SIAM J. SCI. COMPUT.
Vol. 41, No. 3, pp. B576–B600

SEMI-CONSERVATIVE FINITE VOLUME SCHEMES FOR CONSERVATION LAWS*

ROSA MARIA PIDATELLA†, GABRIELLA PUPPO‡, GIOVANNI RUSSO†, AND PIETRO SANTAGATI§

Abstract. This paper aims to introduce a new class of high order conservative schemes to solve systems of conservation laws. The idea is to couple the conservation form of the system with, possibly simpler, alternative formulations, which can be used to speed up the time update. In this work, we illustrate the procedure for a Runge–Kutta time advancement, but other choices are possible. We show that, as long as the last update is carried out in conservative form, all internal stages can be computed using any consistent nonconservative formulation, still ensuring the propagation of shock waves with the correct speeds. The same procedure can be easily extended to finite difference schemes. Tests from classical and relativistic gas dynamics are carried out to study convergence, numerical robustness and performance.

Key words. high order schemes, hyperbolic systems, method of lines, nonconservative variables, relativistic gas dynamics

AMS subject classifications. 65M08, 65M20, 76M12, 83-08

DOI. 10.1137/18M1177421

1. Introduction.

Several physical systems concerning propagation phenomena are modeled by quasi-linear hyperbolic systems of conservation laws. Such systems have been widely studied, both for the enormous relevance in the applications and for the mathematical challenges they lead to. As known, even if smooth initial conditions are imposed, the solution of a quasi-linear hyperbolic system will in general develop singularities in finite times. After such a time, classical solutions cease to exist, and one has to deal with weak solutions which, for smooth initial data, are composed by piecewise smooth regions separated by jump discontinuities satisfying suitable jump conditions. In general, uniqueness of the weak solution is not guaranteed. It can be restored by adopting some regularization techniques, the most common one being the addition of a parabolic term with a small viscosity which produces a unique solution with sharp gradients that become jump discontinuities in the limit as the viscosity parameter vanishes, yielding the so-called viscosity solution.

The mathematical theory of quasi-linear systems of conservation laws is a very active field of research, and existence and uniqueness of the solution for several classes of systems have been proven [8].

The most common schemes to produce numerical solutions of quasi-linear hyperbolic systems of conservation laws are the so-called shock-capturing schemes:

*Submitted to the journal's Computational Methods in Science and Engineering section March 26, 2018; accepted for publication (in revised form) April 11, 2019; published electronically June 20, 2019.

http://www.siam.org/journals/sisc/41-3/M117742.html

Funding: This work was partially supported by the ITN-ETN Marie-Curie Horizon 2020 program ModCompShock, Modeling and Computation of Shocks and Interfaces, project 642768; by project F.I.R. 2014 Charge Transport in Graphene and Low Dimensional Systems, University of Catania; and by the INDAM-GNCS 2017 research project Numerical Methods for Hyperbolic and Kinetic Equation and Applications.

†Università degli Studi di Catania, Catania, Italy (rosa@dmi.unict.it, russo@dmi.unict.it).

‡La Sapienza Università di Roma, Rome, Italy (gabriella.puppo@uniroma1.it).

§TASS International, Rijswijk, The Netherlands (pietro.santagati@tassinternational.com).

formation and propagation of shocks is automatically "captured" by the scheme, which produces a small region with sharp gradients where the shock forms and propagates.

The construction and analysis of shock-capturing schemes has been a very active field of research in recent decades. Such schemes, based on an Eulerian approach, are designed to discretize the system on a fixed grid, by finite volume, finite difference, or finite element methods.

In this paper we will concentrate on finite volume and finite difference schemes, which, together with discontinuous Galerkin schemes, are the most commonly used methods in this context. An account of finite volume schemes for conservation laws can be found in the book by Le Veque [18], whereas a more mathematical oriented book is the one by Godlewski and Raviart [11].

In finite volume schemes, the conservation laws are integrated in space over each grid cell of the domain, obtaining in such a way evolution equations for the cell averages of variables. The unknowns are now the cell average values, which are modified in each time step by the flux through the edges of each cell, and then the choice of the proper numerical flux functions which correctly approximate the flux is a crucial point of the scheme. This flux can be obtained by the computation of numerical flux functions, for example, Godunov, Engquist–Osher, Rusanov, at the edge of each cell, extracting information on point values from the knowledge of the cell averages. This is obtained through an appropriate nonlinear reconstruction algorithm, such as ENO or WENO [31], or the more recent CWENO [6]. In this way we get, from the original system of PDEs, a large system of ODEs for the cell averages. This procedure is called method of lines, and it yields a semidiscrete system. Once a system of ODEs is derived, suitable integrators, such as strongly stability preserving (SSP) Runge–Kutta schemes can be used [12], providing high order accuracy in time, without any spurious oscillations due to time discretization. A conservative discrete form is mandatory in those regions containing discontinuities, because otherwise their speed propagation might be computed inexactly.

In the above approach, and in most finite volume schemes, the basic unknowns are the conservative variables and the equations are always treated in conservative form. However, in many cases there are more convenient ways to write the system of equations. Harabetian and Pego [14] proposed a hybrid approach, whereby the system is solved by a nonconservative scheme in smooth regions and switches to a conservative form in regions with discontinuities. This approach allowed considerable savings in computational time.

An alternative to the semidiscrete finite volume schemes described before is offered by central schemes on staggered grids. After the first second order shock capturing central scheme on staggered grid in one space dimension by Nessyahu and Tadmor [24], several extensions appeared, increasing the order of accuracy [2, 19], the spatial dimensions [16], or both [21].

In such schemes, a piecewise smooth solution is reconstructed in each cell starting from the cell averages at a given time level t^n. At variance with semidiscrete schemes, in central schemes the fluxes are evaluated at the cell center, along time, enjoying the smoothness of the solution for short times, provided a suitable restriction of CFL type on the time step is satisfied. An advantage that has been attributed to central schemes lies in their construction. They do not require use of exact or approximate Riemann solvers, which are needed for schemes based on the solution of Riemann problems. Such advantage, however, is not the main feature. Actually, the choice of the numerical flux function implies a choice of a particular Riemann solver: a great flexibility of such functions is available, ranging from the Godunov flux, based on the exact solution

of the Riemann problem, to the Rusanov flux (also called local Lax–Friedrichs), which only needs an estimation of the Jacobian's spectral radius of the system. Staggered central schemes do not have this choice and are less effective—for instance, the treatment of contact discontinuities in gas dynamics, which are smeared much more than in the case of sharper Riemann solvers. In practice, the choice of the numerical flux function is actually a *weakness* of staggered central schemes and not an advantage!

There is, however, a great advantage of high order staggered central schemes over classical nonstaggered schemes. Since the fluxes are evaluated from a preliminary computation of the solution at the center of each cell, where the solution is (locally) smooth, a large flexibility is provided in the evaluation of such a preliminary solution. This feature was already pointed out by Nessyahu and Tadmor in their original paper, where they noticed that the so-called predictor value, at cell center and half time step, could be computed by the equation written in conservative form (discretizing the flux) or in nonconservative form (written using the product of the Jacobian matrix times the space derivative of the solution).

That feature was further exploited in [25]. In that paper, a method was presented whereby the numerical solution on a staggered grid is computed by a conservative scheme. In this scheme the stage values, needed for the computation of the fluxes at the Runge–Kutta stages, may be computed by discretizing the equation in nonconservative form. This procedure allows us to gain large flexibility in choosing the dependent variables. However, in spite of this large flexibility, central Runge–Kutta (CRK) methods suffered a lack of flexibility in choosing the numerical flux function, typical of staggered central schemes. Furthermore, the formulation of the boundary conditions might be a little bit more complicated.

In the present paper we propose a new class of schemes, which enjoy the flexibility of CRK in the choice of the possibly nonconservative form of the equation to be discretized in time while, at the same time, permitting the usage of arbitrary numerical flux functions at the cell edges, thus delivering sharp treatment of contacts and linear discontinuities. The stage values are computed at the cell center, where the solution is locally smooth, by writing the system in a not necessarily conservative form. Once the (nonconservative) stage values are computed, a preliminary solution is reconstructed at both cell edges by some suitable nonoscillatory technique. The reconstructed values are used to compute the fluxes at the cell edges by some numerical flux function. Once the fluxes are known, the cell averages are updated by the conservative Runge–Kutta step for the computation of the numerical solution. Therefore, the final scheme is in conservative form, though most calculations can be performed using a convenient nonconservative form of the equation.

The larger flexibility of the new approach allows the construction of more efficient schemes in all those cases in which the system has a simpler form when expressed in nonconservative variables. A typical example is given by the Euler equations of relativistic gas dynamics, in which the computation of the pressure by the conservative variables requires the solution of a nonlinear equation. The new approach allows us to compute pressure just once per time step in each cell, as opposed to s times for a classical s-stage Runge–Kutta scheme applied to a semidiscrete finite volume discretization.

The plan of the paper is the following. In the next section we describe the construction of finite volume schemes based on nonconservative evolution of the fields at the cell center, in one spatial dimension. Then we present a series of numerical tests both for classical gas dynamics and for the relativistic case. The purpose of the

tests is to assess the high resolution capability and the computational efficiency of the new approach. Finally in the last section we draw conclusions and mention future perspectives of the new approach.

2. Semiconservative finite volume schemes. The evolution of conserved quantities, such as mass, momentum, and energy, is given by equations of the form

$$\partial_t \int_V u \, \mathrm{d}v + \int_{\partial V} f(u) \cdot n \, \mathrm{d}S = 0 \qquad \forall V \in \mathbb{R}^d, \tag{2.1}$$

where $u : \mathbb{R}^d \times \mathbb{R}^+ \to \Omega \subset \mathbb{R}^m$ are the conserved quantities, $f = [f_1, \ldots, f_d] : \mathbb{R}^m \to \mathbb{R}^m$ is the flux function, and V is any control volume in $\mathbb{R}^d$. Here Ω denotes the set where the variable u is defined: for instance, the density must be positive. If u is smooth, (2.1) can be rewritten as a system of partial differential equations of the form

$$u_t + \nabla \cdot f = 0. \tag{2.2}$$

It is well known that the solution u can develop singularities in a finite time, even from smooth initial data. In this case, (2.2) must be interpreted in a weak sense, while (2.1) continues to hold; see [8] for more details.

Piecewise smooth solutions of (2.1) are allowed, in which jump discontinuities propagate satisfying the co-called Rankine–Hugoniot conditions, which are derived from (2.1). Since (2.2) descends from the conservative principle (2.1), the equations (2.2) are said to be in *conservative form*. They are the only equations consistent with (2.1) which permit us to derive the correct Rankine–Hugoniot conditions, and thus the correct shock speeds.

Here, for simplicity, we consider initial value problems for one-dimensional, quasi-linear hyperbolic systems of conservation laws of the form

$$\begin{cases} u_t + f_x(u) = 0, & t > 0, \\ u(x,0) \quad\; = u_0(x), & x \in \mathbb{R}. \end{cases} \tag{2.3}$$

Since the system is hyperbolic, the Jacobian $A(u) = \nabla_u f$ is diagonalizable with real eigenvalues. As long as the solution is differentiable, system (2.3) can be rewritten in the nonconservative form

$$u_t + A(u) u_x = 0, \tag{2.4}$$

completed by the same initial conditions. For generic quasi-linear systems, the Jacobian matrix A depends explicitly on the solution u.

The key point of this work is that other nonconservative forms of system (2.3) can be formulated, as long as the solution is smooth, which can be more convenient from a computational point of view, and it is possible to exploit these simpler formulations, without losing exact conservation at the discrete level.

Let v denote a new set of variables, related to u by a one to one smooth mapping $\mathcal{M}(v)$:

$$u = \mathcal{M}(v), \quad J = \frac{\partial \mathcal{M}}{\partial v}, \quad \mathbf{det}(J) \neq 0, \quad \forall\, v \in \mathcal{M}^{-1}(\Omega). \tag{2.5}$$

Rewriting system (2.3) in terms of the new set of variables, we get

$$\begin{cases} v_t + B(v) v_x \;= 0, & t > 0, \\ v(x,0) \quad\; = v_0(x) = \mathcal{M}^{-1}(u_0(x)), & x \in \mathbb{R}, \end{cases} \tag{2.6}$$

where $v : \mathbb{R} \times \mathbb{R}^+ \to \mathcal{M}^{-1}(\Omega) \subset \mathbb{R}^m$, $B = J^{-1} A J$.

We will solve (2.3) with the method of lines. To this end, we cover the computational domain with cells centered on the points $x_j \in \mathbb{R}, j \in \mathbb{Z}$. For simplicity, we consider a uniform grid such that $x_{j+1} - x_j \equiv \Delta x \ \forall j$. Let $I_j = [x_{j-1/2}, x_{j+1/2}]$ be the generic cell, enclosed by the interfaces $x_{j-1/2} = x_j - \frac{\Delta x}{2}$, $x_{j+1/2} = x_j + \frac{\Delta x}{2}$.

Let us introduce the cell averages

$$\bar{u}_j(t) = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t)\, dx, \qquad j \in \mathbb{Z}. \tag{2.7}$$

Integrating system (2.3) over the cells I_j, one obtains the finite volume formulation

$$\begin{aligned} \frac{d\bar{u}_j}{dt} &= -\frac{1}{\Delta x} \left(f(u(x_{j+1/2},t)) - f(u(x_{j-1/2},t)) \right), \\ \bar{u}_j(0) &= \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u_0(x)\, dx, \qquad j \in \mathbb{Z}. \end{aligned} \tag{2.8}$$

The numerical solution of system (2.3) in finite volume form (2.8) is based on three key points

1. a reconstruction algorithm $\mathcal{R}$, which gives an estimate of the numerical solution at the interfaces, starting from the cell averages, with the desired accuracy;
2. a numerical flux function $F_{j+1/2}$, approximating $f(u(x_{j+1/2},t))$ at each cell interface;
3. a time advancing scheme to compute the solution at time $t^n + \Delta t$, starting from time t^n.

The purpose of the reconstruction algorithm $\mathcal{R}$ is to obtain estimates of the point values of the solution at the cell interfaces, starting from the cell averages. Typically, one works with piecewise polynomial reconstructions,

$$\mathcal{R}(x, \bar{\mathbf{u}}) = \sum_j P_j(x) \chi_{I_j}(x),$$

where $P_j(x)$ are polynomials of degree d, which match $d+1$ contiguous cell averages, including $\bar{u}_j$, and where $\bar{\mathbf{u}}$ denotes the vector containing all cell averages of the numerical solution. Note that the reconstruction is discontinuous across cells. In particular, let

$$u^+_{j+1/2} = P_{j+1}(x_{j+1/2}) \qquad u^-_{j+1/2} = P_j(x_{j+1/2})$$

be the *boundary extrapolated data* at the cell interfaces. If the solution is smooth, and the data are reconstructed with accuracy p, then the jump at the interface $u^+_{j+1/2} - u^-_{j+1/2} = O(\Delta x)^p$.

The numerical flux $F_{j+1/2}$ is a function of the two estimates of the solution at the interface, namely $F_{j+1/2} = F(u^+_{j+1/2}, u^-_{j+1/2})$, and it is a numerical approximation of the flux $f(u)$ at the cell interface, obtained solving numerically (or exactly) the Riemann problem at the interface defined by the data $u^+_{j+1/2}$ and $u^-_{j+1/2}$. Many popular numerical fluxes can be written in viscous form as

$$F(u^+, u^-) = \frac{1}{2} \left(f(u^+) + f(u^-) \right) - \frac{1}{2} Q(u^+, u^-) \left(u^+ - u^- \right), \tag{2.9}$$

where $Q(u^+, u^-)$ is the viscosity matrix. For instance, for local Lax–Friedrichs (also called Rusanov flux) one has

$$Q(u^+, u^-) = \alpha(u^+, u^-) \mathbb{I},$$

where, if f is convex, $\alpha = \max(\rho(A(u^+), \rho(A(u^-))) = \max(\rho(B(v^+), \rho(B(v^-)))$. Other choices lead to less dissipative numerical monotone fluxes.[1] For example, if a Roe matrix $A(u^+, u^-)$ [28] is available one could use

$$Q(u^+, u^-) = |A(u^+, u^-)|$$

or some approximation of the absolute value of the matrix which is cheaper to compute and does not require full characteristic decomposition of the matrix. Approaches based on the approximation of the matrix absolute value were introduced in [9] and then generalized in various contexts (see, for example, the recent paper [5]).

The reconstruction $\mathcal{R}$ and the numerical flux F provide the space discretization of the scheme. With these ingredients, the semidiscrete form of system (2.8) is given by the system of ODEs

$$\frac{d\bar{u}_j}{dt} = -\frac{1}{\Delta x}\left[F(u^+_{j+1/2}(t), u^-_{j+1/2}(t)) - F(u^+_{j-1/2}(t), u^-_{j-1/2}(t))\right], \qquad j \in \mathbb{Z}. \tag{2.10}$$

Any numerical method for the integration of systems of ODEs can be used as a time advancing scheme to solve (2.10). In this work, we will use explicit Runge–Kutta methods [13]. For a generic initial value problem of the form

$$\begin{aligned} \frac{dy}{dt} &= g(t, y(t)), \quad y(t) : \mathbb{R} \to \mathbb{R}^d, \quad d \in \mathbb{N}, \\ y(t_0) &= y_0 \end{aligned}$$

an explicit ν-stage Runge–Kutta scheme can be written as

$$y^{n+1} = y^n + \Delta t \sum_{l=1}^{\nu} b_l \, g(t^n + c_l \Delta t, Y^{(l)}) \quad \text{(corrector step)}, \tag{2.11}$$

$$Y^{(l)} = y^n + \Delta t \sum_{k=1}^{l-1} a_{lk} \, g(t^n + c_k \Delta t, Y^{(k)}) \quad \text{(predictor step)}, \tag{2.12}$$

$$l = 1, \ldots, \nu,$$

where $\{Y^{(l)}\}_{l=1,\ldots,\nu}$ are the internal stages of the Runge–Kutta step (also known as *stage values*).

The coefficients $\{c_l\}_{l=1,\ldots,\nu}$, $\{b_l\}_{l=1,\ldots,\nu}$, $\{a_{ij}\}_{i,j=1,\ldots,\nu}$ univocally identify the numerical scheme. In standard finite volume schemes with Runge–Kutta time advancement, the evolution of the numerical solution (2.11) is obtained applying the Runge–Kutta scheme to the semidiscrete form (2.10):

$$\begin{aligned} \bar{u}_j^{n+1} &= \bar{u}_j^n - \frac{\Delta t}{\Delta x} \sum_{l=1}^{\nu} b_l \Delta F_j^{(l)}, \\ \bar{u}_j^{(l)} &= \bar{u}_j^n - \frac{\Delta t}{\Delta x} \sum_{k=1}^{l-1} a_{lk} \, \Delta F_j^{(k)}, \quad l = 1, \ldots, \nu, \\ \Delta F_j^{(l)} &= F\left(u^{(l)+}_{j+1/2}, u^{(l)-}_{j+1/2}\right) - F\left(u^{(l)+}_{j-1/2}, u^{(l)-}_{j-1/2}\right), \end{aligned} \tag{2.13}$$

where the values $u^{(l)\pm}_{j+1/2}$ at each cell interface are computed with a reconstruction step from the cell averages $\bar{u}^{(l)}$. Thus, the conservative form of the equation is used for the

[1] A numerical flux $F(u^+, u^-)$ is said to be monotone if the first order scheme produced by the flux is a monotone scheme, i.e., if $u_j^n \geq w_j^n \; \forall j$, then $u_j^{n+1} \geq w_j^{n+1} \; \forall j$. For more details consult [17].

final update (corrector step) and also for each of the ν stage values. This is precisely the point where our new SC (semiconservative) schemes differ from traditional finite volume FC (fully conservative) schemes.

In the semiconservative SC approach we propose, we first seek an alternative simple formulation of the equations in the form (2.6), for a new set of variables v, defined by the smooth one to one mapping (2.5). We then use the conservative form of the equation for the final update, but each of the stage values is computed using the simpler system $v_t + B(v)v_x = 0$. Clearly, the convenience of the method depends on how much simpler system (2.6) is with respect to the conservative formulation and on the number of stages ν. Thus, this approach is particularly interesting for high order schemes.

More precisely, from the initial cell averages $\bar{u}^n$, we apply a reconstruction step which yields the point values u_j^n at the cell centers. From these, we compute $v_j^n = \mathcal{M}^{-1}(u_j^n)\ \forall j$. Next, the stages are computed from (2.6) as

$$v_j^{(l)} = v_j^n - \Delta t \sum_{k=1}^{l-1} a_{lk} B\big(v_j^{(k)}\big)\big(D_x v^{(k)}\big)_j, \quad l = 1, \dots, \nu. \tag{2.14}$$

Here $(D_x v^{(k)})_j$ denotes the numerical discretization of the space derivative of the data $v^{(k)}$, obtained with a suitable reconstruction. After all stages have been computed using the simple system (2.14), the boundary extrapolated data at the lth stage are obtained with a reconstruction step on the point values $v_j^{(l)}$, and the conservative variables u are recovered from $u_{j+1/2}^{(l)+} = \mathcal{M}(v_{j+1/2}^{(l)+})$ and $u_{j+1/2}^{(l)-} = \mathcal{M}(v_{j+1/2}^{(l)-})$. These quantities are used to close the time step with (2.13).

The algorithm is illustrated in Figure 2.1 and in the box appearing in Figure 2.2. Note that different reconstructions are needed: $\mathcal{R}_a$ is the reconstruction which

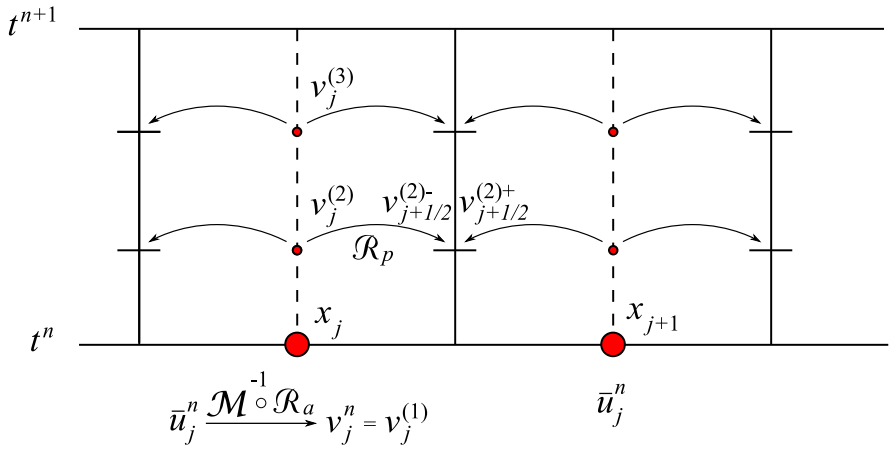

FIG. 2.1. *Diagram for semiconservative finite volume schemes. At time t^n the cell averages are known. Pointwise values u_j^n are computed at cell centers and then converted to the non-conservative variables v_j^n. Such variables are evolved in time and the corresponding stage values $v_j^{(k)}$ are computed at position x_j and time $t^n + c_k \Delta t$. The stage values are reconstructed at each cell edge, obtaining $v_{j\pm\frac{1}{2}}^{(k)}$. From such values the numerical flux at the quadrature nodes in time is computed and adopted in order to obtain the conservative approximation of cell average at the new time t^{n+1}.*

Algorithm: SC schemes

- From $\bar{\mathbf{u}}^n$ find the pointvalues $u_j^n = \mathcal{R}_a(x_j, \bar{\mathbf{u}}^n)\ \forall j$.
- Compute $v_j^n = \mathcal{M}^{-1}(u_j^n)\ \forall j$.
- Compute all stage values $v_j^{(l)}$ from (2.14) $\forall j$.
- Compute the values at interfaces $v_{j+1/2}^{(l)\pm} = \mathcal{R}_p(x_{j+1/2}, \mathbf{v}^{(l)})\ \forall j$.
- Go back to conservative variables $u_{j+1/2}^{(l)\pm} = \mathcal{M}(v_{j+1/2}^{(l)\pm})\ \forall j$.
- Close the time step on conservative variables with (2.13).

FIG. 2.2. *Algorithm for the semiconservative finite volume schemes.*

computes point values from cell averages, while $\mathcal{R}_p$ yields boundary extrapolated data from point values, and D_x is a discrete derivative.

2.1. Constructing high order SC schemes. Semiconservative schemes can be constructed with any order of accuracy. To obtain a scheme of order p, the reconstructions and the time advancement Runge–Kutta scheme must be of matching order. Here we will consider second, third, and fourth order schemes.

Second order. Since, for a smooth function $w(x)$, $w(x_j) = \overline{w}_j + O(\Delta x)^2$, at second order accuracy the reconstruction $\mathcal{R}_a$ is the identity. From the point values $\mathbf{w}$, the approximate slopes σ_j are computed with a piecewise linear reconstruction and a limiter such as MinMod; see [17] and references therein. Thus $\forall j$ we have

$$\begin{aligned} u_j^n &= \bar{u}_j^n, \\ v_j^n &= \mathcal{M}(u_j^n), \\ \sigma_j &= D_x v|_j, \\ v_{j+1/2}^+ = v_{j+1}^n - \tfrac{1}{2}\Delta x\, \sigma_{j+1}, &\qquad v_{j+1/2}^- = v_j^n + \tfrac{1}{2}\Delta x\, \sigma_j, \end{aligned}$$

where the slopes σ are computed from the data $\mathbf{v}$. Then the scheme proceeds as in Figure 2.2. Note that the same reconstruction is used to compute the point values, the discrete derivative $\sigma = D_x v$ in (2.14), and the boundary extrapolated data.

The SSP second order Heun scheme is used for the time advancement.

Fourth order. All reconstructions are obtained with WENO type interpolations; see [32], [3], and the review [31]. For a fourth order scheme, the reconstruction is based on three parabolas, combined in order to maximize accuracy on smooth solutions but, at the same time, preventing spurious oscillations on nonsmooth data. The generic WENO reconstruction can be written as

$$\mathcal{R}(x, \bar{\mathbf{u}}) = \sum_{\ell=-1}^{1} \omega_j^\ell\, P_{j+\ell}(x).$$

When reconstructing point values from cell averages ($\mathcal{R}_a$), the parabolas $P_{j+\ell}$ interpolate the data $\bar{u}_{j+\ell}$ in the sense of averages,

$$\tfrac{1}{\Delta x}\int_{I_{j+\ell}} P_j(x) = \bar{u}_{j+\ell}, \qquad \ell = -1, 0, 1,$$

while the reconstruction from point values $\mathcal{R}_p$ has $P_j(x_{j+l}) = u_{j+l}, l = -1, 0, 1$. In both cases, the generic parabola $P_j(x) = P(x; w_{j-1}, w_j, w_{j+1})$ is given by

$$P_j(x) = w_j - \frac{C}{24}(w_{j+1} - 2w_j + w_{j-1}) + \frac{w_{j+1} - w_{j-1}}{2\Delta x}(x - x_j) + \frac{w_{j+1} - 2w_j + w_{j-1}}{2\Delta x^2}(x - x_j)^2 \tag{2.15}$$

with $C = 1$, $w_\ell = \bar{u}_\ell, \ell = j-1, j, j+1$ for $\mathcal{R}_a$ and $C = 0$, $w_\ell = u_\ell, \ell = j-1, j, j+1$ for $\mathcal{R}_p$. The nonlinear weights $\{\omega_j^\ell\}$ are

$$\omega_j^\ell = \frac{\alpha_j^\ell}{\sum_{k=-1}^{1} \alpha_j^k}, \quad \alpha_j^\ell = \frac{d^\ell}{(\varepsilon + \beta_j^\ell)^2}, \qquad \ell = -1, 0, 1. \tag{2.16}$$

The smoothness indicators β_j^ℓ prevent the selection of stencils with nonsmooth data, thus controlling spurious oscillations. In the case of parabolas they are given by (see [31])

$$\begin{aligned}
\beta_j^{-1} &= \frac{13}{12}(\bar{u}_{j-2} - 2\bar{u}_{j-1} + \bar{u}_j)^2 + \frac{1}{4}(\bar{u}_{j-2} - 4\bar{u}_{j-1} + 3\bar{u}_j)^2, \\
\beta_j^{0} &= \frac{13}{12}(\bar{u}_{j-1} - 2\bar{u}_j + \bar{u}_{j+1})^2 + \frac{1}{4}(\bar{u}_{j-1} - \bar{u}_{j+1})^2, \\
\beta_j^{1} &= \frac{13}{12}(\bar{u}_j - 2\bar{u}_{j+1} + \bar{u}_{j+2})^2 + \frac{1}{4}(3\bar{u}_j - 4\bar{u}_{j+1} + \bar{u}_{j+2})^2.
\end{aligned}$$

The parameter ε prevents division by zero, but it is also involved in the accuracy of the scheme; see [1] or [7]. Here we choose simply $\varepsilon = 10^{-6}$, as in [31].

A key point is the choice of the constants d_ℓ. When the data derive from a smooth function, all smoothness indicators are approximately equal, and the weights $\omega_j^\ell \simeq d^\ell$. Then the constants d_ℓ are determined maximizing the accuracy that can be obtained with a convex combination of the three parabolas involved. The problem is that a convex combination of three parabolas can provide uniform accuracy within the cell only up to third order, even though the stencil contains five cells. To increase accuracy, the constants are determined maximizing the accuracy of the reconstruction at one particular point. Note that each quantity being reconstructed needs a specific set of constants and thus a different reconstruction.

A fourth order reconstruction of point values from cell centers can be obtained by any symmetric choice of the constants $d_\ell, \ell = -1, 0, 1$, as illustrated in [19]. We use $d^{-1} = 3/16$, $d^0 = 5/8$, $d^1 = 3/16$. Higher order accuracy is possible (indeed sixth order can be obtained); however, it requires the use of negative weights. This problem can be tackled with the technique described in [30], but we will not consider this case here. For reconstructing the boundary extrapolated data, the constants are $d^{-1} = 5/16$, $d^0 = 5/8$, $d^1 = 1/16$ for the left value $v^+_{j-1/2}$ and $d^{-1} = 1/16$, $d^0 = 5/8$, $d^1 = 5/16$ for the right value $v^-_{j+1/2}$. The accuracy of the reconstructed data is 5, for smooth functions.

Finally, a reconstruction is needed also to compute the numerical derivative D_x. Now, the reconstruction is given by

$$D_x v|_{x_j} = \mathcal{R}_D(x_j, \mathbf{v}) = \sum_{\ell=-1}^{1} \omega_j^\ell \frac{\mathrm{d}}{\mathrm{d}x} P_{j+\ell}(x_j).$$

The accuracy constants in this case are $d^{-1} = 1/6$, $d^0 = 2/3$, $d^1 = 1/6$, and the accuracy of $D_x v|_{x_j}$ is 4.

A class of WENO type reconstructions with uniform accuracy within the whole cell can be found in [6]: in this case a single reconstruction step can yield all needed quantities. We will illustrate this technique for constructing a third order scheme in the next paragraph.

The time advancement scheme is the standard fourth order Runge–Kutta scheme. In all cases, the numerical flux used is the Lax–Friedrichs flux.

Third order. The reconstruction used here is taken from [20] and can be viewed as a particular case of [6], leading to a third order scheme.

Consider a set of data (point values or cell averages) and a polynomial P_{opt} of degree G, which interpolates in some sense all the given data (*optimal polynomial*). The CWENO operator computes a reconstruction polynomial

$$P_{\mathsf{rec}} = \mathsf{CWENO}(P_{\mathsf{opt}}, P_1, \ldots, P_m) \in \mathbb{P}^G$$

using $P_{\mathsf{opt}} \in \mathbb{P}^G$ and a set of lower order alternative polynomials $P_1, \ldots, P_m \in \mathbb{P}^g$, where $g < G$ and $m \geq 1$. The definition of P_{rec} depends on the choice of a set of positive real coefficients $d_0, \ldots, d_m \in [0, 1]$ such that $\sum_{\ell=0}^{m} d_\ell = 1$, $d_0 \neq 0$ (called *linear coefficients*) as follows:

1. first, introduce the polynomial P_0 defined as

$$P_0(x) = \frac{1}{d_0}\left(P_{\mathsf{opt}}(x) - \sum_{\ell=1}^{m} d_\ell P_\ell(x)\right) \in \mathbb{P}^G; \tag{2.17}$$

2. then the nonlinear coefficients ω_ℓ are computed from the linear ones as in (2.16), where β_ℓ denotes suitable regularity indicators, which can be chosen following [15] as

$$\beta_\ell = \sum_{k\geq 1} \Delta x^{2k-1} \int_{x_j-1/2}^{x_j+1/2} \left(\frac{\mathrm{d}^l}{\mathrm{d}x^l} P_\ell(x)\right)^2 \Delta x, \quad \ell = 0, \ldots, m; \tag{2.18}$$

3. and finally

$$P_{\mathsf{rec}}(x) = \sum_{\ell=0}^{m} \omega_\ell P_\ell(x) \in \mathbb{P}^G. \tag{2.19}$$

In the case of a third order scheme, the degree of P_{opt} and P_0 is $G = 2$, while the $m = 2$ lower degree polynomials are just linear functions. The interesting point is that since P_{rec} is defined everywhere in the cell one can use it to compute the extrapolated data and the discrete derivative. The constants d^ℓ can be chosen quite freely. Here we have $d_0 = \frac{1}{2}, d_1 = d_2 = \frac{1}{4}$.

As time integrator, we employ the third order Runge–Kutta scheme used in [15].

3. SC schemes and the Lax–Wendroff theorem. A crucial issue in the integration of systems of conservation laws is the enforcement of exact conservation. If shock waves appear, exact conservation ensures that the correct wave speeds are captured also at the numerical level. This result is guaranteed by the Lax–Wendroff theorem, which contains sufficient conditions for the convergence of a numerical scheme to a weak solution of conservation laws.

The key fact is that the Lax–Wendroff theorem (see, for instance, [10, p. 100]) requires the scheme to be *conservative*, and this is the main reason why one discretizes directly the conservative form of the equations, thus working in conservative

variables. However, recalling the definition of conservative scheme, we can easily prove that SC schemes are indeed conservative and therefore satisfy the hypotheses of the Lax–Wendroff theorem.

DEFINITION 3.1 (conservative scheme). *The numerical scheme*

$$\bar{u}_j^{n+1} = \bar{u}_j^n - \frac{\Delta t}{\Delta x}\left(F_{j+1/2} - F_{j-1/2}\right)$$

is conservative if the numerical flux $F_{j+1/2} = F(\bar{u}_{j-p}^n, \ldots, \bar{u}_{j+m}^n)$ *(*p *and* m *positive integers) satisfies the following conditions:*

1. $F(u, \ldots, u) = f(u)$ *(consistency),*
2. $F(\bar{u}_{j-p}, \ldots, \bar{u}_{j+m})$ *is at least Lipschitz continuous in all of its arguments.*

Consistency. First note that the scheme is clearly built on a stencil with a finite number of cells. Let then $x_{j-p}, \ldots, x_{j+m}$ be the cell centers in the stencil containing the data needed to compute the numerical flux at the interface $x_{j+1/2}$, with p and m positive integers.

If $\bar{u}_{j-p}^n = \cdots = \bar{u}_{j+m}^n = U$, then, since any piecewise polynomial reconstruction interpolates constants exactly, also the reconstructed point values satisfy $u_{j-p}^n = \cdots = u_{j+m}^n = U$. Then the transformed variables are $v_{j-p}^n = \cdots = v_{j+m}^n = V = \mathcal{M}^{-1}(U)$. Again, the piecewise polynomial reconstruction preserves constants, thus the numerical derivative is zero, and all stage values in (2.14) reduce to $v_k^{(l)} = v_k^n = V\ \forall k$ in the stencil of the cell j. Reconstructing these data, all boundary extrapolated data result in $v_{j+1/2}^{(l),\pm} = V$. Mapping back to conservative variables, $u_{j+1/2}^{(l),\pm} = \mathcal{M}(V) = U$. Since we are using a conservative and consistent numerical flux, $F_{j+1/2}^{(l)} = F(u_{j+1/2}^{(l),+}, u_{j+1/2}^{(l),-}) = F(U,U) = f(U)$. Finally, the numerical flux of the scheme is $F_{j+1/2} = \sum_l b_l F_{j+1/2}^{(l)} = f(U)\sum_l b_l$. So, the consistency of the numerical flux relies ultimately on the consistency of the Runge–Kutta scheme, which ensures that $\sum b_l = 1$.

Lipschitz regularity. All ingredients used in the construction of the numerical fluxes are at least Lipschitz continuous. More precisely, for the second order scheme, the piecewise linear reconstruction using MinMod has just Lipschitz regularity, while WENO reconstructions are C^∞. The Lax–Friedrichs numerical flux is also C^∞. The final numerical flux is just a composition of these functions, and thus it has the required smoothness.

4. Applications and numerical results. We illustrate the performance and the field of applicability of the scheme with examples and numerical tests. We start from scalar conservation laws, where it is easy to appreciate the differences between standard conservative finite volume schemes and the new semiconservative schemes. Next we continue with classical Euler equations, to end with the equations of relativistic gas dynamics, where the new scheme permits us to obtain considerable savings in computational complexity.

4.1. Burgers' equation. The computation of the correct shock speeds is ensured by the Lax–Wendroff theorem, which uses only the *consistency* of the numerical fluxes, appearing in the conservative form of the finite volume formulation.

As an example, consider the following two initial value problems:

$$\partial_t u + \partial_x\left(\frac{1}{2}u^2\right) = 0, \qquad u(x, t=0) = u_0(x) > 0, \tag{4.1}$$

$$\partial_t z + \partial_x\left(\frac{1}{3}\sqrt{(2z)^3}\right) = 0, \qquad z(x, t=0) = \tfrac{1}{2}u_0^2(x). \tag{4.2}$$

If, in the second equation, we take the change of variables $z = \mathcal{M}(v) = \frac{1}{2}v^2$, we find that in the v variables, (4.2) coincides with the characteristic form of (4.1), namely $v_t + vv_x = 0$, with the same initial data. Thus the two equations have the same solution, as long as the solution is smooth. However, an initial step $u_0(x) = u_L + (u_R - u_L)H(x)$, where H is the Heavyside function, yields two different shock speeds in the two initial value problems, namely

$$s_u = \frac{1}{2}(u_L + u_R),$$
$$s_z = \frac{2}{3}\frac{u_L^2 + u_L u_R + u_R^2}{u_L + u_R}.$$

In fact, (4.1) prescribes the conservation of the quantity u, while the second equation prescribes the conservation of the quantity z, and this fact yields two different results for the shock speed, when one applies the Rankine–Hugoniot condition.

In the standard fully conservative scheme, the final update and all stage values are computed directly from the two conservation laws. In the semiconservative approach, for (4.1) we choose the auxiliary variables $v = \mathcal{M}^{-1}(u) = \mathbb{I}(u)$. Then the algorithm is the following (here $\lambda = \frac{\Delta t}{\Delta x}$):

- Reconstruct the point values U_j^n from cell averages, and set $V_j^n = U_j^n$.
- Compute the stage values using the characteristic form $v_t + vv_x = 0$,

$$V_j^{(l)} = V_j^n - \Delta t \sum_{k=1}^{l-1} V_j^{(k)} D_x(V^{(k)})(x_j), \qquad l = 1, \dots, \nu.$$

- Use the point values of the stages to reconstruct the boundary extrapolated data, $(V_{j+1/2}^{(l)})^{\pm}$, and obtain $(U_{j+1/2}^{(l)})^{\pm} = (V_{j+1/2}^{(l)})^{\pm}$.
- Apply the conservative corrector step, evaluating the numerical flux $F^{(l)} = F(U_{j+1/2}^{(l)+}, U_{j+1/2}^{(l)-})$, consistent with $f(u) = \frac{1}{2}u^2$, obtaining the new cell averages

$$\bar{U}_j^{n+1} = \bar{U}_j^n - \lambda \sum_{l=1}^{\nu} b_i \left(F_{j+1/2}^{(l)} - F_{j-1/2}^{(l)} \right).$$

For (4.2), we choose $v = \mathcal{M}^{-1}(z) = \sqrt{2z}$. Then, the semiconservative SC approach results in the following algorithm:

- Reconstruct the point values Z_j^n from cell averages. Set $V_j^n = \sqrt{2Z_j^n}$.
- Compute the stage values using the characteristic form $v_t + vv_x = 0$

$$V_j^{(l)} = V_j^n - \Delta t \sum_{k=1}^{l-1} V_j^{(k)} D_x(V^{(k)})(x_j), \qquad l = 1, \dots, \nu.$$

- Use the point values of the stages to reconstruct the boundary extrapolated data, $(V_{j+1/2}^{(i)})^{\pm}$, and obtain $(Z_{j+1/2}^{(l)})^{\pm} = \frac{1}{2}[(V_{j+1/2}^{(l)})^{\pm}]^2$.
- Apply the conservative corrector step, evaluating the numerical flux $F^{(l)} = F(Z_{j+1/2}^{(l)+}, Z_{j+1/2}^{(l)-})$, consistent with $f(z) = \frac{1}{3}(2z)^{(3/2)}$, obtaining the new cell averages

$$\bar{Z}_j^{n+1} = \bar{Z}_j^n - \lambda \sum_{l=1}^{\nu} b_i \left(F_{j+1/2}^{(l)} - F_{j-1/2}^{(l)} \right).$$

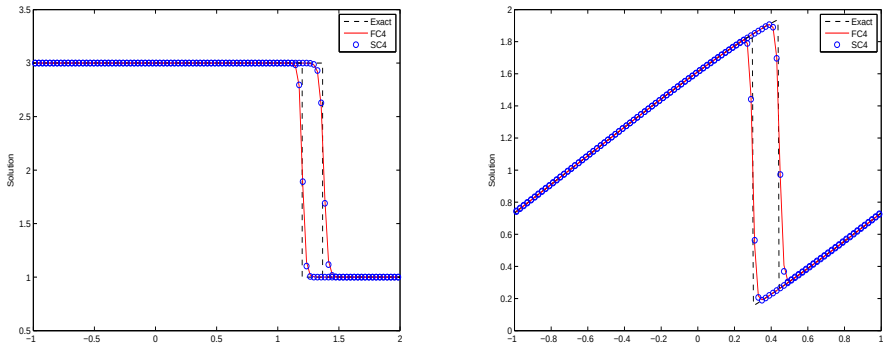

FIG. 4.1. *Shock propagation (left) and shock formation (right). Red continuous line: fully conservative fourth order (FC4) scheme; blue circles: semiconservative fourth order (SC4) scheme.*

The results are shown in Figure 4.1. The plot on the left is obtained with an initial jump, located in $x = -0.8$ with $u_L = 3$ and $u_R = 1$, at time $T = 1$. The Burgers' solution is a shock traveling with speed $s_1 = 2$; the modified Burgers' (4.2) solution is a shock with speed $s_2 = \frac{13}{6}$. The plot contains the solution of both problems obtained with the fully conservative fourth order scheme (FC4) and the semiconservative fourth order scheme (SC4). The plot on the right has as initial data $u_0(x) = \sin(\pi(x-\frac{1}{2}))+1$. For both equations the shock appears at the same time, but it will have different speeds. Note that the FC and SC solutions coincide in all cases, with the correct shock speeds. All numerical solutions were obtained with $N = 100$ grid points, and a CFL number CFL $= 0.9$.

4.2. Accuracy. We carry out accuracy tests on linear advection, using low and high frequency solutions, for schemes of order 2, 3 and 4. The equation is $u_t + u_x = 0$. The low frequency initial datum is

$$u_0(x) = \sin\left(\pi x - \sin(\pi x)/\pi\right),$$

while for high frequency, we consider

$$u_0(x) = \sin(\pi x) + \tfrac{1}{4}\sin(15\pi x)\, e^{-20x^2}.$$

The first test can be found in [1], while the second test is due to [29].

Figure 4.2 contains the convergence history for the low frequency (left panel) and the high frequency (right panel) test. The final time is $T = 2$, with periodic boundary conditions on $[-1, 1]$, so that each solution completes a whole period. The black dashed lines are the expected rates (2, 3, and 4), the green, red, and blue curves refer to the second, third, and fourth order scheme, respectively. The results of the fully conservative schemes are labeled with circles, while the results of the new SC schemes are marked with plus signs. The SC schemes have slightly smaller errors than the traditional FC schemes, except than in the case of the fourth order scheme. This is due to the fact that the WENO reconstruction is fifth order on the boundary extrapolated data (which are the only data needed by the fourth order FC4) but only fourth order on the reconstruction of point values at the cell center, which is needed by SC4.

For the data on the high frequency test, we note that the expected accuracy is obtained only after a transient, when the grid is fine enough to detect the high frequency features of the solution.

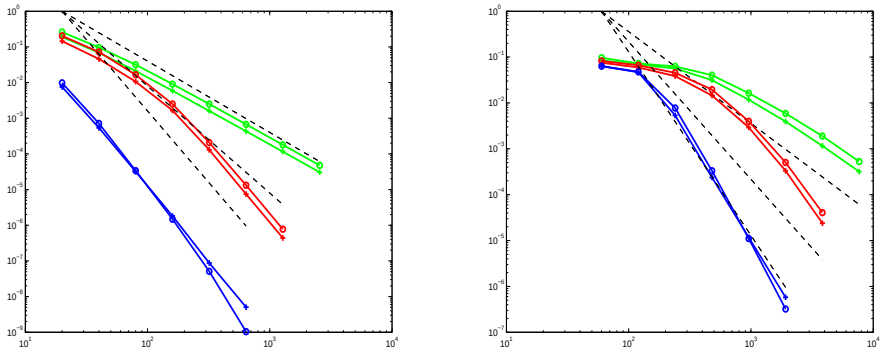

FIG. 4.2. *Accuracy plots. Low frequency (left) and high frequency (right) tests. From top to bottom, and green, red, and blue, respectively: second, third, and fourth order schemes. Semiconservative: +; and fully conservative: •.*

4.3. Euler equations. We consider the standard Euler equations of compressible gas dynamics in one dimension. In the notation of (2.3) $U = [\rho, m, E]$, where ρ is the density, $m = \rho v$ is the momentum, v is the velocity, and E is the total energy per unit volume. The pressure p is linked to the other quantities by the equation of state. Here we take $E = \frac{1}{2}\rho v^2 + \frac{1}{\gamma-1}p$, with $\gamma = 1.4$ the polytropic constant for air.

$$\partial_t \begin{pmatrix} \rho \\ \rho v \\ E \end{pmatrix} + \partial_x \begin{pmatrix} \rho v \\ \rho v^2 + p \\ v(E+p) \end{pmatrix} = 0. \tag{4.3}$$

When the solution is smooth, system (4.3) can be written in terms of primitive variables, obtaining a system of the form (2.6) with $V = [\rho, v, p]$, namely

$$\partial_t \begin{pmatrix} \rho \\ v \\ p \end{pmatrix} + \begin{pmatrix} u & \rho & 0 \\ 0 & v & 1/\rho \\ 0 & \gamma p & v \end{pmatrix} \partial_x \begin{pmatrix} \rho \\ v \\ p \end{pmatrix} = 0. \tag{4.4}$$

As an example, we consider Lax's Riemann problem, which is a standard benchmark in computational gas dynamics. The left and right states are

$$\begin{pmatrix} \rho_L \\ v_L \\ p_L \end{pmatrix} = \begin{pmatrix} 0.445 \\ 0.6989 \\ 3.5277 \end{pmatrix} \qquad \begin{pmatrix} \rho_R \\ v_R \\ p_R \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0 \\ 0.5710 \end{pmatrix}.$$

In this test, a high pressure gas on the left is impinging against a stationary low pressure gas. Figure 4.3 contains the density profiles obtained with the second order FC scheme (on the left) and the SC scheme on the right, for several values of the number of grid points: $N = 100$, 200, 400, 800. As expected, the solution converges to the exact profile (shown with the dashed line) under grid refinement, but it is noteworthy that the SC scheme and the FC one provide undistinguishable solutions.

We do not expect gains in efficiency in Euler equations, using the semiconservative approach, because the inverse of the map $u = \mathcal{M}(v)$, needed by the fully conservative scheme to compute the flux, can be written explicitly, and it is fast to compute. On the other hand, the SC approach requires one more reconstruction per step (from cell averages to point values), and one application of the direct map per stage, to compute the artificial diffusion correction. It is not surprising therefore that the computational

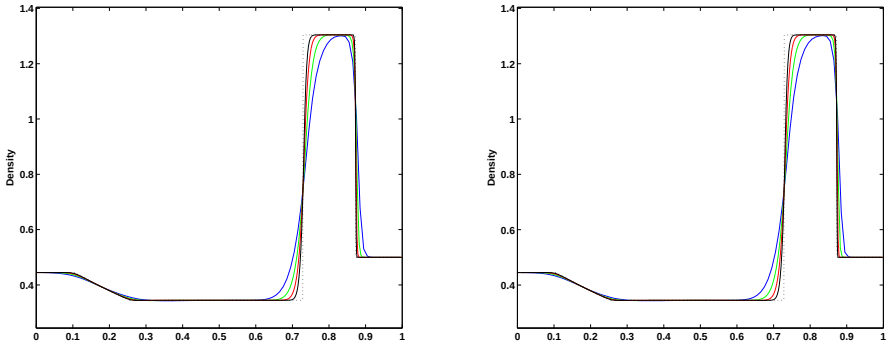

FIG. 4.3. *Lax's test, density profile with the second order FC2 (left) and SC2 (right) schemes, with $N = 100, 200, 400, 800$ (blue, green, red, black, respectively). The dashed profile is the exact solution.*

TABLE 4.1
Computational costs for Lax's test, in seconds of CPU. The two columns on the right refer to the scheme with characteristic projection (CP).

N	FC2	SC2	FC4	SC4	FC4 CP	SC4 CP
100	0.148	0.131	0.295	0.403	9.153	9.244
200	0.213	0.224	0.745	1.002	35.44	33.42
400	0.536	0.559	2.158	2.859	143.4	130.3
800	1.493	1.557	6.940	9.037	559.6	516.4

times of the SC schemes are slightly higher than those obtained by the corresponding FC; see the first four columns of Table 4.1. The CPU times were obtained running the code in MATLAB on a 2.9 GHz Intel Core i5 machine. The code is vectorized, except for the runs with the reconstruction on characteristic variables, as in the last two columns of the table.

Figure 4.4 contains a detail of the density peak obtained with the fourth order FC4 (on the left) and SC4 (on the right). It is well known that high order WENO schemes develop spurious oscillations in Riemann problems, with amplitude decreasing under grid refinement. In fact, this is precisely the meaning of *essentially* nonoscillatory reconstructions. This essentially nonoscillatory behavior is quite apparent in the figure, but note that the SC solution is less oscillatory than its FC counterpart, although in both cases the amplitude of the oscillations decreases under grid refinement.

These oscillations arise in the first steps of the computation, when the waves originated by the Riemannn problem are so close that it is impossible to find a stencil containing only one discontinuity. This problem can be cured projecting the unknown along characteristic directions, before performing the reconstruction, and computing the reconstruction along the direction of the eigenvectors. This procedure was outlined in [27] and it is very effective. The drawback is that it is computationally expensive. Figure 4.5 shows the peak in the density of Lax's Riemann problem when this device is applied. The computational cost is reported in the last two columns of Table 4.1. Now, the SC computation is slightly faster, because one variable is already a characteristic variable.

4.4. Relativistic gas dynamics. As we have seen in the previous section, the semiconservative approach reproduces the correct shock speeds, even though the stage

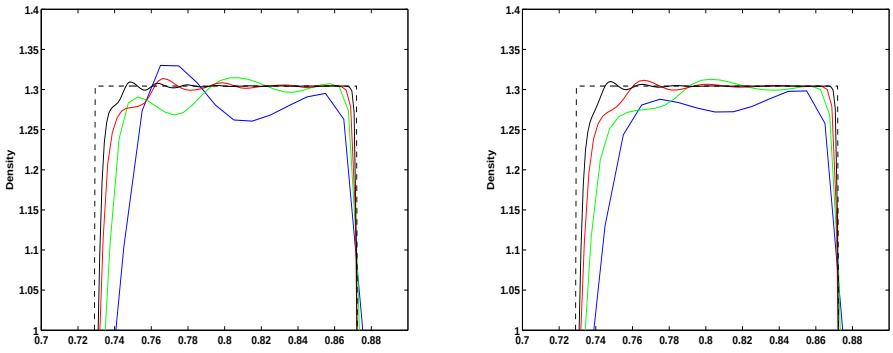

FIG. 4.4. *Lax's test, detail of the density profile with the fourth order FC4 (left) and SC4 (right) schemes, with* $N = 100, 200, 400, 800$ *(blue, green, red, black, respectively). The dashed profile is the exact solution.*

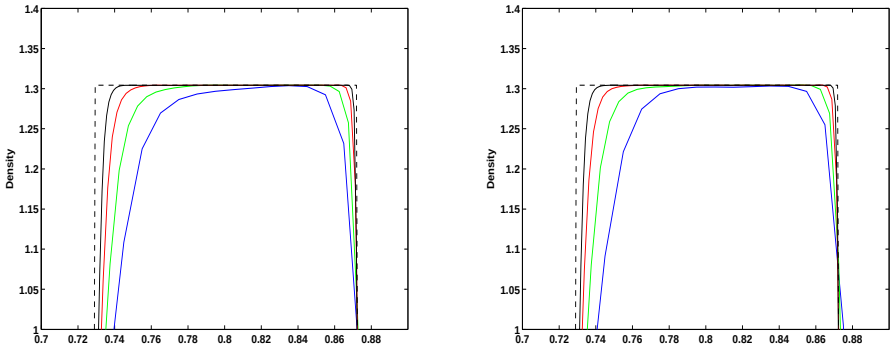

FIG. 4.5. *Lax's test, detail of the density profile with the fourth order FC4 (left) and SC4 (right) schemes, with* $N = 100, 200, 400, 800$ *(blue, green, red, black, respectively). Reconstruction along characteristic directions. The dashed profile is the exact solution.*

values are computed in nonconservative form. Since the mapping between conservative and nonconservative variables $u = \mathcal{M}(v)$ is easily invertible in Euler equations, the semiconservative approach is not computationally faster than standard finite volume schemes. We expect to gain in efficiency when the semiconservative approach is applied to equations for which the mapping $\mathcal{M}$ is not easily invertible.

As an example of this type, we consider the relativistic gas dynamic equations ([22]; see also [23] for a review), which can be written as

$$\partial_t \begin{pmatrix} D \\ S \\ \tau \end{pmatrix} + \partial_x \begin{pmatrix} Dv \\ Sv + p \\ S - Dv \end{pmatrix} = 0, \tag{4.5}$$

where the conservative variables are mass density D, momentum density S, and energy density τ in the laboratory frame of reference. These quantities are linked to the density ρ, the velocity v, and the pressure p through the relations

$$D = \rho W, \tag{4.6}$$

$$S = \rho h W^2 v, \tag{4.7}$$

$$\tau = \rho h W^2 - p - D, \tag{4.8}$$

where $W = (1-v^2)^{-1/2}$ is the Lorentz factor in which v has been nondimensionalized with the speed of light, thus $v \in [-1,1]$; h is the enthalpy per unit mass, $h = 1+e+\frac{p}{\rho}$, and e is the internal energy per unit mass. The pressure p is given by the equation of state, $p = \rho e(\gamma-1)$. To compute the flux on the right-hand side of (4.5), one must compute v and p from the conservative variables.

The velocity v can be easily written in terms of the pressure and of conservative variables using (4.7) and (4.8),

$$v = \frac{S}{\tau + D + p}.$$

The internal energy is $\rho e = \rho h - \rho - p$, and the enthalpy can be written as a function of the pressure and of conservative variables as

$$\rho h = \frac{\tau + D + p}{W^2}.$$

Substituting these quantities in the equation of state $p = (\gamma-1)\rho e$, one obtains a nonlinear equation for the pressure, namely

$$0 = \Im(p(D,S,\tau)) = \left(\gamma W^2 - (\gamma-1)\right) p - (\gamma-1)\left(\tau + D(1-W)\right). \tag{4.9}$$

The conservative variables (D,S,τ) clearly must satisfy $D > 0, \tau > 0$. As already noted, the velocity v cannot surpass the speed of light, i.e., $-1 \le v \le 1$. This condition implies that $\tau + D \ge |S|$. Finally, the root of $\mathcal{F}(p) = 0$ must be positive, and this request brings in a further restriction. In fact, $\mathcal{F}(p)$ is a monotone increasing function (see [22]). Thus the pressure is positive if $\mathcal{F}(p=0) < 0$, which is satisfied provided

$$(\tau + D)^2 > D^2 + S^2. \tag{4.10}$$

In this case, the function $\mathcal{F}(p)$ has a single, positive root. To compute the flux, the nonlinear equation (4.9) must be solved at each grid point. In our tests, (4.9) is solved with Newton's method, using, as a starting guess for the pressure, the local value from the previous time step. Note, however, that condition (4.10) may be violated when spurious oscillations occur, especially when the flow is characterized by a total energy which is almost completely kinetic. In this case, (4.9) may yield a negative value for the pressure or no solution at all, and the integration breaks down. Thus, it is crucial to use nonoscillatory schemes when dealing with low pressure, relativistic gas dynamics.

Clearly, if $v \ll 1$, classical mechanics holds, and one recovers standard compressible gas dynamics.

The equations of relativistic gas dynamics in primitive variables are [22]

$$\partial_t \begin{pmatrix} \rho \\ v \\ p \end{pmatrix} + \begin{pmatrix} v & \dfrac{\rho}{1-v^2c^2} & \dfrac{-v}{hW^2(1-v^2c^2)} \\ 0 & v\dfrac{1-c^2}{1-v^2c^2} & \dfrac{1}{\rho hW^4(1-v^2c^2)} \\ 0 & \dfrac{\rho hc^2}{1-v^2c^2} & \dfrac{v(1-c^2)}{1-v^2c^2} \end{pmatrix} \partial_x \begin{pmatrix} \rho \\ v \\ p \end{pmatrix} = 0, \tag{4.11}$$

where $c^2 = \gamma p/(\rho h)$. These are the equations which will be used in the computation of the stage values.

Now, for the standard finite volume scheme FC, given the cell averages $\bar{D}^n$, $\bar{S}^n$, $\bar{\tau}^n$ one needs to compute the ν stage values, and each stage value requires the evaluation of the inverse of the map $U = \mathcal{M}(V)$ defined by (4.6)–(4.8), which needs the solution of $\mathcal{F}(p; D^{(i)}, S^{(i)}, \tau^{(i)}) = 0$. In the semiconservative schemes SC, instead, given the

cell averages $\bar{D}^n$, $\bar{S}^n$, $\bar{\tau}^n$, we compute the point values D^n, S^n, τ^n, and the primitive variables ρ, v, p inverting again the map $U = \mathcal{M}(V)$, but this is done only once per time step. Next, the ν stages are computed from (4.11), which does not require the inversion of $\mathcal{M}(V)$. Once the stage values $\rho^{(i)}, v^{(i)}, p^{(i)}$ are known, the stage values for the conservative variables $D^{(i)}, S^{(i)}, \tau^{(i)}$ are easily found. This explains why the new SC schemes are faster with respect to the fully conservative schemes in the relativistic case.

We illustrate the behavior of the schemes with three shock tube problems. The first two tests can be found in [22]. The left and right states for the first test are given by

$$\textbf{Test 1} \qquad \begin{pmatrix} \rho \\ v \\ p \end{pmatrix}_L = \begin{pmatrix} 10 \\ 0 \\ 13.3 \end{pmatrix}, \qquad \begin{pmatrix} \rho \\ v \\ p \end{pmatrix}_R = \begin{pmatrix} 1 \\ 0 \\ 0.6\ 10^{-6} \end{pmatrix}.$$

In this case, a gas expands into an extremely low pressure gas. The polytropic parameter is $\gamma = \frac{5}{3}$, the final time is $T = 0.36$, and the Courant number is CFL$= 0.45$ for all schemes. The profiles for density, velocity, and pressure for the second order FC2 and SC2 can be seen in Figure 4.6. The exact solution was computed thanks to the Riemann solver described in [23].

It is apparent that all features of the solution are correctly reproduced by the semiconservative SC scheme. For the fourth order schemes, we show a peak of the density profiles in Figure 4.7. Again, we note that the semidiscrete SC schemes are less oscillatory than the standard finite volume method of the same order. The computational times of the four schemes tested are listed in Table 4.2. Now, the semiconservative schemes are faster than their fully conservative counterpart, because the costly inverse of the map $u = \mathcal{M}(v)$ has to be computed only once per time step. Clearly, the difference is much more apparent in the fourth order case.

The second test is again from [22], but an analogous set up can also be found in [26], [33]:

$$\textbf{Test 2} \qquad \begin{pmatrix} \rho \\ v \\ p \end{pmatrix}_L = \begin{pmatrix} 1 \\ 0 \\ 1000 \end{pmatrix}, \qquad \begin{pmatrix} \rho \\ v \\ p \end{pmatrix}_R = \begin{pmatrix} 1 \\ 0 \\ 0.01 \end{pmatrix}.$$

This shock tube problem results in a rarefaction moving toward the left and a contact and shock traveling right. The difficulty of this test is due to the fact that the contact and the shock travel at almost equal speeds, so that high order schemes have difficulties in selecting a nonoscillatory stencil.

The results obtained with the fourth order semiconservative scheme appear in Figure 4.8. The fully conservative, fourth order scheme fails on this test, because condition (4.10) is violated across the contact wave, after the computation of the first stage values.

A further test, Test 3, is drawn from [33]. The initial left and right states are given by

$$\textbf{Test 3} \qquad \begin{pmatrix} \rho \\ v \\ p \end{pmatrix}_L = \begin{pmatrix} 1 \\ 0.9 \\ 1 \end{pmatrix}, \qquad \begin{pmatrix} \rho \\ v \\ p \end{pmatrix}_R = \begin{pmatrix} 1 \\ 0 \\ 10 \end{pmatrix}.$$

It describes a low pressure gas impinging against a high pressure gas. Figure 4.9 contains the resulting density profiles for the fourth order schemes, with a zoom on the contact wave on the bottom of the figure. In this case, the semiconsenservative scheme is more oscillatory than the fully conservative finite volume scheme.

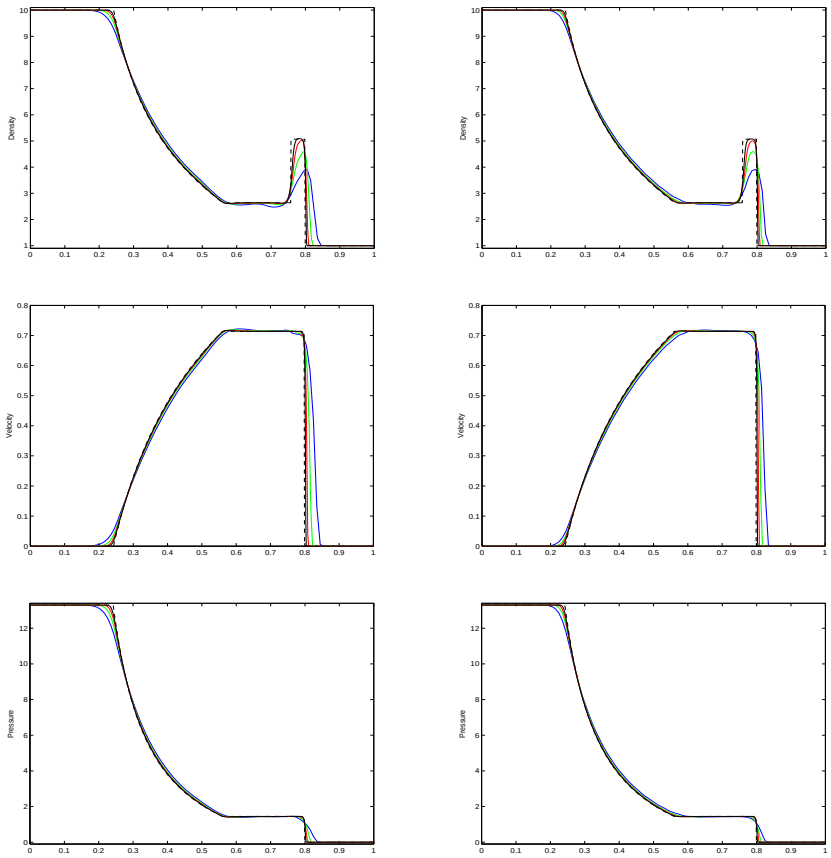

FIG. 4.6. *Martí Müller Test* 1, *density, velocity, and pressure profiles with the second order FC*2 *(left) and SC*2 *(right) schemes, with* $N = 100, 200, 400, 800$ *(blue, green, red, black, respectively). The dashed profile is the exact solution.*

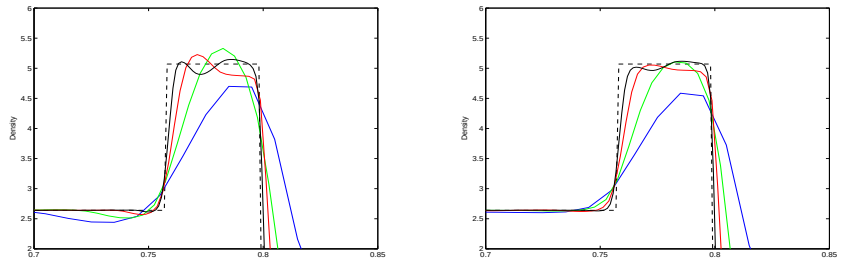

FIG. 4.7. *Martí Müller Test* 1, *zoom on the density profiles with the fourth order FC*4 *(left) and SC*4 *(right) schemes, with* $N = 100, 200, 400, 800$ *(blue, green, red, black, respectively). The dashed profile is the exact solution.*

For this test, we also show the error versus the CPU time of first (green), second (blue), and fourth (red) order schemes, see Figure 4.10. The results obtained with the fully conservative FC schemes are represented with a dot, while the results yielded by the semiconservative schemes appear with a + marker. It is clear that the SC schemes in all cases (except on a very coarse grid) yield consistently smaller CPU times for the same error. This is not a test in which high order schemes work at their

TABLE 4.2
Computational costs for Relativistic gas dynamics, in seconds of CPU. Test 1 *from Martí Müller.*

N	FC2	SC2	FC4	SC4
100	0.155	0.288	0.668	0.409
200	0.341	0.260	1.390	0.788
400	0.798	0.577	3.763	1.922
800	1.973	1.506	10.783	5.611

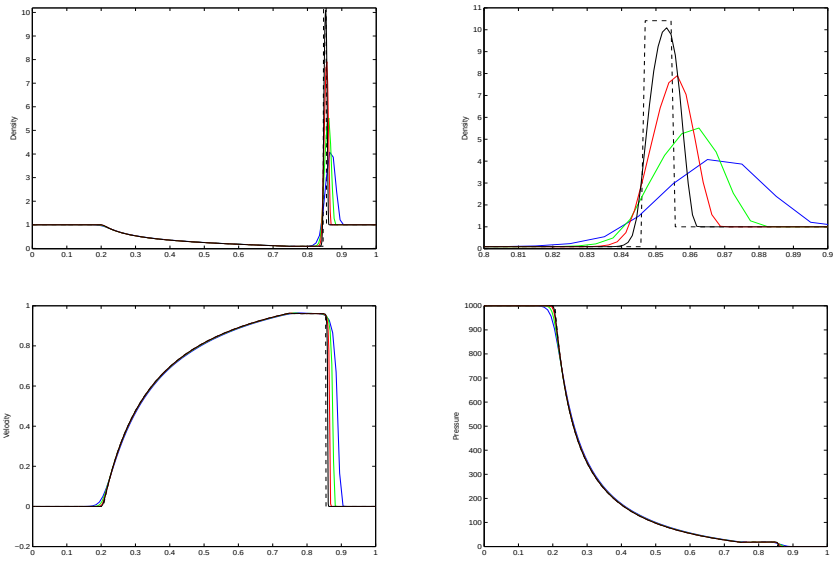

FIG. 4.8. *Martí Müller Test* 2. *At the top: density profile with a zoom on the contact wave. Bottom: velocity and pressure, SC*4 *with* $N = 100, 200, 400, 800$ *(blue, green, red, black, respectively). The dashed profile is the exact solution.*

best, because the solution of a Riemann problem is not full of structure. However, in this case the exact solution is known and quantitative results can be carried out. The most interesting point is that SC is indeed faster than fully conservative schemes.

4.4.1. Two-dimensional tests. Finally, we consider two-dimensional tests. The equations for relativistic gas dynamics in primitive variables are

$$\partial_t V + A_x \partial_x V + A_y \partial_y V = 0, \tag{4.12}$$

where the Jacobians of the flux are given by

$$A_x = \begin{pmatrix} u & \rho G & 0 & -\frac{uG}{hW^2} \\ 0 & uG(1-c^2) & 0 & \frac{G}{\rho h W^2}(1-u^2-c^2v^2) \\ 0 & -\frac{c^2 G}{W^2}v & u & -\frac{G(1-c^2)}{\rho h W^2}uv \\ 0 & \rho h c^2 G & 0 & G(1-c^2)u \end{pmatrix} \tag{4.13}$$

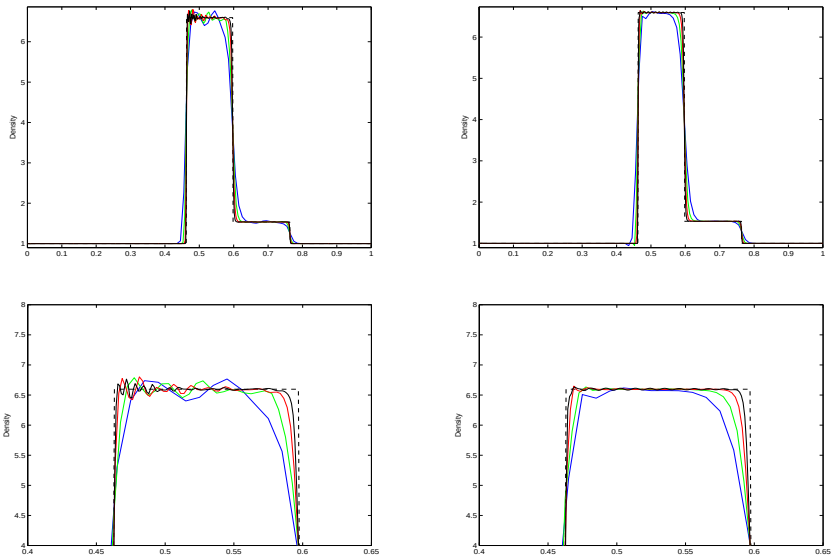

FIG. 4.9. *Zhao Tang, Test 3. At the top: density profiles for SC4 (left) and FC4(right). Bottom: zoom on the contact wave.* $N = 100, 200, 400, 800$ *(blue, green, red, black, respectively). The dashed profile is the exact solution.*

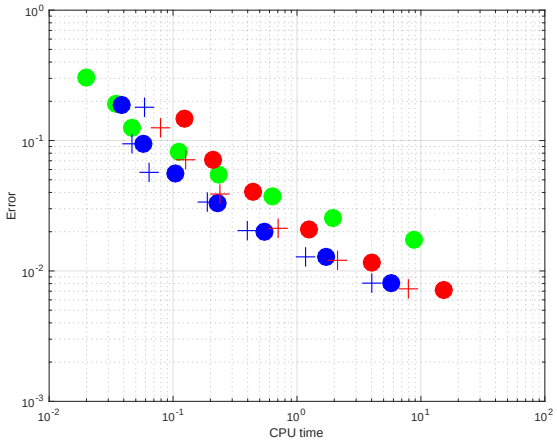

FIG. 4.10. *Error versus CPU time of first (green), second (blue), and fourth (red) order schemes. FC schemes are represented with a dot, SC schemes appear with a + marker.*

and

$$A_y = \begin{pmatrix} v & 0 & \rho G & -\frac{uG}{hW^2} \\ 0 & v & -\frac{c^2 G}{W^2}u & -\frac{G(1-c^2)}{\rho h W^2}uv \\ 0 & 0 & -G(1-c^2)v & \frac{G}{\rho h W^2}(1-c^2u^2-v^2) \\ 0 & 0 & \rho h c^2 G & G(1-c^2)v \end{pmatrix}. \tag{4.14}$$

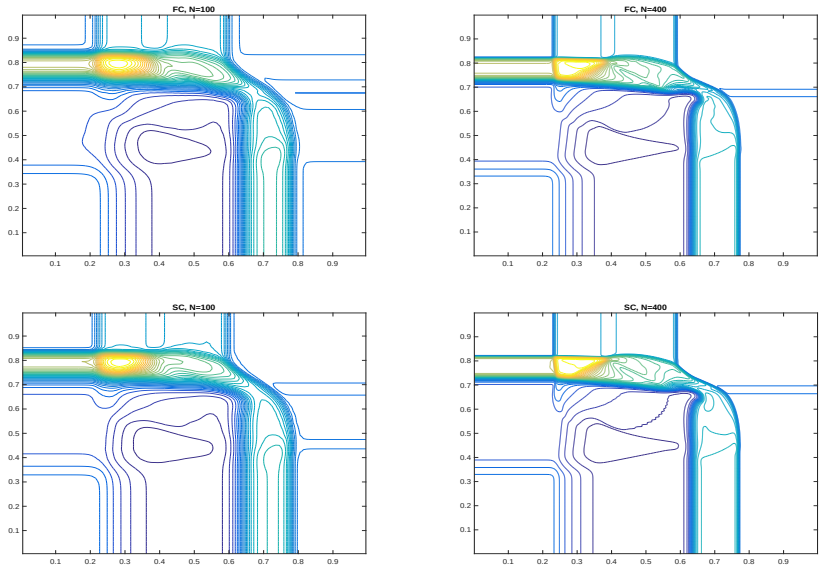

FIG. 4.11. *Two-dimensional Riemann problem for relativistic gas dynamics, density contours, second order FC (top) and SC (bottom), with* $N = 100$ *(left) and* $N = 400$ *(right) points per direction.*

Here, (u, v) are the components of the velocity in the x and y directions, respectively, $W^2 = 1/(1 - (u^2 + v^2))$ and $G = 1/(1 - c^2(u^2 + v^2))$. As a test, we propose a two-dimensional Riemann problem, in which the four states are given by

$$V_{NW} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad V_{NE} = \begin{pmatrix} 2 \\ -0.5 \\ 0.5 \\ 1 \end{pmatrix}, \quad V_{SW} = \begin{pmatrix} 2 \\ 0 \\ 0.5 \\ 1 \end{pmatrix}, \quad V_{SE} = \begin{pmatrix} 2 \\ 0 \\ 0.5 \\ 10 \end{pmatrix},$$

with NW labeling the northwest corner of the computational domain, and similarly for the other labels.

The computational domain is the square $Q = (0, 1)^2$, with free-flow boundary conditions. The final time is $t_f = 0.36$ and the origin of the Riemann problem is in the middle of Q.

We show results obtained with a dimension by dimension piecewise linear reconstruction for second order and the truly two-dimensional third order CWENO reconstruction of [4], to which we added the computation of the slopes. The results can be seen in Figure 4.11 for the second order scheme and Figure 4.12 for the third order scheme. Each figure contains 40 contour lines for the density ρ for the FC scheme (top plots) and for the SC scheme (bottom). The figures also show the effect of grid refinement: the number of grid points along each side is $N = 100$ for the left plots and $N = 400$ for the plots on the right. SC provides in all cases a slight improvement in the resolution of the discontinuities. Further, in this solution with a rich structure, the third order solution exhibits more details than the second order case.

The corresponding computational times can be found in Table 4.3. As in the one-dimensional case, the SC scheme is faster than its corresponding FC, and the computational gain increases with the order.

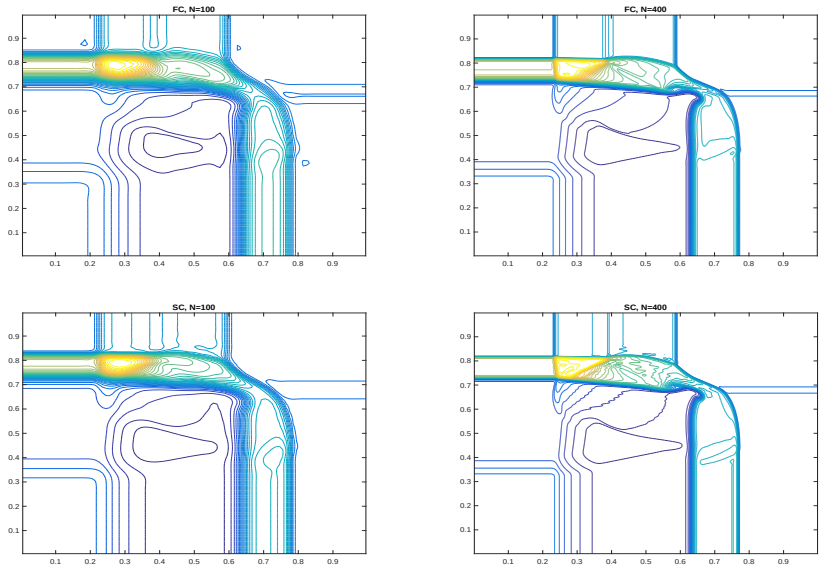

FIG. 4.12. *Two-dimensional Riemann problem for relativistic gas dynamics, density contours, third order FC (top) and SC (bottom), with $N = 100$ (left), and $N = 400$ (right) points per direction.*

TABLE 4.3
Computational costs for the two-dimensional relativistic Riemann problem, in seconds of CPU.

N	FC2	SC2	FC3	SC3
100	1.78	1.82	12.85	9.55
200	13.45	11.54	165.25	121.19
400	138.09	113.88	1875.12	1400.89

5. Conclusions. In this paper we have presented a novel approach to construct conservative finite volume methods for conservation laws. Although the final scheme is conservative and is able to capture shocks with the correct propagation speed, most of the computational work is performed using a nonconservative formulation, in nonconservative variables. This adds a tremendous flexibility to the choice of the unknown variables and on the form of the equations on which most of the computational effort is carried out. We explored in some detail two applications, namely classic and relativistic gas dynamics. In both cases, the nonconservative form of the equations based on primitive variables was chosen. In classical gas dynamics, it is observed that in many cases this choice provides less oscillatory solutions than in standard WENO schemes based on conservative variables. In relativistic gas dynamics, high order schemes greatly benefit from the nonconservative formulation, which allows us to compute the evolution of the fields without solving the nonlinear equation to determine the pressure from the conservative variables. Such an equation has to be solved only once per cell per time step, as opposed to what happens in standard finite volume schemes based on ν stages Runge–Kutta schemes, for which such an equation has to be solved ν times per cell per time step.

The method can be easily extended to the construction of conservative finite difference schemes, which may be very convenient for efficient computation in several space dimensions.

We believe there are several other contexts in which the flexibility introduced by the semiconservative approach can be successfully exploited for producing more effective codes, which are either more efficient or more accurate for the same discretization parameters. The use of the new approach in other contexts as well as in several space dimensions is currently under investigation.

REFERENCES

[1] F. Aràndiga, A. Baeza, A. M. Belda, and P. Mulet, *Analysis of WENO schemes for full and global accuracy*, SIAM J. Numer. Anal., 49 (2011), pp. 893–915.

[2] F. Bianco, G. Puppo, and G. Russo, *High order central schemes for hyperbolic systems of conservation laws*, SIAM J. Sci. Comput., 21 (1999), pp. 294–322.

[3] P. Carlini, E. Ferretti, and G. Russo, *A weighted essentially nonoscillatory, large time-step scheme for Hamilton-Jacobi equations*, SIAM J. Sci. Comput., 23 (2005), pp. 1071–1091.

[4] M. Castro and M. Semplice, *Third and Fourth Order Well-Balanced Schemes for the Shallow Water Equations Based on the CWENO Reconstruction*, arXiv:1807.10069, 2018.

[5] M. J. Castro, J. M. Gallardo, and A. Marquina, *Approximate Osher-Solomon schemes for hyperbolic systems*, Appl. Math. Computation, 272 (2016), pp. 347–368, https://doi.org/10.1016/j.amc.2015.06.104.

[6] I. Cravero, G. Puppo, M. Semplice, and G. Visconti, *CWENO reconstructions for balance laws*, Math. Comp., 87 (2018), pp. 1689–1719.

[7] I. Cravero and M. Semplice, *On the accuracy of WENO and CWENO reconstructions of third order on nonuniform meshes*, J. Sci. Comput., 67 (2016), pp. 1219–1246.

[8] C. Dafermos, *Hyperbolic Conservation Laws in Continuum Physics*, 3rd ed., Springer, New York, 2010.

[9] P. Degond, P.-F. Peyrard, G. Russo, and P. Villedieu, *Polynomial upwind schemes for hyperbolic systems*, C. R. Acad. Sci. Ser. I Math., 328 (1999), pp. 479–483, https://doi.org/https://doi.org/10.1016/S0764-4442(99)80194-3.

[10] E. Godlewski and P. Raviart, *Hyperbolic Systems of Conservation Laws*, Math. Appl., Société de Mathématiques Appliquées et Industrielles, 1991.

[11] E. Godlewski and P. A. Raviart, *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, Appl. Math. Sci., 118, Springer, New York, 1996.

[12] S. Gottlieb, C. Shu, and E. Tadmor, *Strong stability preserving high order time discretization methods*, SIAM Rev., 43 (2001), pp. 89–112.

[13] E. Hairer, S. P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations* I. *Nonstiff Problems*, Comput. Math. 8, Springer, New York, 1993.

[14] E. Harabetian and R. Pego, *Nonconservative hybrid shock capturing schemes*, J. Comput. Phys., 105 (1993), pp. 1–13.

[15] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.

[16] G.-S. Jiang and E. Tadmor, *Nonoscillatory central schemes for multidimensional hyperbolic conservation laws*, SIAM J. Sci. Comput., 19 (1998), pp. 1892–1917.

[17] R. J. Le Veque, *Numerical Methods for Conservation Laws*, 2nd ed., Lectures in Math. ETH Zürich, Birkhäuser Verlag, Basel, 1992.

[18] R. J. Le Veque, *Finite Volume methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, NY, 2002.

[19] D. Levy, G. Puppo, and G. Russo, *Central WENO schemes for hyperbolic systems of conservation laws*, Math. Model. Numer. Anal., 33 (1999), pp. 547–571.

[20] D. Levy, G. Puppo, and G. Russo, *Compact central WENO schemes for multidimensional conservation laws*, SIAM J. Sci. Comput., 22 (2000), pp. 656–672.

[21] D. Levy, G. Puppo, and G. Russo, *A fourth-order central WENO schemes for multidimensional hyperbolic systems of conservation laws*, SIAM J. Sci. Comput., 24 (2002), pp. 480–506.

[22] J. Martì and E. Müller, *Extension of the piecewise parabolic method to one-dimensional relativistic hydrodynamics*, J. Comput. Phys., 123 (1996), pp. 1–14.

[23] J. Martì and E. Müller, *Numerical Hydrodynamics in Special Relativity*, Living Reviews in Relativity, Max Planck Institute, 2003.

[24] H. Nessyahu and E. Tadmor, *Non-oscillatory central differencing for hyperbolic conservation laws*, J. Comput. Phys., 87 (1990), pp. 408–463.

[25] L. Pareschi, G. Puppo, and G. Russo, *Central Runge-Kutta schemes for conservation laws*, SIAM J. Sci. Comput., 26 (2005), pp. 979–999.

[26] S. Qamar and M. Yousaf, *Application of a discontinuous Galerkin finite element method to special relativistic hydrodynamic models*, Comput. Math. Appl., 65 (2013), pp. 1220–1232.
[27] J. Qiu and C. Shu, *On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes*, J. Comput. Phys., 183 (2002), pp. 187–209.
[28] P. L. Roe, *Approximate Riemann solvers, parameter vectors, and difference schemes*, J. Comput. Phys., 43 (1981), pp. 357–372.
[29] M. Semplice, A. Coco, and G. Russo, *Adaptive mesh refinement for hyperbolic systems based on third-order compact WENO reconstruction*, J. Sci. Comput., 66 (2016), pp. 692–724.
[30] J. Shi, C. Hu, and C.-W. Shu, *A technique of treating negative weights in WENO schemes*, J. Comput. Phys., 175 (2002), pp. 108–127.
[31] C.-W. Shu, *Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws*, Lecture Notes in Math., 1697, Springer, New York, 1998.
[32] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 438–471.
[33] J. Zhao and H. Tang, *Central Runge-Kutta Discontinuous Galerkin Methods for the Special Relativistic Hydrodynamics*, arXiv:1609.06792v1, 2016.