Compact Approximate Taylor methods for systems of conservation laws.

H. Carrillo[1], C. Parés[1]

University of Málaga

Abstract

A new family of high order methods for systems of conservation laws are introduced: the Compact Approximate Taylor (CAT) methods. These methods are based on centered $(2p+1)$-point stencils where p is an arbitrary integer. We prove that the order of accuracy is $2p$ and that CAT methods are an extension of high-order Lax-Wendroff methods for linear problems. Due to this, they are linearly L^2-stable under a $CFL - 1$ condition. In order to prevent the spurious oscillations that appear close to discontinuities two shock-capturing techniques have been considered: a flux-limiter technique (FL-CAT methods) and WENO reconstruction for the first time derivative (WENO-CAT methods). We follow [14] in the second approach. A number of test cases are considered to compare these methods with other WENO-based schemes: the linear transport equation, Burgers equation, and the 1D compressible Euler system are considered. Although CAT methods present an extra computational cost due to the local character, this extra cost is compensated by the fact that they still give good solutions with CFL values close to 1.

1. Introduction

Lax-Wendroff methods for linear systems of conservation laws are based on Taylor expansions in time in which the time derivatives are transformed into spatial derivatives using the equations [5], [6],[12], [15]. The spatial derivatives are then discretized by means of centered high-order differentiation formulas. This procedure allows to derive numerical methods of order $2p$, where p is an arbitrary integer, using a centered $(2p+1)$-point stencil that are L^2-stable (a review on these methods will be presented in Section 2).

This paper focuses on the extension of Lax-Wendroff methods to nonlinear systems of conservation laws. This problem is closely related to the design of numerical schemes based on the methods of lines in which the time discretization is performed by means of Taylor or Approximate Taylor methods. Many authors have focused on the design of this type of methods that can be an alternative to time discretizations based on SSP Runge-Kutta methods [9] that lead to some stability restrictions for orders bigger than three. The main difficulty to extend Lax-Wendroff methods to nonlinear problems come from the transformation of time derivatives into spatial derivatives using the equations. A first strategy to do this is given by the Cauchy-Kovalevskaya (CK) procedure. In [9] this procedure has been used together with WENO reconstructions for

---

[1]pares@uma.es, hugo.carrillo@uma.es

the spatial discretization. The main benefit compared to RK time discretizations is that only one WENO reconstruction is needed at each spatial cell per time step. On the other hand, the main drawback comes from the fact that the CK procedure leads to expressions whose number of terms grow exponentially what implies high computational costs and difficult implementations. In [14] an alternative has been presented based on an Approximate Taylor (AT) method in which the time derivatives are approximated using high-order centered differentiation formulas combined with Taylor approximations in time that are computed in a recursive way. The resulting method is easy to implement and shows a good performance. Nevertheless AT schemes are not proper generalizations of Lax-Wendroff methods: they have $(4p + 1)$-point stencils and worse linear stability properties than the original Lax-Wendroff methods. Nevertheless, they can be stabilized by using one WENO reconstruction per spatial cell and time step, like in [9] the resulting methods give good results under a $CFL - 0.5$ condition typically.

In order to design numerical methods that are proper generalization of Lax-Wendroff methods, a compact variant of the AT procedure is introduced here: first, the conservative expression of the high-order Lax-Wendoff methods is considered; then, the derivatives appearing in the expression of the numerical flux are computed using $2p$-point differentiation formulas. This strategy lead to Compact Approximate Taylor (CAT) methods that have $2p + 1$-point stencil and order of accuracy $2p$. They reduce to the Lax-Wendroff method when applied to a linear systems and thus they are linearly L^2-stable under a $CFL - 1$ condition. Nevertheless, the number of operations to perform a step is bigger than the original AT methods: this is due to the fact that the approximation of the time derivatives is local in the sense that they depend both on the point and on the stencil. However, this extra cost is compensated by better stability properties. As it happens with its linear counterpart, CAT methods lead to spurious oscillations near discontinuities. In order to cure them, two shock-capturing techniques are considered here: a flux-limiter technique and the use of a WENO reconstruction per cell and time step, as in [14] and [9].

This paper is organized as follows: in Section 2 a review of high order Lax-Wendroff methods for the linear transport equation is presented, including the study of the order and the L^2-stability as well as the computation and properties of the coefficients. Section 3 is devoted to their extension to nonlinear problems: first the AT technique is recalled and then CAT methods are presented. We show that they reduce to Lax-Wendroff methods when applied to a linear problem and we analyze the order of accuracy. In Section 4 the techniques considered to cure the spurious oscillations near the discontinuities are presented. In Section 5 CAT methods are compared in a number of test cases with WENO-RK methods and AT methods. The linear transport equation, Burgers equation, and the 1D compressible Euler system are considered. Future developments and conclusions are drawn in Section 6.

2. The high-order Lax-Wendroff method for linear problems

Let us first consider the linear scalar equation:

$$u_t + au_x = 0. \tag{1}$$

We consider the numerical method:

$$u_i^{n+1} = u_i^n + \sum_{k=1}^{m} \frac{(-1)^k c^k}{k!} \sum_{j=-p}^{p} \delta_{p,j}^k \, u_{i+j}^n, \tag{2}$$

where $\{x_i\}$ are the nodes of a uniform mesh of step Δx; u_i^n is an approximation of the point value of the solution at x_i at the time $n\Delta t$, where Δt is the time step; $p \geq 1$ is a natural number; $c = a\Delta t/\Delta x$; and $\delta_{p,j}^k$ are the coefficients of the centered interpolatory formula of numerical differentiation based on a $(2p+1)$-point stencil:

$$f^{(k)}(x_i) \simeq D_{p,i}^k(f, \Delta x) = \frac{1}{\Delta x^k} \sum_{j=-p}^{p} \delta_{p,j}^k f(x_{i+j}), \tag{3}$$

where $f^{(k)}$ represents the k-th derivative of a one-variable function f and $f^{(0)} = f$. The expression of the numerical method is obtained by applying a Taylor expansion in time, and replacing time derivatives by space derivatives through the identities

$$\partial_t^k u = (-1)^k a^k \partial_x^k u, \quad k = 1, 2 \ldots \tag{4}$$

2.1. Formulas of numerical differentiation

Besides (3) the following family of interpolatory formulas based on a $2p$-point stencil will be used in this work:

$$f^{(k)}(x_i + q\Delta x) \simeq A_{p,i}^{k,q}(f, \Delta x) = \frac{1}{\Delta x^k} \sum_{j=-p+1}^{p} \gamma_{p,j}^{k,q} f(x_{i+j}), \tag{5}$$

i.e. $A_{p,i}^{k,q}(f, \Delta x)$ is the numerical differentiation formula that approximates the k-th derivative at the point $x_i + q\Delta x$ using the values of the function at the $2p$ points $x_{i-p+1}, \ldots, x_{i+p}$. Observe that the coefficients, like in (3), do not depend on i.

Given a variable w, the following notation will be used:

$$D_{p,i}^k(w_*, \Delta x) = \frac{1}{\Delta x^k} \sum_{j=-p}^{p} \delta_{p,j}^k w_{i+j},$$

$$A_{p,i}^{k,q}(w_*, \Delta x) = \frac{1}{\Delta x^k} \sum_{j=-p+1}^{p} \gamma_{p,j}^{k,q} w_{i+j},$$

to indicate that the formulas are applied to the approximations of w, w_i, and not to its exact point values $w(x_i)$. In cases where there are two or more indexes, the symbol $*$ will be used to indicate with respect to which the differentiation is applied. For instance:

$$\partial_x^k u(x_i + q\Delta x, t_n) \simeq A_{p,i}^{k,q}(u_*^n, \Delta x) = \frac{1}{\Delta x^k} \sum_{j=-p+1}^{p} \gamma_{p,j}^{k,q} u_{i+j}^n,$$

$$\partial_t^k u(x_i, t_n + q\Delta t) \simeq A_{p,n}^{k,q}(u_i^*, \Delta t) = \frac{1}{\Delta t^k} \sum_{r=-p+1}^{p} \gamma_{p,r}^{k,q} u_i^{n+r}.$$

3

Using this notation, the algorithm (2) writes as follows:

$$u_i^{n+1} = u_i^n + \sum_{k=1}^{m} \frac{(-1)^k a^k \Delta t^k}{k!} D_{p,i}^k(u_*^n, \Delta x). \tag{6}$$

Let us discuss some properties of the coefficients of the numerical differentiation formulas (3) and (5) and some relations between them that will be used in that follows. Since the coefficients are independent of Δx and i, we can consider, without loss of generality, the case $i = 0$, $x_0 = 0$, $\Delta x = 1$:

$$f^{(k)}(0) \simeq D_{p,0}^k(f, 1) = \sum_{j=-p}^{p} \delta_{p,j}^k f(j), \tag{7}$$

$$f^{(k)}(q) \simeq A_{p,0}^{k,q}(f, 1) = \sum_{j=-p+1}^{p} \gamma_{p,j}^{k,q} f(j). \tag{8}$$

Since (7) is exact for polynomials of degree $\leq 2p$, by applying the formula to x^s, $s = 0, \ldots, 2p$ at $x = 0$, we get that the coefficients have to satisfy the equalities

$$\sum_{j=-p}^{p} j^k \delta_{p,j}^k = k!, \qquad \sum_{j=-p}^{p} j^s \delta_{p,j}^k = 0, \quad s \neq k \quad , \quad 0 \leq s, k \leq 2p. \tag{9}$$

Analogously:

$$\sum_{j=-p+1}^{p} j^k \gamma_{p,j}^{k,0} = k!, \qquad \sum_{j=-p+1}^{p} j^s \gamma_{p,j}^{k,0} = 0, \quad s \neq k \quad , \quad 0 \leq s, k \leq 2p - 1. \tag{10}$$

$$\sum_{j=-p+1}^{p} \gamma_{p,j}^{k,q} = \begin{cases} 1 & \text{if } k = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

As it is well known, the coefficients $\delta_{p,j}^k$ are related to the Lagrange basis polynomials

$$F_{p,j}(x) = \prod_{r=-p, r \neq j}^{p} \frac{(x - r)}{(j - r)}, \quad -p \leq j \leq p, \tag{12}$$

through the equalities:

$$\delta_{p,j}^k = F_{p,j}^{(k)}(0), \tag{13}$$

which allow us to write the Taylor expansion of $F_{p,j}$ centered at $x = 0$ as follows:

$$F_{p,j}(x) = \sum_{k=0}^{2p} \frac{\delta_{p,j}^k}{k!} x^k. \tag{14}$$

4

Proposition 1 *The coefficients $\delta_{p,j}^k$ of the formula (7), satisfy:*

$$\delta_{p,j}^k = (-1)^k \delta_{p,-j}^k; \tag{15}$$

$$\delta_{p,0}^k = 0 \ \text{if } k \ \text{is odd;} \tag{16}$$

$$\sum_{j=-p}^{p} \delta_{p,j}^k \, j^{(2p+1)} = 0 \ \text{if } k \ \text{is even;} \tag{17}$$

$$\sum_{j=-p}^{p} \delta_{p,j}^k \, j^{(2p+2)} = 0 \ \text{if } k \ \text{is odd.} \tag{18}$$

Proof. (15) is deduced from the equality

$$F_{p,-j}(x) = F_{p,j}(-x);$$

. Using (15) we get (16). (17) and (18) are deduced from (15) and (16).

$\square$

Proposition 2 *For $k \geq 1$ the following relations hold:*

$$\delta_{p,p}^k = \gamma_{p,p}^{k-1,1/2}; \tag{19}$$

$$\delta_{p,j}^k = \gamma_{p,j}^{k-1,1/2} - \gamma_{p,j+1}^{k-1,1/2}, \quad j = -p+1, \ldots, p-1; \tag{20}$$

$$\delta_{p,-p}^k = -\gamma_{p,-p+1}^{k-1,1/2}. \tag{21}$$

Proof. Let us consider the formulas

$$f^{(k-1)}(1/2) \simeq A_{p,0}^{k-1,1/2}(f,1) = \sum_{j=-p+1}^{p} \gamma_{p,j}^{k-1} f(j), \tag{22}$$

$$f^{(k-1)}(-1/2) \simeq A_{p,-1}^{k-1,1/2}(f,1) = \sum_{j=-p+1}^{p} \gamma_{p,j}^{k-1} f(j-1), \tag{23}$$

that are exact for polynomials of degree $\leq 2p - 1$. Let us consider now the formula

$$f^{(k)}(0) \simeq A_{p,0}^{k-1,1/2}(f,1) - A_{p,-1}^{k-1,1/2}(f,1). \tag{24}$$

If f is a polynomial of degree $2p$, then (22) and (23) are exact for f, furthermore

$$A_{p,0}^{k-1,1/2}(f,1) - A_{p,-1}^{k-1,1/2}(f,1) = f^{(k-1)}(1/2) - f^{(k-1)}(-1/2) = f^{(k)}(0),$$

where we have used that the formula

$$g'(0) \simeq g(1/2) - g(-1/2),$$

5

is exact for polynomials of degree 1. Therefore, (24) coincide with (7). The proof is finished by writing (24) in the form

$$f^{(k)}(0) \simeq \gamma_{p,p}^{k-1,1/2} f(p) + (\gamma_{p,p-1}^{k-1,1/2} - \gamma_{p,p}^{k-1,1/2}) f(p-1) + \dots$$
$$+ (\gamma_{p,-p+1}^{k-1,1/2} - \gamma_{p,-p+2}^{k-1,1/2}) f(-p+1) - \gamma_{p,-p+1}^{k-1,1/2} f(-p),$$

and matching the weights.

$\square$

Proposition 3 *Given* $1 \le k \le 2p - 1$, $0 \le s \le k$:

$$\sum_{j=-p+1}^{p} \gamma_{p,j}^{s,q} \gamma_{p,l}^{k-s,j} = \gamma_{p,l}^{k,q}, \quad l = -p+1, \dots, p. \tag{25}$$

Proof. The proof is similar to the one of the preceding in Proposition 2: consider the formula

$$f^{(k)}(q) \simeq \sum_{j=-p+1}^{p} \gamma_{p,j}^{s,q} f_j^{(k-s)},$$

with

$$f_j^{(k-s)} = \sum_{l=-p+1}^{p} \gamma_{p,l}^{k-s,j} f(l);$$

check that it is exact for polynomials of degree $2p - 1$; write it in the form:

$$f^{(k)}(q) \simeq \sum_{l=-p+1}^{p} \left(\sum_{j=-p+1}^{p} \gamma_{p,j}^{s,q} \gamma_{p,l}^{k-s,j} \right) f(l);$$

and match its weights with those of (8).

$\square$

2.2. Conservative form of (2)

From the proof of Proposition 2 we deduce an alternative form for (3):

$$f^{(k)}(x_i) \simeq \frac{1}{\Delta x} \left(A_{p,i}^{k-1,1/2}(f, \Delta x) - A_{p,i-1}^{k-1,1/2}(f, \Delta x) \right). \tag{26}$$

Using this form in (6), the numerical method (2) can be written as:

$$u_i^{n+1} = u_i^n + \frac{\Delta t}{\Delta x} \left(F_{i-1/2}^p - F_{i+1/2}^p \right), \tag{27}$$

with

$$F_{i+1/2}^p = \sum_{k=1}^{2p} (-1)^{k-1} \frac{a^k \Delta t^{k-1}}{k!} A_{p,i}^{k-1,1/2}(u_*^n, \Delta x). \tag{28}$$

Using (11) it is straightforward to verify that $F_{i+1/2}^p$ is a consistent numerical flux, what proves that (2) is a conservative method.

6

2.3. Computation of the coefficients: an iterative algorithm

Notice that (9) constitutes a $(2p+1) \times (2p+1)$ linear system with a Vandermonde matrix that can be used to compute $\delta^k_{p,i}$. Nevertheless, as it is well-known, this system is ill-conditioned, so that it is recommendable to compute them by using an alternative algorithm: we adapt the recursive algorithm proposed in [3]. The following notation is adopted:

$$\delta^k_{p,j} = 0 \text{ if } k > 2p \text{ or } k < 0.$$

Let us derive some recurrence formulas to compute the coefficients:

1. $\delta^k_{p,j}$ for $j = 0, .., p-1$.
   From (12), we obtain

$$F_{p,j}(x) = \frac{x^2 - p^2}{j^2 - p^2} F_{p-1,j}(x). \tag{29}$$

   Using then the Taylor expansions (14) in (29) we get

$$\delta^k_{p,j} = \frac{1}{p^2 - k^2} \left[p^2 \delta^k_{p-1,j} - k(k-1)\delta^{k-2}_{p-1,j} \right], \tag{30}$$

2. $\delta^k_{p,j}$ with $j = p$.
   Substituting $j{=}p$ in (12), we get

$$F_{p,p}(x) = \frac{1}{(2p)(2p-1)}(x^2 + x - p(p-1))F_{p-1,p-1}(x), \tag{31}$$

   and, using (14), we obtain:

$$\delta^k_{p,p} = \frac{1}{2p(2p-1)} \left[k(k-1)\delta^{k-2}_{p-1,p-1} + k\delta^{k-1}_{p-1,p-1} - p(p-1)\delta^k_{p-1,p-1} \right]. \tag{32}$$

3. $\delta^k_{p,j}$ for $j = -p, \ldots, -1$. (15) is used.

The algorithm is computed only once in increasing order of p. The coefficients $\gamma^{k,q}_{p,j}$ are computed using the algorithms described in [3],[2] and $\gamma^{k,1/2}_{p,j}$ is obtained from $\delta^{k+1}_{p,j}$.

2.4. Order of accuracy

Proposition 4 *The formula of numerical differentiation (3) has order of accuracy $\alpha_k - k$, with,*

$$\alpha_k = \begin{cases} 2p + 1 & \text{if } k \text{ is odd,} \\ 2p + 2 & \text{if } k \text{ is even.} \end{cases}$$

Proof. Let f be a function of class C^{α_k+1}. Applying Taylor expansions and properties (9) and (17), we obtain:

$$\frac{1}{\Delta x^k} \sum_{j=-p}^{p} \delta^k_{p,j} f(x_{i+j}) = f^{(k)}(x_i) + \varphi_k \frac{\Delta x^{\alpha_k - k}}{\alpha_k!} f^{(\alpha_k)}(x_i) + \mathcal{O}(\Delta x^{\alpha_k - k + 1}), \tag{33}$$

k \ p	1	2	3	4
1	2	4	6	8
2	2	4	6	8
3		2	4	6
4		2	4	6
5			2	4
6			2	4
7				2
8				2

Table 1: Order of the formula (3).

where

$$\varphi_k = \sum_{j=-p}^{p} \delta_{p,j}^k \, j^{\alpha_k}. \tag{34}$$

$\square$

Table 1 shows the order of (3) for different values of p and k.

Proposition 5 *The discretization error of the numerical method (2) is of order $O(\Delta t^{m+1} + \Delta x^{2p+1})$.*

Proof. Let u be a smooth enough solution of (1). Using Proposition 4 we obtain

$$u(x_i, t_{n+1}) - u(x_i, t_n) - \sum_{k=1}^{m} \frac{(-1)^k c^k}{k!} \sum_{j=-p}^{p} \delta_{p,j}^k \, u(x_{i+j}, t_n)$$

$$= u(x_i, t_{n+1}) - u(x_i, t_n)$$
$$\quad - \sum_{k=1}^{m} \frac{(-1)^k a^k \Delta t^k}{k!} \left(\partial_x^k u(x_i, t_n) + \varphi_k \frac{\Delta x^{\alpha_k - k}}{\alpha_k!} \partial_x^{\alpha_k} u(x_i, t_n) + \mathcal{O}(\Delta x^{\alpha_k - k + 1}) \right)$$

$$= u(x_i, t_{n+1}) - u(x_i, t_n) - \sum_{k=1}^{m} \frac{\Delta t^k}{k!} \partial_t^k u(x_i, t_n)$$

$$\quad - \sum_{k=1}^{m} \varphi_k \frac{(-1)^k c^k}{k! \alpha_k!} \Delta x^{\alpha_k} \, \partial_x^{\alpha_k} u(x_i, t_n) + O(\Delta x^{\alpha_k + 1})$$

$$= \frac{1}{(m+1)!} \partial_t^{m+1} u(x_i, t_n) \Delta t^{m+1}$$

$$\quad + \left(\sum_{k=0}^{p-1} \frac{\varphi_{2k+1} c^{2k+1}}{(2p+1)!(2k+1)!} \right) \partial_x^{2p+1} u(x_i, t_n) \Delta x^{2p+1} + O(\Delta t^{m+2} + \Delta x^{2p+2}),$$

where (4) has been used.

8

□

As a consequence, the order of accuracy of (2) is $\min(m, 2p)$. Therefore, the optimal combination of these parameters is $m = 2p$. From now on, we shall assume that this relation holds.

2.5. Modified equation and stability

Taking into account that $m = 2p$ and (18), the local discretization error is as follows:

$$u(x_i, t_{n+1}) - u(x_i, t_n) - \sum_{k=1}^{m} \frac{(-1)^k c^k}{k!} \sum_{j=-p}^{p} \delta_{p,j}^k \, u(x_{i+j}, t_n)$$

$$= \frac{1}{(2p+1)!} \partial_t^{2p+1} u(x_i, t_n) \Delta t^{2p+1} + \frac{1}{(2p+2)!} \partial_t^{2p+2} u(x_i, t_n) \Delta t^{2p+2}$$

$$+ \left(\sum_{k=0}^{p-1} \frac{\varphi_{2k+1} c^{2k+1}}{(2p+1)!(2k+1)!} \right) \partial_x^{2p+1} u(x_i, t_n) \Delta x^{2p+1}$$

$$- \left(\sum_{k=1}^{p} \frac{\varphi_{2k} c^{2k}}{(2p+2)!(2k)!} \right) \partial_x^{2p+2} u(x_i, t_n) \Delta x^{2p+2} + O(\Delta x^{2p+3}).$$

Using (34) and (14) we get:

$$\sum_{k=0}^{p-1} \frac{\varphi_{2k+1} c^{2k+1}}{(2k+1)!} = \sum_{j=-p}^{p} \left(\sum_{k=0}^{p-1} \frac{\delta_{p,j}^{2k+1} c^{2k+1}}{(2k+1)!} \right) j^{2p+1}$$

$$= \frac{1}{2} \sum_{j=-p}^{p} \left(\sum_{l=1}^{2p} \left(\frac{\delta_{p,j}^l}{l!} - \frac{\delta_{p,-j}^l}{l!} \right) c^l \right) j^{2p+1}$$

$$= \frac{1}{2} \sum_{j=-p}^{p} \left(F_{p,j}(c) - F_{p,-j}(c) \right) j^{2p+1}$$

$$= \frac{1}{2} \left[\sum_{j=-p}^{p} F_{p,j}(c) j^{2p+1} - \sum_{j=-p}^{p} F_{p,j}(-c) j^{2p+1} \right]$$

$$= \frac{1}{2} \left[q(c) - q(-c) \right],$$

where $q(c)$ is the polynomial of degree $\leq 2p$ that interpolates the points

$$\{ (-p, (-p)^{2p+1}), \ldots, (0, 0), \ldots, (p, p^{2p+1}) \}.$$

Since q is clearly an odd function, we finally obtain:

$$\sum_{k=0}^{p-1} \frac{\varphi_{2k+1} c^{2k+1}}{(2k+1)!} = q(c). \tag{35}$$

9

Reasoning in a similar way, we get:

$$\sum_{k=1}^{p} \frac{\varphi_{2k} c^{2k}}{(2k)!} = r(c), \tag{36}$$

where r is the polynomial of degree $\leq 2p$ that interpolates the points

$$\{(-p, (-p)^{2p+2}), ..., (0, 0), ..., (p, p^{2p+2})\}.$$

Using now (4), (35), and (36), the local discretization error can be written as follows:

$$u(x_i, t_{n+1}) - u(x_i, t_n) - \sum_{k=1}^{m} \frac{(-1)^k c^k}{k!} \sum_{j=-p}^{p} \delta_{p,j}^k \, u(x_{i+j}, t_n)$$

$$= \frac{h_1(c)}{(2p+1)!} \partial_x^{2p+1} u(x_i, t_n) \Delta x^{2p+1} - \frac{h_2(c)}{(2p+2)!} \partial_x^{2p+2} u(x_i, t_n) \Delta x^{2p+2} + \mathcal{O}(\Delta^{2p+3}),$$

with

$$h_1(c) = q(c) - c^{2p+1}, \tag{37}$$
$$h_2(c) = r(c) - c^{2p+2}. \tag{38}$$

Therefore, the numerical method solves with order $\mathcal{O}(\Delta^{2p+2})$ the following modified equation

$$u_t + a u_x = \mu_1 \partial_x^{2p+1} u - \mu_2 \partial_x^{2p+2} u, \tag{39}$$

where

$$\mu_1 = \frac{h_1(c)}{(2p+1)! \Delta t} \Delta x^{2p+1}, \qquad \mu_2 = \frac{h_2(c)}{(2p+2)! \Delta t} \Delta x^{2p+2}. \tag{40}$$

In order to study the stability, let us look for an elementary solution $u(x, t)$ of (40) of the form

$$u(x, t) = e^{\alpha t} \cdot e^{ikx},$$

where α is complex number. The following equality has to be satisfied:

$$\alpha u + ikau = \mu_1 (-1)^p ik^{2p+1} u + \mu_2 (-1)^p k^{2p+2} u.$$

Therefore:

$$\alpha = -\mu_2 (-1)^{p+1} k^{2p+2} - (ka - \mu_1 (-1)^{p+1} k^{2p+1}) i.$$

The numerical method is thus expected to be stable if the real part is negative, i.e.

$$\mu_2 (-1)^p \leq 0,$$

or, equivalently

$$h_2(c)(-1)^p \leq 0. \tag{41}$$

h_2 is a pair polynomial of degree $2p + 2$ such that

$$\lim_{c \to \pm\infty} h_2(c) = -\infty.$$

10

Moreover, 0 is a double root of h_2 and $\pm 1, \ldots, \pm p$ are single roots. Analyzing the change of signs of h_2, it can be easily checked that

$$h_2(c) \leq 0, \quad \forall c \in [0,1] \quad \text{if } p \text{ even,}$$
$$h_2(c) \geq 0, \quad \forall c \in [0,1] \quad \text{if } p \text{ odd,}$$

and thus (41) is satisfied if $c \in [0,1]$ (see Figure 1). This argument shows that the stability of the method under the standard CFL condition $c \leq 1$ can be formalized: extended details and a formal proof can be found in [8]. A study of the modified equation for the Lax Wendroff method can also be found in [13].
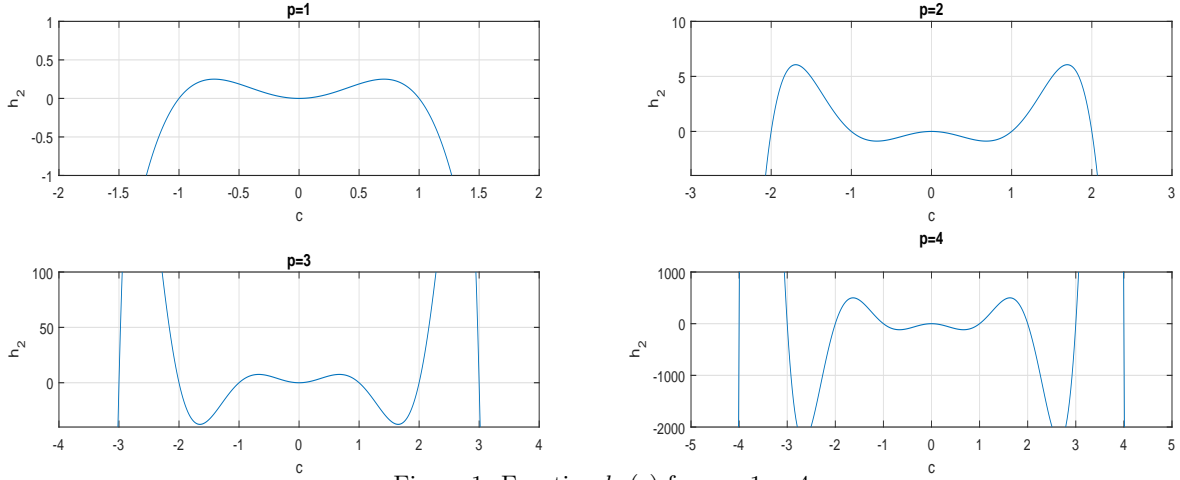

Figure 1: Function $h_2(c)$ for $p = 1, .., 4$.

3. Extension to nonlinear problems

3.1. Approximate Taylor method

Following [14], instead of using the Cauchy-Kovaleskaya process to extend (27)-(28) to nonlinear problems

$$u_t + f(u)_x = 0, \tag{42}$$

we use the equalities

$$\partial_t^k u = -\partial_x \partial_t^{k-1} f(u). \tag{43}$$

To derive the expression of the numerical method, let us suppose that approximations

$$\tilde{f}_i^{(k-1)} \cong \partial_t^{k-1} f(u)(x_i, t_n),$$

are available. Then,

$$\partial_t^k u(x_i, t_n) \cong \tilde{u}_i^{(k)} = -D_{p_{k-1},i}^1(\tilde{f}_*^{(k-1)}, \Delta x) = -\frac{1}{\Delta x} \sum_{j=-p_{k-1}}^{p_{k-1}} \delta_{p_{k-1},j}^1 \tilde{f}_{i+j}^{(k-1)},$$

11

being

$$p_k = \lceil (p - k/2) \rceil, \tag{44}$$

where, $\lceil \cdot \rceil$ denotes the ceiling function.

Using these approximations to approximate the Taylor expansion, we obtain the method

$$u_i^{n+1} = u_i^n + \sum_{k=0}^{2p} \frac{\Delta t^k}{k!} \tilde{u}_i^{(k)}. \tag{45}$$

Equivalently, using (26), the numerical method can be written in conservative form (27) with numerical flux

$$F_{i+1/2}^p = \sum_{k=1}^{2p} \frac{\Delta t^{k-1}}{k!} A_{p_{k-1},i}^{0,1/2}(\tilde{f}_*^{(k-1)}, \Delta x), \tag{46}$$

being

$$A_{p_{k-1},i}^{0,1/2}(\tilde{f}_*^{(k-1)}, \Delta x) = \sum_{j=-p_{k-1}+1}^{p_{k-1}} \gamma_{p_{k-1},j}^{0,1/2} \tilde{f}_{i+j}^{(k-1)}. \tag{47}$$

Now, to compute the approximations $\tilde{f}_i^{(k-1)}$, new Taylor expansions in time are used recursively as follows:

- Compute $\tilde{f}_i^{(0)} = f(u_i^n)$.

  - For $k = 1 \ldots 2p$:
    * Compute
    $$\tilde{u}_i^{(k)} = -D_{p_{k-1},i}^1(\tilde{f}_*^{(k-1)}, \Delta x).$$
    * Compute
    $$\tilde{f}_i^{k,n+r} = f\left(u_i^n + \sum_{l=1}^{k} \frac{(r\Delta t)^l}{l!} \tilde{u}_i^{(l)} \right), \quad r = -p, \ldots, p.$$
    * Compute
    $$\tilde{f}_i^{(k)} = D_{p,n}^k(\tilde{f}_i^{k,*}, \Delta t),$$
    where
    $$D_{p,n}^k(\tilde{f}_i^{k,*}, \Delta t) = \frac{1}{\Delta t^k} \sum_{r=-p}^{p} \delta_{p,r}^k \tilde{f}_i^{k,n+r}.$$

Observe that Taylor expansions are used to approximate $f(u(x_i, t_n + r\Delta t))$ and once these approximations have been computed, the centered formula of numerical differentiation (3) is used to approximate the temporal derivatives.

This method is order $2p$, but it is not a generalization of (2) in the sense that this latter method is not recovered if $f(u) = au$. To see this, consider $p = 1$ and $f(u) = au$: it can be easily checked that (45) writes as follows

$$u_i^{n+1} = u_i^n - \frac{c}{2}(u_{i+1}^n - u_{i-1}^n) - \frac{c^2}{8}(u_{i+2}^n - 2u_i^n + u_{i-2}^n), \tag{48}$$

12

which is different from the standard Lax-Wendroff method: (45) is a $(4p + 1)$-point method whose stability properties are worse than those of the standard Lax-Wendroff method. (see [6]).

3.2. Compact Approximate Taylor method

In order to prevent the increase of the stencil observed for Approximate Taylor methods, we consider a modification based on the conservative form of the method. The numerical flux $F_{i+1/2}^p$ will be computed using only the approximations

$$u_{i-p+1}^n, \ldots, u_{i+p}^n, \tag{49}$$

so that the values used to update u_i^{n+1} are only those of the centered $(2p+1)$-point stencil, like in the linear case. In fact, we will show that this modification is a proper generalization of the Lax-Wendroff method for linear problems.

In order to be able to compute the numerical fluxes using only (49), for every i we will compute *local* approximations of

$$\partial_t^{k-1} f(u(x_{i-p+1}, t^n), \ldots, \partial_t^{k-1} f(u(x_{i+p}, t^n),$$

that will be represented by

$$\tilde{f}_{i,j}^{(k-1)} \cong \partial_t^{k-1} f(u)(x_{i+j}, t_n), \quad j = -p+1, \ldots, p.$$

These approximations are local in the sense that $i_1 + j_1 = i_2 + j_2$, does not imply that $\tilde{f}_{i_1,j_1}^{(k-1)} = \tilde{f}_{i_2,j_2}^{(k-1)}$. Once these approximations have been computed, the numerical flux is given by

$$F_{i+1/2}^p = \sum_{k=1}^{2p} \frac{\Delta t^{k-1}}{k!} A_{p,0}^{0,1/2}(\tilde{f}_{i,*}^{(k-1)}, \Delta x), \tag{50}$$

with

$$A_{p,0}^{0,1/2}(\tilde{f}_{i,*}^{(k-1)}, \Delta x) = \sum_{j=-p+1}^{p} \gamma_{p,j}^{0,1/2} \tilde{f}_{i,j}^{(k-1)}. \tag{51}$$

Now, given i, to compute the approximations $\tilde{f}_{i,j}^{(k-1)}$, new Taylor expansions in time are used recursively as follows:

- Compute $\tilde{f}_{i,j}^{(0)} = f(u_{i+j}^n), \quad j = -p+1, \ldots, p.$

  - For $k = 1 \ldots 2p$:
    * Compute
    $$\tilde{u}_{i,j}^{(k)} = -A_{p,0}^{1,j}(\tilde{f}_{i,*}^{(k-1)}, \Delta x),$$
    where
    $$A_{p,0}^{1,j}(\tilde{f}_{i,*}^{(k-1)}, \Delta x) = \frac{1}{\Delta x} \sum_{r=-p+1}^{p} \gamma_{p,r}^{1,j} \tilde{f}_{i,r}^{(k-1)}.$$

13

* Compute

$$\tilde{f}_{i,j}^{k,n+r} = f\left(u_{i+j}^n + \sum_{l=1}^k \frac{(r\Delta t)^l}{l!}\tilde{u}_{i,j}^{(l)}\right), \quad j, r = -p+1, \ldots, p.$$

* Compute

$$\tilde{f}_{i,j}^{(k)} = A_{p,n}^{k,0}(\tilde{f}_{i,j}^{k,*}, \Delta t), \quad j = -p+1, \ldots, p.$$

with

$$A_{p,n}^{k,0}(\tilde{f}_{i,j}^{k,*}, \Delta t) = \frac{1}{\Delta t^k} \sum_{r=-p+1}^p \gamma_{p,r}^{k,0} \tilde{f}_{i,j}^{k,n+r}.$$

Notice that, unlike the Approximate Taylor methods (in which all the derivatives were approximated using the centered $(2p+1)$-point formula), in this algorithm the stencil $x_{i-p+1}, \ldots, x_{i+p}$ is used for the space derivatives and the stencil $t_{n-p+1}, \ldots, t_{n+p}$ for the time derivative.

Theorem 1 *The compact approximate Taylor method reduces to (2) when $f(u) = au$.*

Proof. For $k > 1$ we have:

$$
\begin{aligned}
\tilde{f}_{i,j}^{(k-1)} &= \frac{1}{\Delta t^{k-1}} \sum_{r=-p+1}^p \gamma_{p,r}^{k-1,0} \tilde{f}_{i,j}^{k-1,n+r} \\
&= \frac{a}{\Delta t^{k-1}} \sum_{r=-p+1}^p \gamma_{p,r}^{k-1,0} \left(u_{i+j}^n + \sum_{l=1}^{k-1} \frac{(r\Delta t)^l}{l!}\tilde{u}_{i,j}^{(l)}\right) \\
&= \frac{a}{\Delta t^{k-1}} \left(\left(\sum_{r=-p+1}^p \gamma_{p,r}^{k-1,0}\right) u_{i+j}^n + \sum_{l=1}^{k-1} \frac{\Delta t^l}{l!} \left(\sum_{r=-p+1}^p \gamma_{p,r}^{k-1,0} r^l\right) \tilde{u}_{i,j}^{(l)}\right) \\
&= a\tilde{u}_{i,j}^{(k-1)},
\end{aligned}
$$

where (10) has been used. On the other hand:

$$
\begin{aligned}
\tilde{u}_{i,j}^{(k)} &= -\frac{1}{\Delta x} \sum_{r=-p+1}^p \gamma_{p,r}^{1,j} \tilde{f}_{i,r}^{(k-1)} \\
&= -\frac{a}{\Delta x} \sum_{r=-p+1}^p \gamma_{p,r}^{1,j} \tilde{u}_{i,r}^{(k-1)} \\
&= \frac{a^2}{\Delta x^2} \sum_{r=-p+1}^p \gamma_{p,r}^{1,j} \sum_{s=-p+1}^p \gamma_{p,s}^{1,r} \tilde{u}_{i,s}^{(k-2)} \\
&= \frac{a^2}{\Delta x^2} \sum_{s=-p+1}^p \left(\sum_{r=-p+1}^p \gamma_{p,r}^{1,j} \gamma_{p,s}^{1,r}\right) \tilde{u}_{i,s}^{(k-2)} \\
&= \frac{a^2}{\Delta x^2} \sum_{s=-p+1}^p \gamma_{p,s}^{2,j} \tilde{u}_{i,s}^{(k-2)},
\end{aligned}
$$

14

where (25) has been used. By recurrence:

$$\tilde{u}_{i,j}^{(k)} = \frac{(-1)^k a^k}{\Delta x^k} \sum_{r=-p+1}^{p} \gamma_{p,r}^{k,j} u_{i+r}^n. \tag{52}$$

Next,

$$
\begin{aligned}
A_{p,0}^{0,1/2}(\tilde{f}_{i,*}^{(k-1)}, \Delta x) &= \frac{1}{\Delta x} \sum_{j=-p+1}^{p} \gamma_{p,j}^{0,1/2} \tilde{f}_{i,j}^{(k-1)} \\
&= \frac{a}{\Delta x} \sum_{j=-p+1}^{p} \gamma_{p,j}^{0,1/2} \tilde{u}_{i,j}^{(k-1)} \\
&= (-1)^{k-1} \frac{a^k}{\Delta x^k} \sum_{j=-p+1}^{p} \gamma_{p,j}^{0,1/2} \sum_{r=-p+1}^{p} \gamma_{p,r}^{k-1,j} u_{i+r}^n \\
&= (-1)^{k-1} \frac{a^k}{\Delta x^k} \sum_{r=-p+1}^{p} \left(\sum_{j=-p+1}^{p} \gamma_{p,j}^{0,1/2} \gamma_{p,r}^{k-1,j} \right) u_{i+r}^n \\
&= (-1)^{k-1} \frac{a^k}{\Delta x^k} \sum_{r=-p+1}^{p} \gamma_{p,j}^{k-1,1/2} u_{i+r}^n \\
&= (-1)^{k-1} a^k A_{p,i}^{k-1,1/2}(u_*^n, \Delta x),
\end{aligned}
$$

where (25) has been used. Finally,

$$
\begin{aligned}
F_{i+1/2}^p &= \sum_{k=1}^{2p} \frac{\Delta t^{k-1}}{k!} A_{p,0}^{0,1/2}(\tilde{f}_{i,*}^{(k-1)}, \Delta x) \\
&= \sum_{k=1}^{2p} (-1)^{k-1} \frac{a^k \Delta t^{k-1}}{k!} A_{p,i}^{k-1,1/2}(u_*^n, \Delta x),
\end{aligned}
$$

what is the numerical flux (28) corresponding to (2), as we wanted to prove.

□

As a consequence, we obtain that the compact approximate Taylor method is linearly stable (in the L^2 sense) under the usual CFL condition

$$\max_i(|f'(u_i)|)\frac{\Delta t}{\Delta x} \leq 1. \tag{53}$$

Theorem 2 *The compact approximate Taylor method is order $2p$.*

Proof. Let us perform a step of the method starting from the point values at time t^n, $u(x_i, t_n)$, of a smooth enough exact solution. We assume that $\Delta t/\Delta x$ remains constant.

First we have:

$$\tilde{u}_{i,j}^{(1)} = -A_{p,0}^{1,j}(\tilde{f}_{i,*}^{(0)}, \Delta x) = -\partial_x f(u)(x_{i+j}, t_n) + O(\Delta x^{2p-1}) = \partial_t u(x_{i+j}, t_n) + O(\Delta x^{2p-1}).$$

Next:
$$\tilde{f}_{i,j}^{1,n+r} = f(u(x_{i+j}, t_n) + \tilde{u}_{i,j}^{(1)} r\Delta t) = f(P_{i,j}^1(r\Delta t)) + O(\Delta x^{2p}),$$

where
$$P_{i,j}^1(s) = u(x_{i+j}, t_n) + s\partial_t u(x_{i+j}, t_n),$$

is the first order Taylor polynomial in time of u in (x_{i+j}, t_n). Then

$$
\begin{aligned}
\tilde{f}_{i,j}^{(1)} &= A_{p,n}^{1,0}(\tilde{f}_{i,j}^{k,*}, \Delta t) \\
&= \frac{1}{\Delta t} \sum_{r=-p+1}^{p} \gamma_{p,j}^{1,0} \tilde{f}_{i,j}^{1,n+r} \\
&= \frac{1}{\Delta t} \sum_{r=-p+1}^{p} \gamma_{p,j}^{1,0} f(P_{i,j}^1(r\Delta t)) + O(\Delta x^{2p}) \\
&= \frac{1}{\Delta t} \sum_{r=-p+1}^{p} \gamma_{p,j}^{1,0} \sum_{k=0}^{2p-1} \frac{1}{k!} d^k(f \circ P_{i,j}^1)(t_n) r^k \Delta t^k + O(\Delta x^{2p-1}) \\
&= \frac{1}{\Delta t} \sum_{k=0}^{2p-1} \frac{1}{k!} d^k(f \circ P_{i,j}^1)(t_n) \Delta t^k \sum_{r=-p+1}^{p} \gamma_{p,j}^{1,0} r^k + O(\Delta x^{2p-1}) \\
&= d^1(f \circ P_{i,j}^1)(t_n) + O(\Delta x^{2p-1}) \\
&= \partial_t f(u)(x_{i+j}, t_n) + O(\Delta x^{2p-1}),
\end{aligned}
$$

where (10) has been used. This result can be extended by induction to every k between 1 and $2p-1$ as follows:

$$\tilde{f}_{i,j}^{(k)} = \partial_t^k f(u)(x_{i+j}, t_n) + O(\Delta t^{2p-k}), \quad k = 1, \dots, 2p-1. \tag{54}$$

Using this equality we get:

$$u(x_i, t_{n+1}) - u(x_i, t_n) + \frac{\Delta t}{\Delta x} \left(F^p_{i+1/2} - F^p_{i-1/2} \right)$$

$$= u(x_i, t_{n+1}) - u(x_i, t_n) + \frac{1}{\Delta x} \sum_{k=1}^{2p} \frac{\Delta t^k}{k!} \left(A^{0,1/2}_{p,0}(\tilde{f}^{(k-1)}_{i,*}, \Delta x) - A^{0,1/2}_{p,0}(\tilde{f}^{(k-1)}_{i-1,*}, \Delta x) \right)$$

$$= u(x_i, t_{n+1}) - u(x_i, t_n) + \frac{1}{\Delta x} \sum_{k=1}^{2p} \frac{\Delta t^k}{k!} \left(A^{0,1/2}_{p,i}(\partial_t^{k-1} f(u), \Delta x) - A^{0,1/2}_{p,i-1}(\partial_t^{k-1} f(u), \Delta x) \right)$$

$$+ O(\Delta x^{2p+1})$$

$$= u(x_i, t_{n+1}) - u(x_i, t_n) + \frac{1}{\Delta x} \sum_{k=1}^{2p} \frac{\Delta t^k}{k!} D^1_{p,i}(\partial_t^{k-1} f(u), \Delta x) + O(\Delta x^{2p+1})$$

$$= u(x_i, t_{n+1}) - u(x_i, t_n) + \frac{1}{\Delta x} \sum_{k=1}^{2p} \frac{\Delta t^k}{k!} \partial_t^{k-1} f(u)(x_i, t_n) + O(\Delta x^{2p+1})$$

$$= u(x_i, t_{n+1}) - u(x_i, t_n) - \frac{1}{\Delta x} \sum_{k=1}^{2p} \frac{\Delta t^k}{k!} \partial_t^k u(x_i, t_n) + O(\Delta x^{2p+1})$$

$$= O(\Delta x^{2p+1}).$$

$\square$

Remark: In the Approximate Taylor method proposed in [14] the derivatives $\tilde{u}_i^{(k+1)}$ are computed by applying the $2p_k + 1$-point centered differentiation formula for first derivatives to $\tilde{f}_i^{(k)}$, where p_k is given by (44): notice that p_k decreases as k increases. The same reduction of the stencil used to compute $\tilde{u}_{i,j}^{(k)}$ could be applied here, what would allow us to reduce the number of computations while preserving the overall order of accuracy. Nevertheless, the resulting method will not be an extension of the linear Lax-Wendroff method. On the other hand, the CPU reduction will be not significant.

3.3. Example: Fourth order compact approximate Taylor method

Since the method is conservative, we will only show in detail how to compute the numerical flux (50) and, to do this, it is enough to specify how to compute

$$\kappa^k_{i+1/2} = A^{0,1/2}_{2,0}(\tilde{f}^{(k-1)}_{i,*}, \Delta x), \quad k = 1, 2, 3, 4. \tag{55}$$

The procedure is as follows:

- $\kappa^1_{i+1/2}$: First the assignment

$$\tilde{f}^{(0)}_{i,j} = f(u^n_{i+j}), \quad j = -1, ..., 2$$

is done and then:

$$\kappa^1_{i+1/2} = A^{0,1/2}_{2,0}(\tilde{f}^{(0)}_{i,*}, \Delta x) = \frac{-\tilde{f}^{(0)}_{i,-1} + 7\tilde{f}^{(0)}_{i,0} + 7\tilde{f}^{(0)}_{i,1} - \tilde{f}^{(0)}_{i,2}}{12}.$$

- $\kappa^2_{i+1/2}$: The first order time derivatives of u at the nodes $i-1, \ldots, i+2$ are approximated by applying the corresponding differentiation numerical formula to $\tilde{f}^{(0)}_{i,j}$:

$$\tilde{u}^{(1)}_{i,-1} = -A^{1,-1}_{2,0}(\tilde{f}^{(0)}_{i,*}, \Delta x) = -\frac{11/6\tilde{f}^{(0)}_{i,-1} - 3\tilde{f}^{(0)}_{i,0} + 3/2\tilde{f}^{(0)}_{i,1} - 1/3\tilde{f}^{(0)}_{i,2}}{\Delta x},$$

$$\tilde{u}^{(1)}_{i,0} = -A^{1,0}_{2,0}(\tilde{f}^{(0)}_{i,*}, \Delta x) = -\frac{1/3\tilde{f}^{(0)}_{i,-1} + 1/2\tilde{f}^{(0)}_{i,0} - \tilde{f}^{(0)}_{i,1} + 1/6\tilde{f}^{(0)}_{i,2}}{\Delta x},$$

$$\tilde{u}^{(1)}_{i,1} = -A^{1,1}_{2,0}(\tilde{f}^{(0)}_{i,*}, \Delta x) = -\frac{-1/6\tilde{f}^{(0)}_{i,-1} + \tilde{f}^{(0)}_{i,0} - 1/2\tilde{f}^{(0)}_{i,1} - 1/3\tilde{f}^{(0)}_{i,2}}{\Delta x},$$

$$\tilde{u}^{(1)}_{i,2} = -A^{1,2}_{2,0}(\tilde{f}^{(0)}_{i,*}, \Delta x) = -\frac{1/3\tilde{f}^{(0)}_{i,-1} - 3/2\tilde{f}^{(0)}_{i,0} + 3\tilde{f}^{(0)}_{i,1} - 11/6\tilde{f}^{(0)}_{i,2}}{\Delta x}.$$

Next first order Taylor expansions are used to approximate the values of the flux sixteen space-time local nodes: for $r = -1, \ldots, 2$

$$\tilde{f}^{1,r}_{i,-1} = f(u^n_{i-1} + r\Delta t\,\tilde{u}^{(1)}_{i,-1}),$$

$$\tilde{f}^{1,r}_{i,0} = f(u^n_{i,0} + r\Delta t\,\tilde{u}^{(1)}_{i,0}),$$

$$\tilde{f}^{1,r}_{i,1} = f(u^n_{i+1} + r\Delta t\,\tilde{u}^{(1)}_{i,+1}),$$

$$\tilde{f}^{1,r}_{i,2} = f(u^n_{i+2} + r\Delta t\,\tilde{u}^{(1)}_{i,+2}).$$

Then, the first order time derivates of the flux at the nodes $i-1, \ldots, i+2$ are approximated by applying the corresponding differentiation numerical formula to $\tilde{f}^{1,r}_{i,j}$:

$$\tilde{f}^{(1)}_{i,-1} = A^{1,0}_{2,n}(\tilde{f}^{1,*}_{i,-1}, \Delta t) = \frac{-1/3\tilde{f}^{1,n-1}_{i,-1} - 1/2\tilde{f}^{1,n}_{i,-1} + \tilde{f}^{1,n+1}_{i,-1} - 1/6\tilde{f}^{1,n+2}_{i,-1}}{\Delta t},$$

$$\tilde{f}^{(1)}_{i,0} = A^{1,0}_{2,n}(\tilde{f}^{1,*}_{i,0}, \Delta t) = \frac{-1/3\tilde{f}^{1,n-1}_{i,0} - 1/2\tilde{f}^{1,n}_{i,0} + \tilde{f}^{1,n+1}_{i,0} - 1/6\tilde{f}^{1,n+2}_{i,0}}{\Delta t},$$

$$\tilde{f}^{(1)}_{i,1} = A^{1,0}_{2,n}(\tilde{f}^{1,*}_{i,1}, \Delta t) = \frac{-1/3\tilde{f}^{1,n-1}_{i,1} - 1/2\tilde{f}^{1,n}_{i,1} + \tilde{f}^{1,n+1}_{i,1} - 1/6\tilde{f}^{1,n+2}_{i,1}}{\Delta t},$$

$$\tilde{f}^{(1)}_{i,2} = A^{1,0}_{2,n}(\tilde{f}^{1,*}_{i,2}, \Delta t) = \frac{-1/3\tilde{f}^{1,n-1}_{i,2} - 1/2\tilde{f}^{1,n}_{i,2} + \tilde{f}^{1,n+1}_{i,2} - 1/6\tilde{f}^{1,n+2}_{i,2}}{\Delta t}.$$

Finally;

$$\kappa^2_{i+1/2} = A^{0,1/2}_{2,0}(\tilde{f}^{(1)}_{i,*}, \Delta x) = \frac{-\tilde{f}^{(1)}_{i,-1} + 7\tilde{f}^{(1)}_{i,0} + 7\tilde{f}^{(1)}_{i,1} - \tilde{f}^{(1)}_{i,2}}{12}.$$

- $\kappa^3_{i+1/2}$: the second order time derivatives at the nodes are approximated by

$$\tilde{u}^{(2)}_{i,-1} = -A^{1,-1}_{2,0}(\tilde{f}^{(1)}_{i,*}, \Delta x) = -\frac{11/6\tilde{f}^{(1)}_{i,-1} - 3\tilde{f}^{(1)}_{i,0} + 3/2\tilde{f}^{(1)}_{i,1} - 1/3\tilde{f}^{(1)}_{i,2}}{\Delta x},$$

$$\tilde{u}^{(2)}_{i,0} = -A^{1,0}_{2,0}(\tilde{f}^{(1)}_{i,*}, \Delta x) = -\frac{1/3\tilde{f}^{(1)}_{i,-1} + 1/2\tilde{f}^{(1)}_{i,0} - \tilde{f}^{(1)}_{i,1} + 1/6\tilde{f}^{(1)}_{i,2}}{\Delta x},$$

$$\tilde{u}^{(2)}_{i,1} = -A^{1,1}_{2,0}(\tilde{f}^{(1)}_{i,*}, \Delta x) = -\frac{-1/6\tilde{f}^{(1)}_{i,-1} + \tilde{f}^{(1)}_{i,0} - 1/2\tilde{f}^{(1)}_{i,1} - 1/3\tilde{f}^{(1)}_{i,2}}{\Delta x},$$

$$\tilde{u}^{(2)}_{i,2} = -A^{1,2}_{2,0}(\tilde{f}^{(1)}_{i,*}, \Delta x) = -\frac{1/3\tilde{f}^{(1)}_{i,-1} - 3/2\tilde{f}^{(1)}_{i,0} + 3\tilde{f}^{(1)}_{i,1} - 11/6\tilde{f}^{(1)}_{i,2}}{\Delta x}.$$

Second order Taylor expansions are used to compute the fluxes at the sixteen nodes in the space-time mesh: for $r = -1, \ldots 2$

$$\tilde{f}^{2,r}_{i,-1} = f\left(u^n_{i-1} + r\Delta t\,\tilde{u}^{(1)}_{i,-1} + \tfrac{r^2\Delta t^2}{2}\,\tilde{u}^{(2)}_{i,-1}\right),$$

$$\tilde{f}^{2,r}_{i,0} = f\left(u^n_i + r\Delta t\,\tilde{u}^{(1)}_{i,0} + \tfrac{r^2\Delta t^2}{2}\,\tilde{u}^{(2)}_{i,0}]\right),$$

$$\tilde{f}^{2,r}_{i,1} = f\left(u^n_{i+1} + r\Delta t\,\tilde{u}^{(1)}_{i,1} + \tfrac{r^2\Delta t^2}{2}\,\tilde{u}^{(2)}_{i,1}\right),$$

$$\tilde{f}^{2,r}_{i,2} = f\left(u^n_{i+2} + r\Delta t\,\tilde{u}^{(1)}_{i,2} + \tfrac{r^2\Delta t^2}{2}\,\tilde{u}^{(2)}_{i,2}\right).$$

Next, compute

$$\tilde{f}^{(2)}_{i,-1} = A^{2,0}_{2,n}(\tilde{f}^{2,*}_{i,-1}, \Delta t) = \frac{\tilde{f}^{2,n-1}_{i,-1} - 2\tilde{f}^{2,n}_{i,-1} + \tilde{f}^{2,n+1}_{i,-1}}{\Delta t^2},$$

$$\tilde{f}^{(2)}_{i,0} = A^{2,0}_{2,n}(\tilde{f}^{2,*}_{i,0}, \Delta t) = \frac{\tilde{f}^{2,n-1}_{i,0} - 2\tilde{f}^{2,n}_{i,0} + \tilde{f}^{2,n+1}_{i,0}}{\Delta t^2},$$

$$\tilde{f}^{(2)}_{i,1} = A^{2,0}_{2,n}(\tilde{f}^{2,*}_{i,1}, \Delta t) = \frac{\tilde{f}^{2,n-1}_{i,1} - 2\tilde{f}^{2,n}_{i,1} + \tilde{f}^{2,n+1}_{i,1}}{\Delta t^2},$$

$$\tilde{f}^{(2)}_{i,2} = A^{2,0}_{2,n}(\tilde{f}^{2,*}_{i,2}, \Delta t) = \frac{\tilde{f}^{2,n-1}_{i,2} - 2\tilde{f}^{2,n}_{i,2} + \tilde{f}^{2,n+1}_{i,2}}{\Delta t^2}.$$

And finally;

$$\kappa^3_{i+1/2} = A^{0,1/2}_{2,0}(\tilde{f}^{(2)}_{i,*}, \Delta x) = \frac{-\tilde{f}^{(2)}_{i,-1} + 7\tilde{f}^{(2)}_{i,0} + 7\tilde{f}^{(2)}_{i,1} - \tilde{f}^{(2)}_{i,2}}{12}.$$

- $\kappa^4_{i+1/2}$: the third order time derivatives at the nodes are approximated by

$$\tilde{u}^{(3)}_{i,-1} = -A^{1,-1}_{2,0}(\tilde{f}^{(2)}_{i,*}, \Delta x) = -\frac{11/6\tilde{f}^{(2)}_{i,-1} - 3\tilde{f}^{(2)}_{i,0} + 3/2\tilde{f}^{(2)}_{i,1} - 1/3\tilde{f}^{(2)}_{i,2}}{\Delta x},$$

$$\tilde{u}^{(3)}_{i,0} = -A^{1,0}_{2,0}(\tilde{f}^{(2)}_{i,*}, \Delta x) = -\frac{1/3\tilde{f}^{(2)}_{i,-1} + 1/2\tilde{f}^{(2)}_{i,0} - \tilde{f}^{(2)}_{i,1} + 1/6\tilde{f}^{(2)}_{i,2}}{\Delta x},$$

$$\tilde{u}^{(3)}_{i,1} = -A^{1,1}_{2,0}(\tilde{f}^{(2)}_{i,*}, \Delta x) = -\frac{-1/6\tilde{f}^{(2)}_{i,-1} + \tilde{f}^{(2)}_{i,0} - 1/2\tilde{f}^{(2)}_{i,1} - 1/3\tilde{f}^{(2)}_{i,2}}{\Delta x},$$

$$\tilde{u}^{(3)}_{i,2} = -A^{1,2}_{2,0}(\tilde{f}^{(2)}_{i,*}, \Delta x) = -\frac{1/3\tilde{f}^{(2)}_{i,-1} - 3/2\tilde{f}^{(2)}_{i,0} + 3\tilde{f}^{(2)}_{i,1} - 11/6\tilde{f}^{(2)}_{i,2}}{\Delta x}.$$

Compute the approximations of the fluxes: for $r = -1, \ldots, 2$

$$\tilde{f}^{3,r}_{i,-1} = f\left(u^n_{i-1} + r\Delta t\, \tilde{u}^{(1)}_{i,-1} + \frac{r^2\Delta t^2}{2}\tilde{u}^{(2)}_{i,-1} + \frac{r^3\Delta t^3}{6}\tilde{u}^{(3)}_{i,-1}\right),$$

$$\tilde{f}^{3,r}_{i,0} = f\left(u^n_i + r\Delta t\, \tilde{u}^{(1)}_{i,0} + \frac{r^2\Delta t^2}{2}\tilde{u}^{(2)}_{i,0} + \frac{r^3\Delta t^3}{6}\tilde{u}^{(3)}_{i,0}\right),$$

$$\tilde{f}^{3,r}_{i,1} = f\left(u^n_{i+1} + r\Delta t\, \tilde{u}^{(1)}_{i,1} + \frac{r^2\Delta t^2}{2}\tilde{u}^{(2)}_{i,1} + \frac{r^3\Delta t^3}{6}\tilde{u}^{(3)}_{i,1}\right),$$

$$\tilde{f}^{3,r}_{i,2} = f\left(u^n_{i+2} + r\Delta t\, \tilde{u}^{(1)}_{i,2} + \frac{r^2\Delta t^2}{2}\tilde{u}^{(2)}_{i,2} + \frac{r^3\Delta t^3}{6}\tilde{u}^{(3)}_{i,2}\right).$$

Next, compute:

$$\tilde{f}^{(3)}_{i,-1} = A^{3,0}_{2,n}(\tilde{f}^{3,*}_{i,-1}, \Delta t) = \frac{-\tilde{f}^{3,n-1}_{i,-1} + 3\tilde{f}^{3,n}_{i,-1} - 3\tilde{f}^{3,n+1}_{i,-1} + \tilde{f}^{3,n+2}_{i,-1}}{\Delta t^3},$$

$$\tilde{f}^{(3)}_{i,0} = A^{3,0}_{2,n}(\tilde{f}^{3,*}_{i,0}, \Delta t) = \frac{-\tilde{f}^{1,n-1}_{i,0} + 3\tilde{f}^{3,n}_{i,0} - 3\tilde{f}^{3,n+1}_{i,0} + \tilde{f}^{3,n+2}_{i,0}}{\Delta t^3},$$

$$\tilde{f}^{(3)}_{i,1} = A^{3,0}_{2,n}(\tilde{f}^{3,*}_{i,1}, \Delta t) = \frac{-\tilde{f}^{3,n-1}_{i,1} + 3\tilde{f}^{3,n}_{i,1} - 3\tilde{f}^{3,n+1}_{i,1} + \tilde{f}^{3,n+2}_{i,1}}{\Delta t^3},$$

$$\tilde{f}^{(3)}_{i,2} = A^{3,0}_{2,n}(\tilde{f}^{3,*}_{i,2}, \Delta t) = \frac{-\tilde{f}^{3,n-1}_{i,2} + 3\tilde{f}^{3,n}_{i,2} - 3\tilde{f}^{3,n+1}_{i,2} + \tilde{f}^{3,n+2}_{i,2}}{\Delta t^3}.$$

Finally;

$$\kappa^4_{i+1/2} = A^{0,1/2}_{2,0}(\tilde{f}^{(2)}_{i,*}, \Delta x) = \frac{-\tilde{f}^{(2)}_{i,-1} + 7\tilde{f}^{(2)}_{i,0} + 7\tilde{f}^{(2)}_{i,1} - \tilde{f}^{(2)}_{i,2}}{12}.$$

If $f(u) = au$, then:

$$F^2_{i+1/2} = \frac{a}{12}(-u^n_{i-1} + 7u^n_i + 7u^n_{i+1} - u^n_{i+2}) + \frac{a^2\Delta t}{24\Delta x}(-u^n_{i-1} + 15u^n_i - 15u^n_{i+1} + u^n_{i+2}) \quad (56)$$

$$+ \frac{a^3\Delta t^2}{12\Delta x^2}(u^n_{i-1} - u^n_i - u^n_{i+1} + u^n_{i+2}) + \frac{a^4\Delta t^3}{24\Delta x^3}(u^n_{i-1} - 3u^n_i + 3u^n_{i+1} - u^n_{i+2}), \quad (57)$$

which coincides with the numerical flux of the fourth order Lax-Wendroff in conservative form.

4. Shock-capturing techniques

Although the Compact Approximate Taylor methods are linearly stable in the L^2 sense under the usual CFL-1 condition, they may produce strong oscillations close to a discontinuity of the solution. The goal of this section is to modify the numerical method to avoid these oscillations. Two different techniques are considered here:

4.1. FLUX LIMITER-CAT methods

We consider the numerical method (27) with

$$F_{i+1/2} = (1 - \varphi_{i+1/2})F^L_{i+1/2} + \varphi_{i+1/2}F^p_{i+1/2}, \tag{58}$$

where $F^L_{i+1/2}$ is a first order robust numerical flux, $F^p_{i+1/2}$ is given by (50), and $\varphi_{i+1/2}$ is the flux limiter function, see [4] [7], [12]. We consider here

$$\varphi_{i+1/2} = \varphi(r_{i+1/2}), \tag{59}$$

where φ is the van Albada second version flux limiter:

$$\varphi(r) = \max\left(0, \frac{2r}{1 + r^2}\right), \tag{60}$$

and

$$r_{i+1/2} = \frac{\Delta upw}{\Delta loc} = \begin{cases} \dfrac{u^n_i - u^n_{i-1}}{u^n_{i+1} - u^n_i} & \text{if } a_{i+1/2} > 0, \\ \dfrac{u^n_{i+2} - u^n_{i+1}}{u^n_{i+1} - u^n_i} & \text{if } a_{i+1/2} < 0, \end{cases}$$

where, $a_{i+1/2}$ is an estimate of the wave speed.

4.2. WENO-CAT methods

Following [9] WENO reconstructions of the flux are used to stabilize the method. The only differences with the algorithm described in Section 3.2 are the computation of $\tilde{u}^{(1)}_{i,j}$, that is now performed as follows:

$$\tilde{u}^{(1)}_{i,j} = -\frac{\hat{f}_{i+j+1/2} - \hat{f}_{i+j-1/2}}{\Delta x},$$

where $\hat{f}_{i+1/2}$ denotes the WENO flux splitting, reconstructions at $x_{i+1/2}$ of the flux function described in [10]. The expression of the numerical flux is then given by:

$$F^p_{i+1/2} = \hat{f}_{i+1/2} + \sum_{k=2}^{2p} \frac{\Delta t^{k-1}}{k!} A^{0,1/2}_{p,0}(\tilde{f}^{(k-1)}_{i,*}, \Delta x). \tag{61}$$

4.3. Systems of conservation laws

Although the methods have been described in the scalar case for simplicity, they can be easily applied to systems using vector notation.

5. Numerical Experiments

In this section we apply to some scalar conservation laws and to the 1D Euler equation the following numerical methods:

- LW-CATq: Compact Approximate Taylor method of order q (space and time).

- FL-CATq: Compact Approximate Taylor method of order q with flux limiter technique. The first order methods considered are Lax-Friedrich for scalar problems and HLL for systems.

- WENOs-CATq: Compact Approximate Taylor method of order q with WENO reconstructions of order s to compute $\tilde{u}_{t,i}^{(1)}$.

- WENOs-RKq: WENO method of order s for the space discretization and TVD-RKq for the time discretization, see [10].

- WENOs-LWAq: Approximate Taylor method of order q with WENO reconstructions of order s to compute $\tilde{u}_{t,i}^{(1)}$, see [14].

5.1. Linear transport equation

We consider first (1) with $a = 1$, in the space interval $[0, 1]$, with initial condition

$$u(x,0) = \begin{cases} 1 & 0 \leq x < 1/2, \\ 2 & 1/2 \leq x < 1, \end{cases} \tag{62}$$

and periodic boundary conditions. A uniform mesh with $N = 80$ points is considered and the LW-CAT method (that, in this case, coincides with the Lax-Wendroff method) is applied for $p = 1, \ldots, 5$.
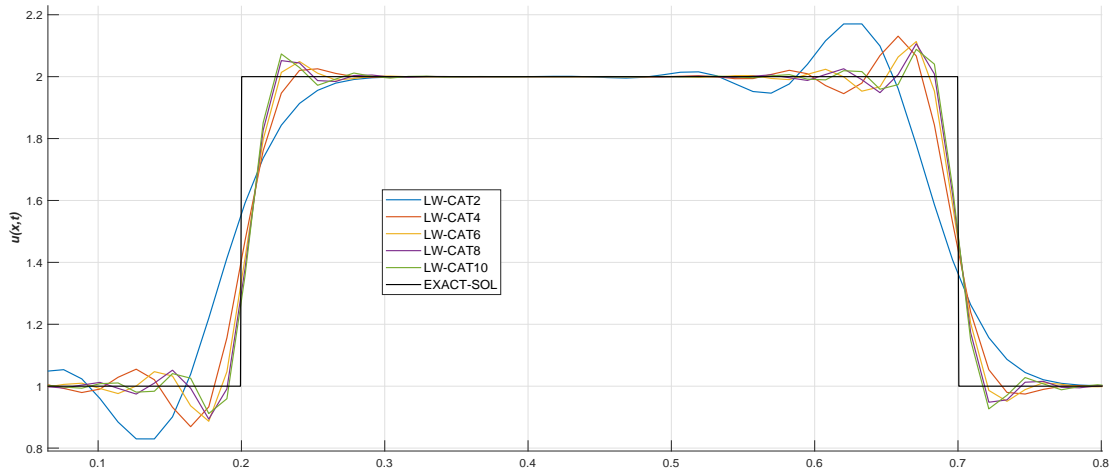

Figure 2: Transport equation with initial condition (62), $CFL = 0.9$ and $t = 1$s: numerical results obtained with LW-CATq, $q = 2, 4, 6, 8, 10$.

Numerical simulations are shown in Figure 2: the L^2 stability of the scheme and the appearance of oscillations near the discontinuities can be observed. Next, we apply to the same problem LW-CAT4, FL-CAT4, WENO5-CAT4, WENO5-RK3, and WENO5-LWA5 methods. A general view is shown in Figure 3 together with a zoom of the area of interest. As it can be observed, the results given by WENO5-CAT4, WENO5-RK3 and WENO5-LWA5 are almost identical. Nevertheless, as it will be seen in the next test problem, WENO5-CAT4 still gives good results for CFL close to one, what is not the case for WENO5-RK3 or WENO5-LWA5.
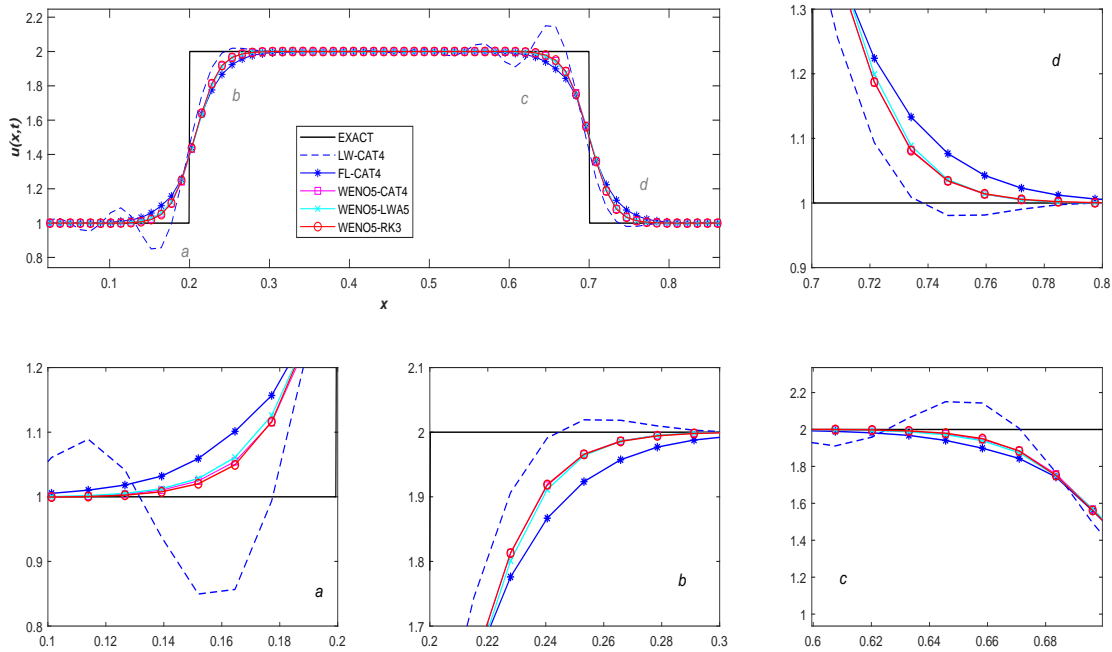

Figure 3: Transport equation with initial condition (62), $CFL = 0.5$ and $t = 1$s. Left-up: general viewt. **a**,**b**,**c** and **d**: enlarged view of interest areas.

Using the algorithm in Section 3.2 CAT methods are easily extended to any $2p$ order, nevertheless, this involves a significant increase of the CPU time simulation and flops (number of operations required), that should be considered, see Table 2.

Order Rate	LW-CAT2	LW-CAT4	LW-CAT6	LW-CAT8	LW-CAT10
time	1	2.98	7.72	18.87	42.66
flops	1	1.61	2.51	3.69	5.16

Table 2: Average rate time and flops to increase from LW-CAT2 to LW-CAT2p, with $p = 2, 3, 4, 5$ for the scalar transport equation with initial conditions (62) and $t = 1$s.

Finally, we consider (1) in the space interval $[0, 2]$ with initial condition,

$$u(x, 0) = 0.25 \sin(\pi x), \tag{63}$$

23

and periodic boundary conditions. Table 3 shows the error and the empirical order for LW-CAT2, LW-CAT4, LW-CAT6, and Table 4 for WENO5-RK3 and WENO5-LWA5 which coincides in all cases with the theoretical one. For smooth solutions WENO-CATq and FL-CATq reduce to the corresponding LW-CATq, so that the accuracy test is not necessary.

	LW-CAT2		LW-CAT4		LW-CAT6	
Δx	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$
0.1053	3.68e-02		1.40e-02		7.88e-03	
0.0526	6.84e-03	2.43	3.50e-05	8.64	4.25e-08	7.50
0.0263	1.70e-03	2.00	2.19e-06	4.00	6.49e-10	6.03
0.0132	4.27e-04	2.00	1.36e-07	4.00	9.89e-12	6.04
0.0066	1.06e-04	2.00	8.55e-09	4.00	1.53e-13	6.01
0.0033	2.66e-05	2.00	5.34e-10	4.00	2.64e-15	5.96

Table 3: Linear transport equation with initial condition (63), $CFL = 0.5$ and $t = 1$s: L^1 errors and accuracy order for LW-CAT2p, $p = 1, 2, 3$.

	WENO5-RK3		WENO-LWA5	
Δx	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$
0.1053	2.03e-03		5.44e-05	
0.0526	6.06e-05	5.06	1.65e-06	5.04
0.0263	1.87e-06	5.02	5.04e-08	5.04
0.0132	5.83e-08	5.00	1.51e-09	5.05
0.0066	1.82e-09	5.00	4.41e-11	5.10
0.0033	5.65e-11	5.01	1.15e-12	5.25

Table 4: Linear transport equation with initial condition (63), $CFL = 0.5$ and $t = 1$s: L^1 errors and accuracy order for WENO5-RK3 and WENO5-LWA5.

5.2. Burgers equation

We consider Burgers equation, i.e. (42) with

$$f(u) = \frac{u^2}{2}.$$

When CAT methods are applied to approximate a discontinuous solution of this nonlinear problem, the oscillations appearing close to the shocks tend to grow and to spoil the numerical solution. Nevertheless, it is still possible to apply these methods by reducing the CFL parameter (the reduction increases with p): for instance, Figure 4 shows the results obtained with CAT-LW2p, $p = 1, 2, 3, 4$ and $CFL = 0.8, 0.4, 0.2, 0.1$, respectively, with initial conditions (62).

Next, the same test problem is solved using LW-CAT4, FL-CAT4, WENO5-RK3 and WENO5-LWA5 methods. Using $CFL = 0.5$ we obtain numerical solutions without spurious oscillations for all the methods. Figure 5 shows a general view of solutions and the van
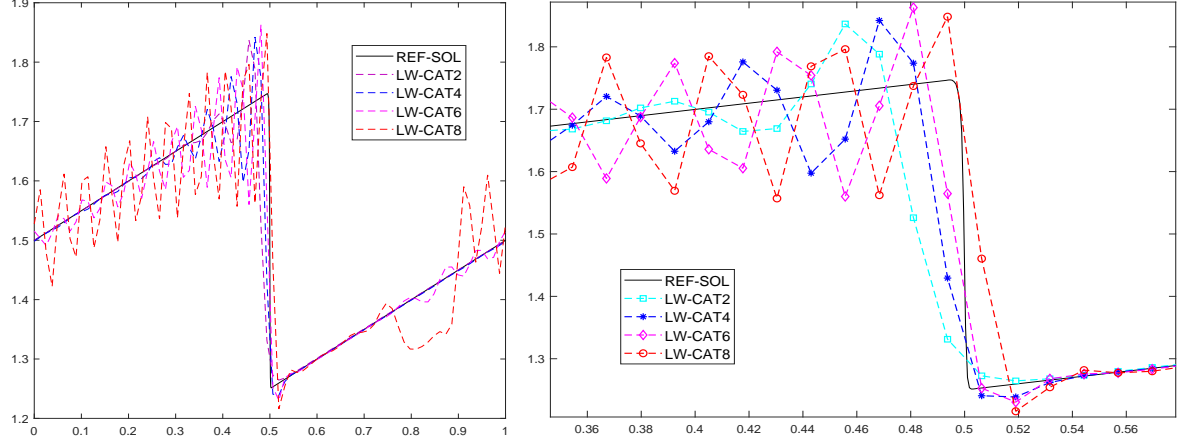
Figure 4: Burgers equation with initial condition (62), $CFL = 0.5$ and $t = 2$s: numerical results obtained with LW-CATq, $q = 2, 4, 6, 8$. Left: general view. Right: enlarged view.

Albada flux limiter function on every inter cell. In Figure 6 the results are compared with those obtained with $CFl = 0.9$. From Figures 5, 6 we can conclude:
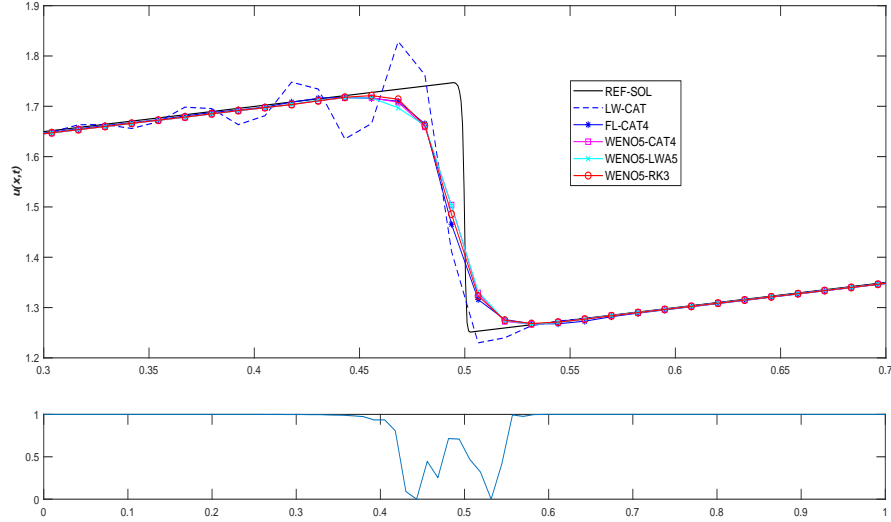

Figure 5: Burgers equation with initial condition (62), $CFL = 0.5$ and $t = 2s$. Up: general view. Down: flux limiter function $\varphi_{i+1/2}$ for FL-CAT4.

- $CFL \leq 0.5$

  - LW-CAT4 show oscillations near the discontinuities, but it is stable.
  - FL-CAT4 is very diffusive near to the discontinuities, due to the selected first order accurate flux limiter function.

25

– WENO5-CAT4, WENO5-LWA4 and WENO5-RK3 show good results, stable and essentially the same values.

- $CFL > 0.5$

    – LW-CAT4: the amplitude of oscillations increases near the discontinuities. However, they remain stable.
    – FL-CAT4: conversely to the previous CFL condition, it shows acceptable solutions near the discontinuities.
    – WENO5-CAT4 ,WENO5-LWA5 and WENO5-RK3 : slight oscillations appear near the discontinuities at the beginning of the simulations. Nevertheless, as the time increases, these oscillations tend to dismiss and the result remains acceptable and stable for WENO5-CAT4 while the solutions given by WENO5-LWA5 is very diffusive and the one given by WENO5-RK3 is overdamped.
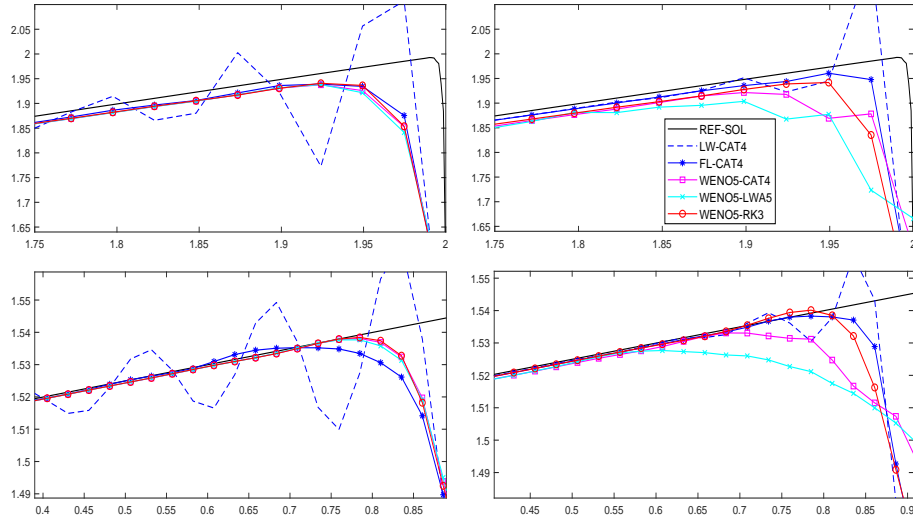

Figure 6: Burgers equation with initial condition (62), $CFL = 0.5$ and $CFL = 0.9$, $t = 2$s and $t = 20$s: zooms of the numerical results. Left-up: $CFL = 0.5$ and $t = 2$s. Left-down: $CFL = 0.5$ and $t = 20$s. Right-up: $CFL = 0.9$ and $t = 2$s. Right-down: $CFL = 0.9$ and $t = 20$s.

Although FL-CAT4 shows better results for bigger CFL, it fails in smooth regions close to critical points and for systems (as it will be seen in Euler 1D equation). For $CFL \leq 1/2$, WENO5-LWA5 is faster than CAT methods (in the computational cost sense): a comparsion is shown in Table 5. However, we can obtain good solutions for CAT methods using $CFL > 0.5$ and obtain similar CPU time computation, see Table 6.

In order to study the order of convergence, we consider again initial condition (63) and periodic boundary conditions. A reference solution at time $t = 0.5$s (when the solution is still smooth) is obtained with WENO5-RK3 using a fine grid of 1400 nodes. The errors and the empirical order are shown in Table 7: the numerical results verify the theoretical analysis.

Nodes	LW-CAT4	FL-CAT4	WENO5-CAT4	WENO5-LWA5
80	0.469	0.4702	0.5508	0.1998
160	0.8090	0.9668	0.9700	0.4616
320	1.9838	1.9908	2.0064	1.0346

Table 5: Burgers equation with initial condition (62), $CFL = 0.5$ and $t = 2$s: computational time in seconds

Nodes	FL-CAT4	WENO5-CAT4
80	0.2003	0.2078
160	0.4024	0.5001
320	1.0494	1.045

Table 6: Burgers equation with initial condition (62), $CFL = 0.9$ and $t = 2$s: computational time in seconds

	LW-CAT2		LW-CAT4		LW-CAT6	
Δx	Error $\\|\cdot\\|_1$	Order $\\|\cdot\\|_1$	Error $\\|\cdot\\|_1$	Order $\\|\cdot\\|_1$	Error $\\|\cdot\\|_1$	Order $\\|\cdot\\|_1$
0.1053	7.94e-03		9.01e-04		2.09e-04	
0.0526	2.08e-03	1.93	6.13e-05	3.88	4.27e-06	5.62
0.0263	5.22e-04	1.99	3.89e-06	3.98	7.49e-08	5.83
0.0132	1.29e-04	2.01	2.44e-07	4.00	1.20e-09	5.96
0.0066	3.08e-05	2.00	1.51e-08	4.00	1.87e-11	6.00
0.0033	6.16e-06	2.00	8.76e-10	4.00	2.84e-13	6.00

Table 7: Burgers equation with initial condition (63), $CFL = 0.5$ and $t = 0.5$s: L^1 errors and accuracy order for LW-CAT2p, $p = 1, 2, 3$.

5.3. 1D Euler Equation

We solve the $1D$ Euler equation for gas dynamics

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0, \tag{64}$$

with

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} \rho u \\ p + \rho u^2 \\ u(E + p) \end{bmatrix}, \tag{65}$$

where ρ is the density, u the velocity, E the total energy per unit volume, and p the pressure. We assume an ideal gas with the equation of state,

$$p(\rho, e) = (\gamma - 1)\rho e, \tag{66}$$

being γ the ratio of specific heat capacities of the gas taken as 1.4 and e is the internal energy per unit mass given by:

$$E = \rho(e + 0.5u^2). \tag{67}$$

We consider the space interval $[0, 2]$ with the initial condition:

$$\begin{aligned}
\rho(x, 0) &= 0.75 + 0.5\sin(\pi x), \\
\rho u(x, 0) &= 0.25 + 0.5\sin(\pi x), \\
E(x, 0) &= 0.75 + 0.5\sin(\pi x),
\end{aligned} \tag{68}$$

and periodic boundary conditions. For this test we take $CFL = 0.5$ and $t = 0.5$s. We use a fine grid with 1400-point mesh to compute LW-CAT8 as a reference solution. The results in Table 8 support the theoretically obtained accuracy.

	LW-CAT2		LW-CAT4		LW-CAT6	
Δx	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$	Error $\|\cdot\|_1$	Order $\|\cdot\|_1$
0.1053	3.34e-03		8.57e-04		5.49e-04	
0.0526	8.82e-03	1.92	9.93e-05	3.11	3.53e-05	4.96
0.0263	2.28e-04	1.95	7.31e-06	3.76	1.01e-06	5.12
0.0132	5.69e-05	2.01	4.81e-07	3.93	1.94e-08	5.71
0.0066	1.35e-05	2.07	3.02e-08	3.99	3.21e-10	5.92
0.0033	2.71e-06	2.30	1.78e-09	4.08	4.99e-12	6.01

Table 8: 1D Euler equations with initial condition (68), $CFL = 0.5$ and $t = 0.5$s: L^1 errors and accuracy order for LW-CAT2p, $p = 1, 2, 3$.

Finally, two tests involving discontinuities are considered:

• The Sod Shock tube problem. The initial condition is given by:

$$(\rho, \rho u, p) = \begin{cases} (1, 0, 1) & \text{if } x < 0, \\ (0.125, 0, 0.1) & \text{if } x > 0. \end{cases}$$

Here, $x \in [-5, 5]$, $CFL = 0.5$, $t = 1$s, and outflow boundary conditions are considered at both sides. For details of this problem see [11]. We compare FL-CAT4, WENO5-CAT4, WENO5-LWA5 and WENO5-RK3 using 450 points. A reference solution is computed with WENO5-RK3 using a 1400-point mesh. The flux limiter function is computed for every variable.
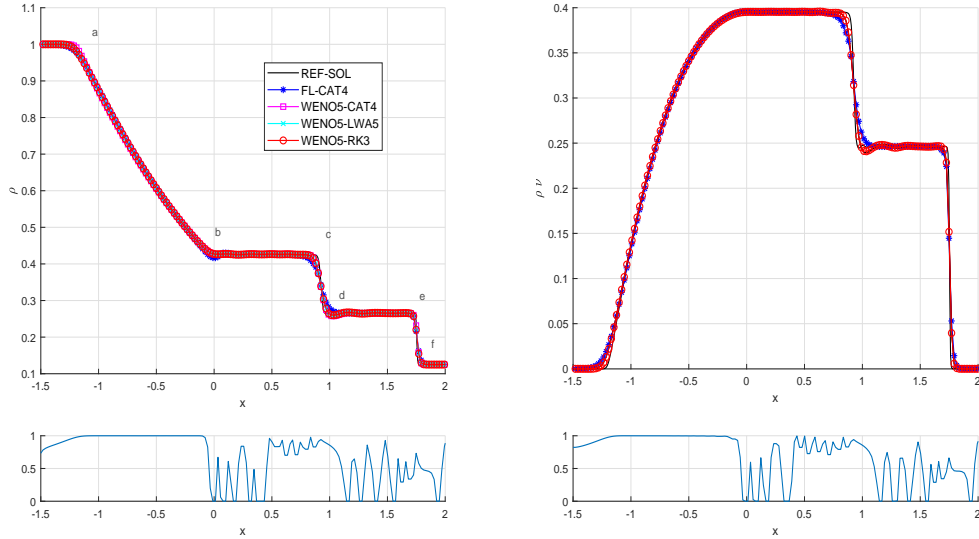

Figure 7: Sod shock tube problem, $CFL = 0.5$ and $t = 0.5$s. Left-up: general view of numerical solutions for ρ. Left-down: flux limiter function for ρ. Right-up: general view of numerical solutions for ρu. Right-down: flux limiter function for ρu.

While all numerical solutions show stable and similar values over smooth regions (see Figure 7), the quality is different in the interest regions (**a,b,c,..,f**): an enlarged view of them can be seen in Figure 8. In the figure we can observe that the solution given by FL-CAT4 is the most diffusive one (due to the condition $CFL = 0.5$) and that WENO5-LWA5 and WENO5-RK3 give essentially the same results. WENO5-CAT4 achieves some improvements, specially in **a**,**c** and **d**.

- The Shu-Osher problem. The initial condition is given by:

$$(\rho, \rho u, p) = \begin{cases} (3.8571, 2.6293, 10.3333) & \text{if } x < -4, \\ (1 + 0.2\sin(5x), 0, 1) & \text{if } x > -4. \end{cases}$$

We consider the space interval $x \in [-5, 5]$, $CFL = 0.5$ and time $t = 1$s. For details see [10] test 8. We compare FL-CAT4, WENO5-CAT4, WENO5-LWA5 and WENO5-RK3 using 450- point mesh and a reference solution computed with WENO5-RK3 using a 1400-point mesh.

Again, all solutions are closely similar and near to the reference solution with exception of FL-CAT4.
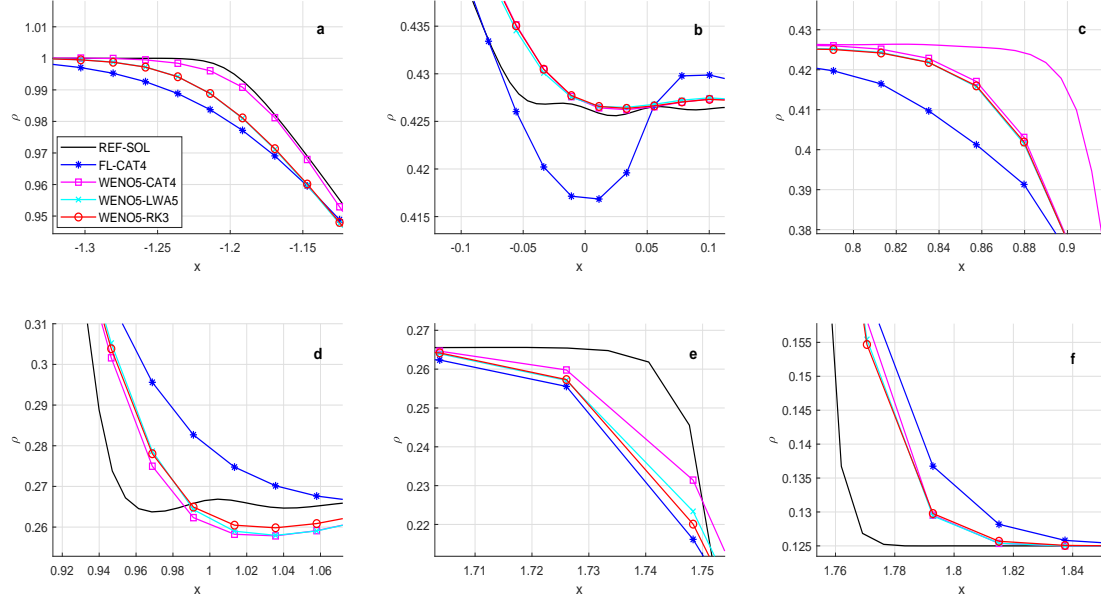
Figure 8: Sod shock tube problem, $CFL = 0.5$ and $t = 0.5$s: zooms of the numerical results for ρ close to regions **a**,**b**,**c**,..,**f**.
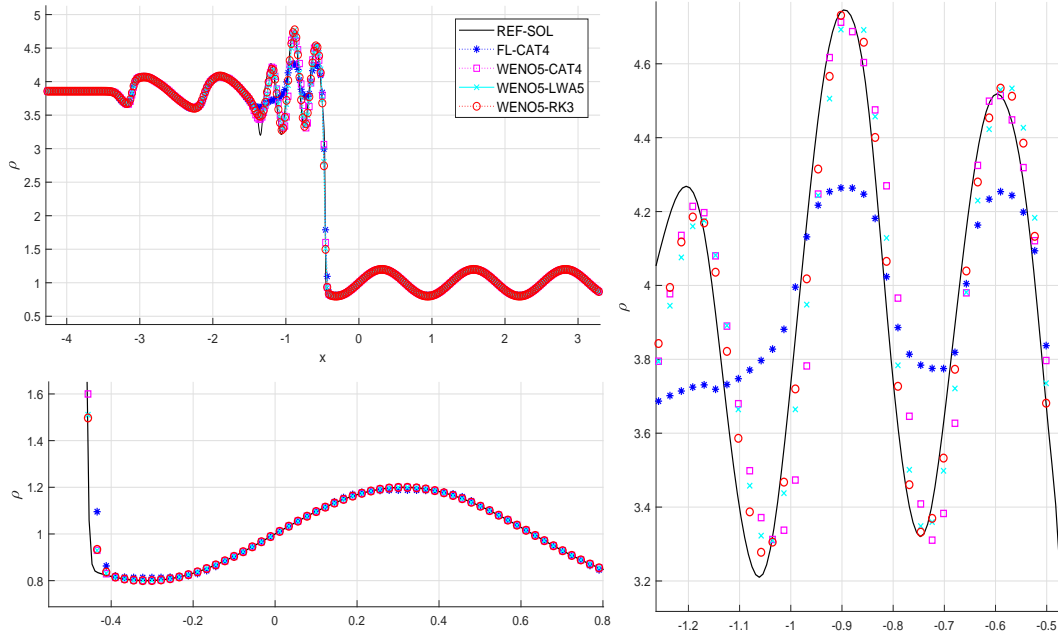

Figure 9: Shu-Osher problem, $CFL = 0.5$ and $t = 1$s: Left-up: general view of numerical solutions for density. Left-down: enlarged view. Right: enlarged view.

6. Conclusions

In this work, first a review of high order Lax-Wendroff methods for the linear transport equation has been presented, including the study of the order and the L^2-stability as well as the computation and properties of the coefficients. Next, an extension to nonlinear conservation laws has been introduced with arbitrary even order $2p$ of accuracy, the so-called Compact Approximate Taylor (CAT) methods. Unlike previous applications of Taylor methods to conservation laws, CAT methods have $2p + 1$-point centered stencils, like Lax-Wendroff methods for linear problems. Moreover, since they inherit the stability properties of Lax-Wendroff methods, they are linearly L^2-stable under a CFL-1 condition. In order to prevent the spurious oscillations that appear close to discontinuities two shock-capturing techniques have been considered: a flux-limiter technique (FL-CAT methods) and WENO reconstruction for the first time derivative (WENO-CAT methods). We follow [14] in the second approach.

These new methods have been compared in a number of test cases with WENO-RK methods (Finite Differences WENO reconstructions in space, TVD-RK in time) and with the WENO-LW methods introduced in [14] (Finite Differences WENO reconstruction for the first time derivative, Approximate Taylor in time). The linear transport equation, Burgers equation, and the 1D compressible Euler system have been considered.For $CFL \leq 0.5$ all the numerical methods work correctly, and the results obtained with all the methods using WENO reconstructions are similar, while the FL-CAT method is more diffusive as expected. Nevertheless, CAT methods are more expensive in computational time and number of operations due to its local character (FL-CAT is less expensive than WENO-CAT as reconstructions are avoided). However, the extra computational cost of CAT methods is compensated by the fact that they still give good solutions with CFL values close to 1.

Future developments include:

- Parallel implementation.

- Use of fast WENO reconstructions: see [1]

- Order adaptive CAT methods based on smooth indicators.

- Application to systems of balance laws.

- Extension to multidimensional problems.

31

References

[1] A. Baeza, R. BÃijrger, P. Mulet, and D. Zorío. On the efficient computation of smoothness indicators for a class of weno reconstructions. *CIMA Pre-PublicaciÃşn 2018-37*, 2018.

[2] B. Fornberg. Classroom note:calculation of weights in finite difference formulas. *SIAM Review*, 40(3):685–691, 1998.

[3] Bengt Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51(184):699–706, 1988.

[4] Friedemann Kemm. A comparative study of tvd-limitersâĂŤwell-known limiters and an introduction of new ones. *International Journal for Numerical Methods in Fluids*, 67(4):404–440, 2010.

[5] Peter Lax and Burton Wendroff. Systems of conservation laws. *Communications on Pure and Applied Mathematics*, 13(2):217–237, 1960.

[6] Randall LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems (Classics in Applied Mathematics Classics in Applied Mathemat)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007.

[7] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

[8] Jiequan Li and Zhicheng Yang. The von neumann analysis and modified equation approach for finite difference schemes. *Applied Mathematics and Computation*, 225:610 – 621, 2013.

[9] Jianxian Qiu and Chi-Wang Shu. Finite difference weno schemes with lax–wendroff-type time discretizations. *SIAM Journal on Scientific Computing*, 24(6):2185–2198, 2003.

[10] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii. *J. Comput. Phys.*, 83(1):32–78, July 1989.

[11] Gary A Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1 – 31, 1978.

[12] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, 2009.

[13] R.F Warming and B.J Hyett. The modified equation approach to the stability and accuracy analysis of finite-difference methods. *Journal of Computational Physics*, 14(2):159 – 179, 1974.

[14] D. Zorío, A. Baeza, and P. Mulet. An approximate lax–wendroff-type procedure for high order accurate schemes for hyperbolic conservation laws. *Journal of Scientific Computing*, 71(1):246–273, Apr 2017.

[15] Gideon Zwas and Saul Abarbanel. Third and fourth order accurate schemes for hyperbolic equations of conservation law form. *Mathematics of Computation*, 25(114):229–236, 1971.