}QP C_2^2e_k^3 + O(e_k^4) \\ &= C_2^2e_k^2 + (C_2Y_3 + Y_3C_2 - P^{-1}QP C_2^2)e_k^3 + O(e_k^4). \end{aligned} \tag{10}$$

In order to simplify the notation, we denote  $V_3$  as follows

$$V_3 = (C_2Y_3 + Y_3C_2 - P^{-1}QP C_2^2),$$

obtaining then that  $v_k = C_2^2e_k^2 + V_3e_k^3 + O(e_k^4)$ .

Therefore, the expression of  $K_k$  is as follows

$$K_k = I \left( \frac{1}{1 + \lambda v_k} \right) = I - \lambda C_2^2e_k^2 + \lambda(C_2^4 - V_3)e_k^3 + O(e_k^4). \tag{11}$$

From (10) and (11), we obtain that the expressions of  $p_k$  and  $q_k$  are

$$\begin{aligned} p_k &= K_k(1 + \psi v_k) = (I - \lambda C_2^2e_k^2 + \lambda(C_2^4 - V_3)e_k^3)(1 + \psi v_k) + O(e_k^4) \\ &= I + (\psi - \lambda)C_2^2e_k^2 + ((-\lambda + \psi)V_3 + \lambda C_2^4)e_k^3 + O(e_k^4), \\ q_k &= 2K_k v_k = 2(I - \lambda C_2^2e_k^2 + \lambda(C_2^4 - V_3)e_k^3)v_k + O(e_k^4) \\ &= 2C_2^2e_k^2 + 2V_3e_k^3 + O(e_k^4). \end{aligned} \tag{12}$$

Therefore,

$$\begin{aligned} p_k F(y^{(k)}) + q_k F(x^{(k)}) &= C_2e_k^2 + C_3(2I + r^2F'(\alpha)^2)e_k^3 + \\ &\quad (-C_4 + 2V_3 + 2C_2C_3 + 3C_3C_2 + (-1 - \lambda + \psi)C_2^3 + r^2C_3F'(\alpha)^2C_2)e_k^4 \\ &\quad + O(e_k^5). \end{aligned} \tag{13}$$

Therefore, from (4) and (13), the error equation is

$$\begin{aligned} e_{k+1} &= y^{(k)} - \alpha - [x^{(k)} + rF(x^{(k)}), x^{(k)} - rF(x^{(k)}); F]^{-1} (p_k F(y^{(k)}) + q_k F(x^{(k)})) \\ &= (-2V_3 + 4C_2C_3 + 3C_3C_2 + (-7 + \lambda - \psi)C_2^3 + 2r^2C_2C_3F'(\alpha)^2 + r^2C_3F'(\alpha)^2C_2) e_k^4 \\ &\quad + O(e_k^5). \end{aligned}$$

Since

$$\begin{aligned}
 V_3 &= (C_2 Y_3 + Y_3 C_2 - P^{-1} Q P C_2^2) \\
 &= C_2(-2C_2^2 + C_3(2I + r^2 F'(\alpha)^2)) + (-2C_2^2 + C_3(2I + r^2 F'(\alpha)^2))C_2 - P^{-1} Q P C_2^2 \\
 &= -2C_2^3 + C_2 C_3(2I + r^2 F'(\alpha)^2) - 2C_2^3 + C_3(2I + r^2 F'(\alpha)^2)C_2 - P^{-1} Q P C_2^2 \\
 &= -4C_2^3 + C_2 C_3(2I + r^2 F'(\alpha)^2) + C_3(2I + r^2 F'(\alpha)^2)C_2 - P^{-1} Q P C_2^2,
 \end{aligned}$$

the error equation can be rewritten as

$$\begin{aligned}
 e_{k+1} &= y^{(k)} - \alpha - \left[ x^{(k)} + rF(x^{(k)}), x^{(k)} - rF(x^{(k)}); F \right]^{-1} (p_k F(y^{(k)}) + q_k F(x^{(k)})) \\
 &= (8C_2^3 - 2C_2 C_3(2I + r^2 F'(\alpha)^2) - 2C_3(2I + r^2 F'(\alpha)^2)C_2 + 2P^{-1} Q P C_2^2 + \\
 &\quad + 4C_2 C_3 + 3C_3 C_2 + (-7 + \lambda - \psi)C_2^3 + 2r^2 C_2 C_3 F'(\alpha)^2 + r^2 C_3 F'(\alpha)^2 C_2) e_k^4 \\
 &\quad + O(e_k^5) \\
 &= (2P^{-1} Q P C_2^2 + (1 + \lambda - \psi)C_2^3 - C_3(I + r^2 F'(\alpha)^2)C_2) e_k^4 + O(e_k^5).
 \end{aligned}$$

Therefore, it is proven that the members of the parametric family (3) have order of convergence 4. □

### 3 Efficiency index

To compare different iterative methods, the efficiency index suggested by Ostrowski is widely used. The formula of this index is the following

$$I_E = p^{1/d},$$

where  $p$  is the order of convergence of the method and  $d$  represents the number of functional evaluations needed to perform the method per iteration.

Another classical measure of the efficiency of iterative methods is the operational efficiency index proposed by Traub, with the expression:

$$I_O = p^{1/op},$$

where  $op$  is the number of operations, expressed in units of product, needed to calculate each iteration.

In several occasions, a combination of both is also used, called the computational efficiency index, whose expression is

$$I_{EC} = p^{1/(d+op)}.$$

In this section, we will study these different indices for our parametric family, as well as compare the results obtained with the efficiency indices of other methods known in the literature, which are *S2S* from [7], *CJTS5* from [8], *WF6S* from [9] and *WZ7S* from [10].

When working on systems of size  $n \times n$ , the number of functional evaluations used in calculating  $F(z)$  is  $n$  while the number used in calculating a divided difference operator as used in the iterative method is  $n^2 - n$ .

In the case of the *CRTT* parametric family, we compute  $F(x^{(k)})$ ,  $F(y^{(k)})$ ,  $F(x^{(k)} + rF(x^{(k)}))$  and  $F(x^{(k)} - rF(x^{(k)}))$  and just one divided difference operator, so the number of functional evaluations is:

$$4n + n^2 - n = n^2 + 3n.$$

Below is a list of the number of products and quotients needed to perform the operations involved:

- Each scalar/vector product costs  $n$ .
- Each transpose vector/vector product costs  $n$ .
- Each matrix/vector product costs  $n^2$ .
- Each matrix/matrix product costs  $n^3$ .
- The number of quotients of a divided difference operator is  $n^2$ .
- Each LU decomposition costs  $\frac{1}{3}(n^3 - n)$ .
- Each system resolution costs  $n^2$ .

In this case, we calculate three scalar/vector products, two transpose vector/vector products, a divided difference operator, and a single *LU* decomposition and we solve two systems, so the number of operations is:

$$3n + 2n + n^2 + \frac{1}{3}(n^3 - n) + 2n^2 = \frac{1}{3}(n^3 - n) + 3n^2 + 5n.$$

Then, the addition of evaluations and operations is:

$$\frac{1}{3}(n^3 - n) + 3n^2 + 5n + n^2 + 3n = \frac{1}{3}(n^3 - n) + 4n^2 + 8n.$$

In the following, we present and compute the efficiency of the methods with which we compare the indices.

The *S2S* method has the following iterative expression:

$$y^{(k)} = x^{(k)} - \left[ [x^{(k)} + F(x^{(k)}), x^{(k)} - F(x^{(k)}); F] \right]^{-1} F(x^{(k)}),$$



There are 3 evaluations and a divided difference, so the number of functional evaluations is:

$$3n + n^2 - n = n^2 + 2n.$$

On the other hand, a divided difference operator is calculated and a single system is solved, so the number of operations is:

$$n^2 + \frac{1}{3}(n^3 - n) + n^2 = \frac{1}{3}(n^3 - n) + 2n^2.$$

Thus, the total number of functional evaluations and operations is:

$$n^2 + 2n + \frac{1}{3}(n^3 - n) + 2n^2 = \frac{1}{3}(n^3 - n) + 3n^2 + 2n.$$

The iterative expression of *CJTS5* is:

$$\begin{aligned} y^{(k)} &= x^{(k)} - [a^{(k)}, b^{(k)}; F]^{-1} F(x^{(k)}), \\ z^{(k)} &= y^{(k)} - \alpha[a^{(k)}, b^{(k)}; F]^{-1} F(y^{(k)}), \\ t^{(k)} &= z^{(k)} - \beta[a^{(k)}, b^{(k)}; F]^{-1} F(z^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \gamma[a^{(k)}, b^{(k)}; F]^{-1} F(t^{(k)}), \end{aligned}$$

where  $a^{(k)} = x^{(k)} + F(x^{(k)})$  and  $b^{(k)} = x^{(k)} - F(x^{(k)})$ .

This method calculates 6 functional evaluations and a divided difference, then the number of evaluations is:

$$6n + n^2 - n = n^2 + 5n.$$

On the other hand, it performs 3 scalar/vector products, a divided difference operator, and a single LU decomposition and solves 4 systems with the same matrix; therefore, the number of operations is:

$$3n + n^2 + \frac{1}{3}(n^3 - n) + 4n^2 = \frac{1}{3}(n^3 - n) + 5n^2 + 3n.$$

The total number of evaluations and operations is:

$$n^2 + 5n + \frac{1}{3}(n^3 - n) + 5n^2 + 3n = \frac{1}{3}(n^3 - n) + 6n^2 + 8n.$$

The iterative expression of the *WF6S* method is as follows:

$$\begin{aligned} y^{(k)} &= x^{(k)} - [w^{(k)}, s^{(k)}; F]^{-1} F(x^{(k)}), \\ \mu_1 &= (3I - 2[w^{(k)}, s^{(k)}; F]^{-1}[y^{(k)}, x^{(k)}; F])[w^{(k)}, s^{(k)}; F]^{-1} \\ z^{(k)} &= y^{(k)} - \mu_1 F(y^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \mu_1 F(z^{(k)}), \end{aligned}$$

where  $w^{(k)} = x^{(k)} + F(x^{(k)})$  and  $s^{(k)} = x^{(k)} - F(x^{(k)})$ .

This method calculates 5 functional evaluations and 2 divided difference operators. Then, the number of functional evaluations is:

$$5n + 2n^2 - 2n = 2n^2 + 3n.$$

The number of operations performed is as follows: two divided difference operators are calculated, 3 systems are solved with the same coefficient matrix, a multiplication between 2 matrices is done, a scalar matrix multiplication is done and a vector matrix multiplication is done, so the number of operations is:

$$2n^2 + \frac{1}{3}(n^3 - n) + 3n^2 + n^2 + n^2 + n^3 = \frac{1}{3}(n^3 - n) + n^3 + 7n^2.$$

The total of evaluations and operations is:

$$2n^2 + 3n + \frac{1}{3}(n^3 - n) + n^3 + 7n^2 = \frac{1}{3}(n^3 - n) + n^3 + 9n^2 + 3n.$$

Finally, the iterative expression of the method *WZ7S* is:

$$\begin{aligned} y^{(k)} &= x^{(k)} - [w^{(k)}, x^{(k)}; F]^{-1} F(x^{(k)}), \\ z^{(k)} &= y^{(k)} - \left( [y^{(k)}, x^{(k)}; F] + [y^{(k)}, w^{(k)}; F] - [w^{(k)}, x^{(k)}; F] \right)^{-1} F(y^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \left( [z^{(k)}, x^{(k)}; F] + [z^{(k)}, y^{(k)}; F] - [y^{(k)}, x^{(k)}; F] \right)^{-1} F(z^{(k)}), \end{aligned}$$

being  $w^{(k)} = x^{(k)} + F(x^{(k)})$ .

In this case, 4 evaluations of  $F$  and 5 divided difference operators are calculated, so the number of functional evaluations is:

$$4n + 5(n^2 - n) = 5n^2 - n.$$

The number of operations performed is as follows: 5 different divided difference operators are calculated and 2 different LU decompositions are performed and a system is solved with each of the decompositions. Thus, the total number of operations is:

$$5n^2 + \frac{2}{3}(n^3 - n) + 2n^2 = \frac{2}{3}(n^3 - n) + 7n^2.$$

Thus, the number of total evaluations and operations is:

$$5n^2 - n + \frac{2}{3}(n^3 - n) + 7n^2 = \frac{2}{3}(n^3 - n) + 12n^2 - n.$$

Below are some images showing the efficiency index, the operational index and the computational efficiency index of the previously mentioned methods for different sizes of the system to be solved, in order to compare the different methods.

Figure 1 illustrates the efficiency rate of the different methods for different system sizes. In these figures, we can see that the methods with the highest efficiency rates are *CJTS5* and *CRTT*.

Figure 2 illustrates the operational efficiency index of the different methods for different system sizes. In these figures, we can see that for small sizes the methods *CJTS5* and *CRTT* are not as competitive, but when the size is greater than or equal to 10, these are the two methods with the highest operational efficiency index.

Figure 3 illustrates the computational efficiency index for different system sizes. As with the other indices, the methods *CJTS5* and *CRTT* stand out for system sizes greater than 1.

### 4 Numerical performance of CRTT

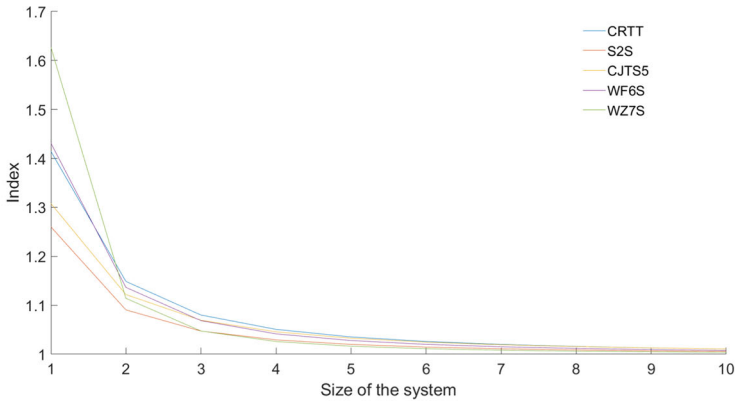
We compare some methods of CRTT class, namely CJF4S, TJF4S and CRTT4, with values of the parameters  $\psi = 0$ ,  $r = 1$  in all cases and  $\lambda = \{-4, -5, 0\}$ , respectively, with others, of high efficiency or order, S2S, CJTS5, WF6S and WZ7S, of orders 2, 5, 6 and 7, respectively, in the same problems, which have been define in the previous section. Note that CRTT4 is the version with the least computational effort of all of the family.

The maximum number of iterations considered is 50 and the stopping criterion is  $\|x^{(k+1)} - x^{(k)}\| < 10^{-100}$  or  $\|F(x^{(k+1)})\| < 10^{-100}$ . Each calculation was performed using Matlab R2022b with variable-precision arithmetic with a 500-digit mantissa to minimize round-off errors.

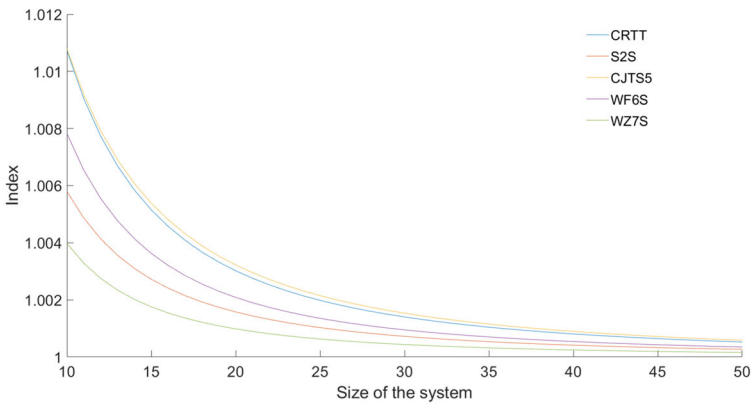
We must take into account that the approximate computational order of convergence (ACOC), see [11]:

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

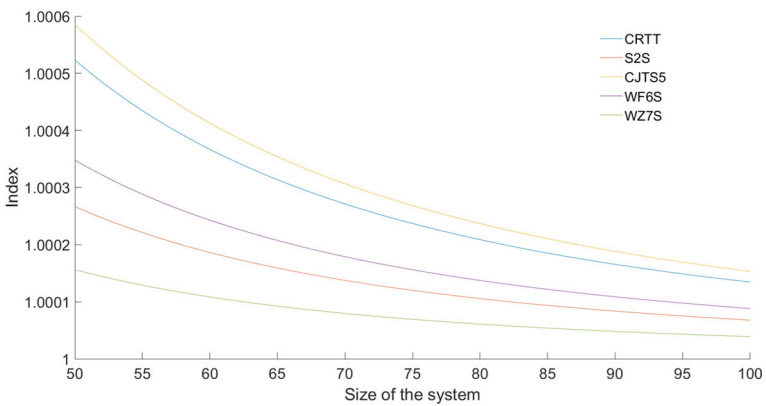
will be calculated that gives the numerical approximation of the theoretical order of an iterative method.



(a) For  $n = 1$  to  $n = 10$

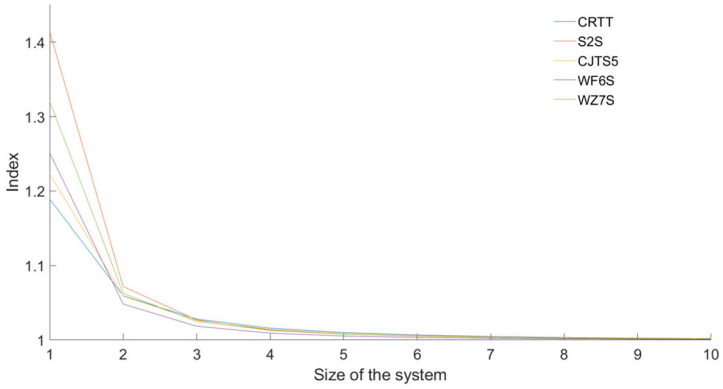


(b) For  $n = 10$  to  $n = 50$

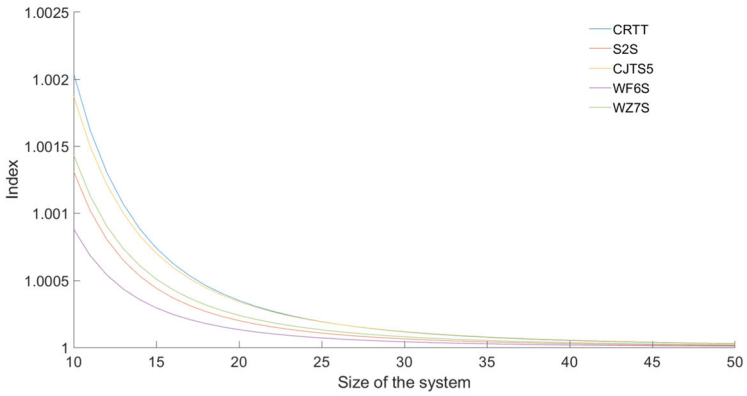


(c) For  $n = 50$  to  $n = 100$

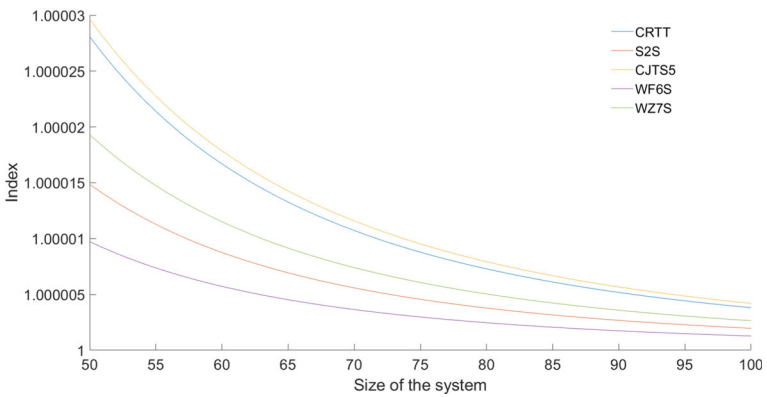
**Fig. 1** Efficiency index (a) For  $n = 1$  to  $n = 10$ , (b) For  $n = 10$  to  $n = 50$ , (c) For  $n = 50$  to  $n = 100$



(a) For  $n = 1$  to  $n = 10$

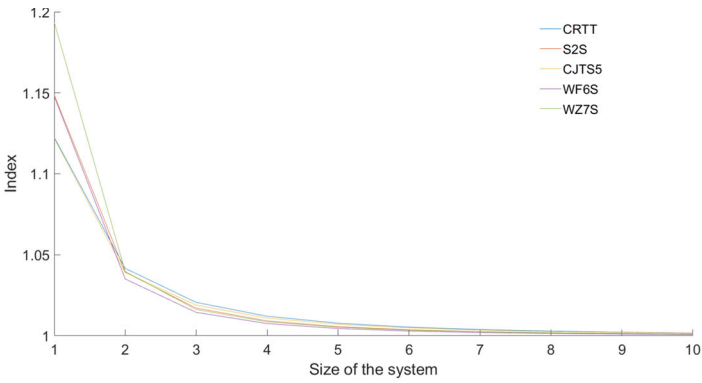


(b) For  $n = 10$  to  $n = 50$

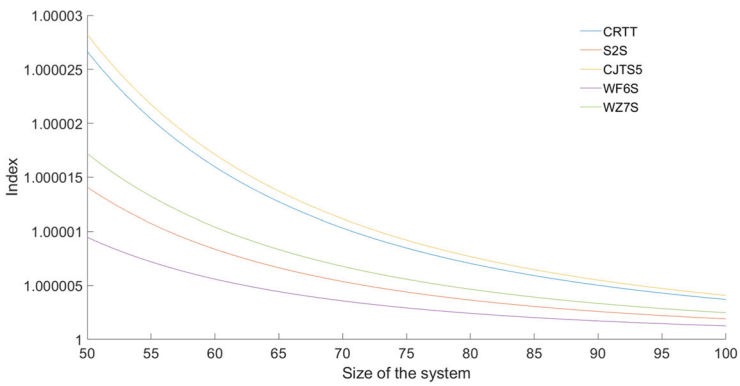


(c) For  $n = 50$  to  $n = 100$

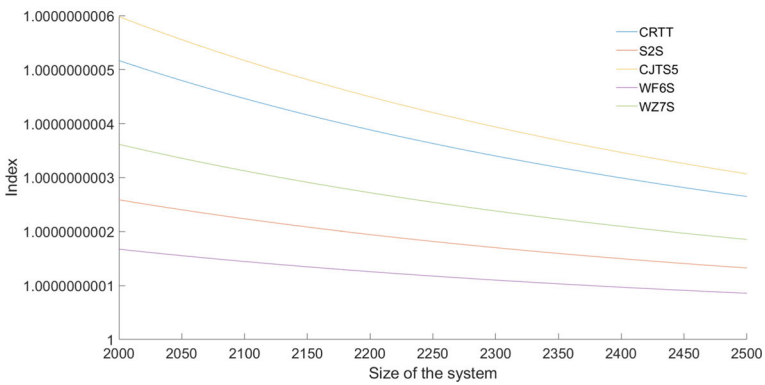
**Fig. 2** Operational efficiency index (a) For  $n = 1$  to  $n = 10$ , (b) For  $n = 10$  to  $n = 50$ , (c) For  $n = 50$  to  $n = 100$



(a) For  $n = 1$  to  $n = 10$



(b) For  $n = 50$  to  $n = 100$



(c) For  $n = 2000$  to  $n = 2500$

**Fig. 3** Computational efficiency index (a) For  $n = 1$  to  $n = 10$ , (b) For  $n = 50$  to  $n = 100$ , (c) For  $n = 2000$  to  $n = 2500$

### 4.1 Academical problems

Consider the system of equations:

$$-x_i - 3 + \sum_{j=1}^m x_j - e^{x_i} + 4 \cos(2 \ln(|x_i + 1|)) = 0, \quad i = 1, 2, \dots, m.$$

By taking  $m = 200$ , the initial estimate that we chose is  $x^{(0)} = (\frac{1}{100}, \dots, \frac{1}{100})^T$  to obtain the solution

$$\alpha = (0, 0, \dots, 0)^T.$$

In Table 1, the results obtained by the different methods for the academic problem are shown. The table shows that the elements of the CRTT family take the least computational time to reach the required tolerance, being considerably less than other methods that double their ACOC. Moreover, excluding the S2S method, they require the same iterations as the other methods with the highest order of convergence.

### 4.2 Application to boundary value problem

We consider a particular case of the partial derivative equation, called the transport equation (see [12]). This case is non-linear and non-homogeneous. This equation models physical phenomena, referring to the movement of different entities, such as inertial moment, mass or energy, through a medium, solid or fluid, with particular conditions that exist within the medium and it is expressed by the non-differentiable partial differential equation:

$$u_t + u_x = -2u|u|,$$

for  $x \in [0, 1], t \geq 0$ . The boundary conditions on  $u(x, t)$  are imposed as  $u(0, t) = \frac{1}{1+t}, u(1, t) = \frac{1}{2+t}$  for all  $t \geq 0$ , along with the initial condition as  $u(x, 0) = \frac{1}{1+x}$ , for  $0 \leq x \leq 1$ .

**Table 1** Academic problem

Method	CPUTime	Iter	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	ACOC
S2S	2.66	6	$4.35 \times 10^{-91}$	$1.14 \times 10^{-181}$	2.00
CJTS5	2.39	3	$4.10 \times 10^{-50}$	$1.93 \times 10^{-253}$	5.57
WF6S	4.14	3	$1.08 \times 10^{-53}$	$7.90 \times 10^{-320}$	6.07
WZ7S	5.00	3	$5.38 \times 10^{-97}$	$9.45 \times 10^{-507}$	8.81
CJF4S	1.91	3	$7.99 \times 10^{-27}$	$6.97 \times 10^{-106}$	4.14
TJF4S	1.94	3	$7.85 \times 10^{-27}$	$6.51 \times 10^{-106}$	4.14
CRTT4	1.66	3	$8.55 \times 10^{-27}$	$9.13 \times 10^{-106}$	4.14

System size = 200

Applying the method of characteristics, we have the system of ordinary differential equations:

$$\begin{cases} \frac{dt}{ds} = 1, \\ \frac{dx}{ds} = 1, \\ \frac{du}{ds} = -2u|u| \end{cases}$$

We solve the system numerically applying in each case the trapezium method (or second-order implicit Runge-Kutta). With the first two equations, we have that  $\Delta x + \Delta t = 2\Delta s$ .

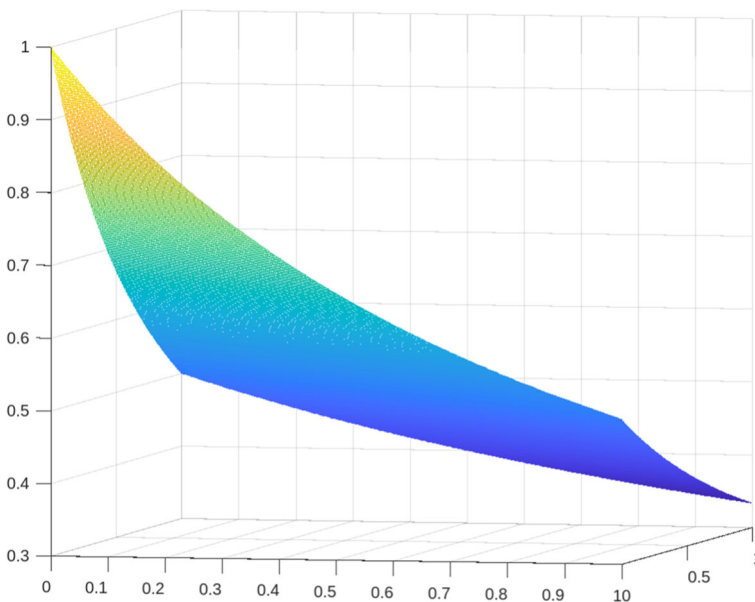
From the third equation, we form a system of  $n = 500$  equations, taking as step size  $\Delta s = (b - a)/(n - 1)$ , we obtain our initial approximation, applying the initial condition.

Now, we must solve the system of size  $500 \times 500$ , numerically:

$$\begin{aligned} u_1 - 1 &= 0, \\ u_i - u_{i-1} + \Delta s (u_i |u_i| + u_{i-1} |u_{i-1}|) &= 0, \quad i = 2, 3, \dots, 500. \end{aligned} \tag{14}$$

We take the same stopping criteria as in the academic problem. Next, we form the corresponding mesh, considering the domain

$$D = \{(x, t), (x, t) \in [0, 1] \times [0, 1]\},$$



**Fig. 4** Numerical approximation of the solution of the non-linear and non-homogeneous transport equation  $u_t + u_x = -2u|u|$



**Table 2** Boundary value problem

Method	CPUTime	Iter	$\ x^{(k+1)} - x^{(k)}\ $	$\ F(x^{(k+1)})\ $	ACOC
S2S	0.31	5	$8.16 \times 10^{-141}$	$8.17 \times 10^{-141}$	2.00
CJTSS	0.36	3	$2.96 \times 10^{-214}$	$2.96 \times 10^{-214}$	5.00
WF6S	–	–	–	–	–
WZ7S	0.10	1	0	$5.32 \times 10^{-505}$	–
CJF4S	0.28	3	$8.16 \times 10^{-141}$	$8.17 \times 10^{-141}$	4.00
TJF4S	0.24	3	$1.62 \times 10^{-217}$	$1.61 \times 10^{-217}$	5.00
CRTT4	0.26	3	$2.55 \times 10^{-137}$	$2.55 \times 10^{-137}$	4.00

System size = 500

with the partition of spatial step  $\Delta x = 1/(n - 1)$  and time step  $\Delta t = 1/(n - 1)$ , respectively, for the domains, from the solution of the first two equations, and we obtain the solution shown in Fig. 4.

In Table 2, the results obtained for the boundary value problem by the different methods are shown, where ‘–’ denotes lack of convergence or unstable value of the ACOC. In this table, we can see that the *WZ7S* method only requires a single iteration, while the rest require 3 iterations or more. We can also see that the *TJF4S* element of the *CRTT* family increases the ACOC by one unit for this problem.

The methods of the *CRTT* family maintain the theoretical order in very large systems, considering sizes 200 and 500. They can also compete with methods of equal or higher order, as can be seen in Tables 1 and 2.

About computational effort, note that the methods of *CRTT* family make much better use of the computational effort compared to the others.

## 5 Conclusion

In this work, a parametric family of Jacobian-free fourth-order convergence iterative methods has been presented. This class has been designed from the family defined in [4] in order to maintain the properties by modifying the Jacobian matrix by a divided difference operator, to make it suitable for non-differentiable problems. An efficiency study and several numerical experiments have been carried out, showing that it is a family of competitive methods compared to other known methods in the literature with similar characteristics.

**Acknowledgements** The authors would like to thank the anonymous reviewers for their suggestions and comments that have improved the final version of this manuscript.

**Author Contributions** RVRH and PTN wrote the draft version of the main manuscript text, JRT and AC wrote the revised version of the manuscript, RVRH prepared the figures, and the numerical tests have been made by RVRH and PTN. All authors reviewed the manuscript.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Availability of data and materials** Not applicable.

## Declarations

**Ethical approval** Not applicable.

**Conflict of interest** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Traub, J.F.: *Iterative Methods for the Solution of Equations*. Prentice-Hall, New York, USA (1964)
2. Amat, S., Busquier, S.: *Advances in Iterative Methods for Nonlinear Equations*, vol. 10. Springer, Switzerland (2016)
3. Arroyo, V., Cordero, A., Torregrosa, J.R.: Approximation of artificial satellites' preliminary orbits: the efficiency challenge. *Math. Comput. Model.* **54**(7), 1802–1807 (2011). <https://doi.org/10.1016/j.mcm.2010.11.063>
4. Cordero, A., Rojas-Hiciano, R., Torregrosa, J.R., Penkova, M.: A highly efficient class of optimal fourth-order methods for solving nonlinear systems. *Numer. Algorithm.* (2023). <https://doi.org/10.1007/s11075-023-01631-9>
5. Singh, H., Sharma, J.R., Kumar, S.: A simple yet efficient two-step fifth-order weighted-newton method for nonlinear models. *Numer. Algorithm.* **93**(1), 203–225 (2022). <https://doi.org/10.1007/s11075-022-01412-w>
6. Villalba, E.G., Hernandez, M., Hueso, J.L., Martínez, E.: Using decomposition of the nonlinear operator for solving non-differentiable problems. *Math. Methods Appl. Sci.* (2023). <https://doi.org/10.1002/mma.9455>
7. Samanskii, V.: On a modification of the Newton method. *Ukr. Math. J.* **19**, 133–138 (1967)
8. Cordero, A., Jordán, C., Sanabria, E., Torregrosa, J.R.: A new class of iterative processes for solving nonlinear systems by using one divided differences operator. *Mathematics.* **7**, 776 (2019). <https://doi.org/10.3390/math7090776>
9. Wang, X., Fan, X.: Two efficient derivative-free iterative methods for solving nonlinear systems. *Algorithms.* **9**, 14 (2016). <https://doi.org/10.3390/a9010014>
10. Wang, X., Zhang, T.: A family of Steffensen type methods with seventh-order convergence. *Numer. Algorithm.* **62**, 429–444 (2013). <https://doi.org/10.1007/s11075-012-9597-3>
11. Cordero, A., Torregrosa, J.R.: Variants of Newton's method using fifth-order quadrature formulas. *Appl. Math. Comput.* **190**, 686–698 (2007). <https://doi.org/10.1016/j.amc.2007.01.062>
12. Boutin, B., Nguyen, T.H.T., Sylla, A., Tran-Tien, S., Coulombel, J.-F.: High order numerical schemes for transport equations on bounded domains\*. *ESAIM: ProcS.* **70**, 84–106 (2021). <https://doi.org/10.1051/proc/202107006>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Authors and Affiliations

**Alicia Cordero**<sup>1</sup> · **Renso V. Rojas-Hiciano**<sup>2,3</sup> · **Juan R. Torregrosa**<sup>1</sup> · **Paula Triguero-Navarro**<sup>4</sup>

Renso V. Rojas-Hiciano  
rv.rojas@ce.pucmm.edu.do

Juan R. Torregrosa  
jr Torre@mat.upv.es

Paula Triguero-Navarro  
ptriguero@edem.es

- <sup>1</sup> Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Camí de Vera, s/n, Valencia 46022, Spain
- <sup>2</sup> Ciencias Naturales y Exactas, Pontificia Universidad Católica Madre y Maestra, Sarasota Ave., Santo Domingo, D.N. 10109, Dominican Republic
- <sup>3</sup> Ciencias Básicas y Ambientales, Instituto Tecnológico de Santo Domingo, Los Próceres, Santo Domingo, D.N. 10602, Dominican Republic
- <sup>4</sup> Centro Universitario EDEM-Escuela de empresarios, Plaça de L'aigua, s/n, Valencia 46024, Spain