

Numerical Entropy and Adaptivity for Finite Volume Schemes

Gabriella Puppo^{1,*} and Matteo Semplice²

¹ Dipartimento di Matematica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italia.

² Dipartimento di Fisica e Matematica, Università dell'Insubria, Via Valleggio 11, 22100 Como, Italia.

Received 25 September 2009; Accepted (in revised version) 21 January 2011

Communicated by Chi-Wang Shu

Available online 2 August 2011

Abstract. We propose an a-posteriori error/smoothness indicator for standard semi-discrete finite volume schemes for systems of conservation laws, based on the numerical production of entropy. This idea extends previous work by the first author limited to central finite volume schemes on staggered grids. We prove that the indicator converges to zero with the same rate of the error of the underlying numerical scheme on smooth flows under grid refinement. We construct and test an adaptive scheme for systems of equations in which the mesh is driven by the entropy indicator. The adaptive scheme uses a single nonuniform grid with a variable timestep. We show how to implement a second order scheme on such a space-time non uniform grid, preserving accuracy and conservation properties. We also give an example of a p -adaptive strategy.

AMS subject classifications: 65M08, 65M50, 76M12

Key words: Finite volume schemes, hyperbolic systems, local grid refinement, entropy.

1 Introduction

The a-posteriori error indicator proposed in this paper can be constructed for multidimensional systems of conservation laws. However, for simplicity, in the following we will mainly consider one-dimensional hyperbolic systems of equations of the form:

$$u_t + f_x(u) = 0. \quad (1.1)$$

*Corresponding author. Email addresses: gabriella.puppo@polito.it (G. Puppo), matteo.semplice@uninsubria.it (M. Semplice)

Here $u(t,x)$ is a function from $\mathbb{R}^+ \times \mathbb{R} \rightarrow \mathbb{R}^m$, where m is the number of equations of the system, f is the flux function, and we assume that f is a smooth function from $\mathbb{R}^m \rightarrow \mathbb{R}^m$. We suppose that the system is strictly hyperbolic, i.e., that the Jacobian $A(u) = J(f)$ has m real eigenvalues and a complete set of corresponding eigenvectors.

It is well known that solutions of initial value problems for (1.1) may lose their regularity in a finite time even if the initial data are smooth, developing shock waves. In this case, the solution must be understood in the weak sense, and uniqueness of the solution is lost. In order to retrieve uniqueness, the system must be completed with an entropy inequality, characterizing the unique admissible weak solutions of (1.1). Thus, we will consider systems of the form (1.1) possessing an entropy-entropy flux pair, that is we assume there exist a convex function $\eta(u)$ and a corresponding entropy flux $\psi(u)$ verifying the compatibility condition $\nabla^T \eta A(u) = \nabla^T \psi$ (see [8]). Then it is well known that entropy solutions of (1.1) must satisfy the entropy inequality

$$\eta_t + \psi_x(u) \leq 0 \quad (1.2)$$

in the weak sense for all entropies.

Numerical integration of (1.1) is challenging because the solution can exhibit a very complex structure: discontinuities may arise and disappear through the interaction with other waves present in the flow. For these reasons, several attempts to the construction of adaptive grids have appeared in the literature. Adaptive grids seek to achieve a good resolution in regions where the flow varies rapidly, and an effective error control where the flow does not have a complex structure, and high resolution is not needed.

High resolution can be obtained using a fine grid, and/or using a high order scheme. In the first case, the CPU time increases rapidly, because the CFL stability condition imposes an upper bound on the grid ratio $\lambda = \Delta t/h$, where Δt is the time step and h is the grid spacing, so that a fine grid requires a small time step. In the second case, the presence of discontinuities may result in non-linear instabilities in a high order numerical solution, so that the scheme itself must become non linear to prevent the onset of spurious oscillations. There is a huge literature on non-oscillatory high order schemes for hyperbolic problems. Here we just mention the reference [18] especially for second order schemes, and [28] for higher order schemes. As the order increases, these schemes become more costly, not only because their structure and the mechanisms designed to prevent an oscillatory behavior become more complex, but also because usually they seek non oscillatory stencils to compute the solution, and this requires the use of characteristic variables to prevent the selection of stencils containing wave interactions, see [27].

Finally, finite volume schemes usually give a good resolution of shocks even on coarse grids, because the smearing effect of numerical viscosity is counterbalanced by the steepening mechanism of converging characteristics. This does not occur on contact discontinuities, along which the characteristic fields are parallel, and only numerical diffusion is active [11].

Thus an effective adaptive algorithm must be driven by an indicator which should be able not only to provide a robust a posteriori measure of the local error, but should also

recognize the qualitative structure of the flow, to pivot both grid and scheme adaptivity.

Error control and grid adaptivity have been particularly successful in the framework of Finite Element Methods for elliptic or convection-diffusion problems. Here we are particularly interested in error control through a posteriori error estimates based on a measure of the local residual, as in [33]. Since in hyperbolic problems, the solution exists only in the weak sense, the residual can be estimated weakly, as in [14]. There the residual is estimated integrating the equations against compactly supported test functions. For high order schemes, the smoothness of the test functions must be increased, and their support therefore widens. This of course interferes with the localization of the indicator, which provides information on a cluster of cells whose size depends on the support of the test function, both in space and in time. See also [15].

In another approach, the error indicator is computed estimating the weak residual of the entropy inequality, as in [16] for first order upwind schemes applied to scalar conservation laws, and generalized in [5] to higher order Discontinuous Galerkin schemes, still in the case of scalar conservation laws, for which Kruzkov entropies can be defined.

Another interesting approach to grid adaptivity is based on the solution of an adjoint problem, depending on a functional whose purpose is the minimization of the error not in the whole computational region, but on a certain quantity of interest, as, for instance, the lift of an airfoil, see [13] or the monograph [7].

We also mention an approach to adaptivity based on multiresolution techniques, following the pioneering work by Harten [12], and later applied by [2] and [22].

In this work, we are interested in finite volume schemes. In this case, system (1.1) and the entropy inequality (1.2) are integrated on control volumes

$$V_j^n = I_j \times [t^n, t^{n+1}],$$

where $I_j = [x_j - h_j/2, x_j + h_j/2]$ is the interval of width h_j centered around the grid point x_j , and $t^n = t^{n-1} + \Delta t_n$. For the time being, we will suppose that h_j and Δt_n are constants, so that $x_{j+1} - x_j = h_j \equiv h$ and $\Delta t_n \equiv \Delta t$. Let:

$$\bar{u}_j^n = \frac{1}{h} \int_{I_j} u(x, t^n) dx$$

be the cell average of the function u around the point x_j at the time t^n . Integrating (1.1) on the control volume V_j^n we find the finite volume formulation for the hyperbolic system:

$$\bar{u}_j^{n+1} = \bar{u}_j^n - \frac{1}{h} \int_{t^n}^{t^{n+1}} [f(u(I_j^+, \tau)) - f(u(I_j^-, \tau))] d\tau, \quad (1.3)$$

where I_j^+ and I_j^- denote respectively the right and left end points of the interval I_j . Note that this equation is still exact. Integrating the entropy inequality (1.2) on the same control volume, we find the finite volume formulation of the entropy inequality, namely:

$$\bar{\eta}_j^{n+1} - \bar{\eta}_j^n + \frac{1}{h} \int_{t^n}^{t^{n+1}} [\psi(u(I_j^+, \tau)) - \psi(u(I_j^-, \tau))] d\tau \leq 0. \quad (1.4)$$

If the solution is smooth, it is well known that both (1.3) and (1.4) are equalities. On the other hand, if the solution has a singularity, than (1.3) is still satisfied, while the sign of (1.4) selects the unique physically relevant weak solution of (1.1).

We now consider a finite volume scheme for (1.1). Let $U(t, x)$ be the numerical solution and denote with \bar{U}_j^n its cell average on the interval I_j at time t^n . The finite volume scheme can be written as

$$\bar{U}_j^{n+1} = \bar{U}_j^n - \lambda (F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}), \quad (1.5)$$

where $F_{j+1/2}$ is a suitable numerical flux consistent with the exact flux $f(u)$ and $\lambda = \Delta t/h$ is the mesh ratio. Following [26], we define the numerical entropy production

$$S_j^n = \frac{1}{\Delta t} (\overline{\eta(U^{n+1})}_j - \overline{\eta(U^n)}_j + \lambda [\Psi_{j+\frac{1}{2}} - \Psi_{j-\frac{1}{2}}]), \quad (1.6)$$

where $\Psi_{j+1/2}$ is a numerical entropy flux. In the following, we will see that, if $\Psi_{j+1/2}$ is consistent with the exact entropy flux $\psi(u)$, then S_j^n converges to zero with the same rate of the local truncation error, in regions of smoothness. On the other hand, if a shock occurs in the control volume V_j^n , then the local entropy production S_j^n is bounded by terms of order $\mathcal{O}(1/\Delta t)$. For these reasons, the quantity S_j^n can be considered as an a posteriori error indicator for the numerical solution obtained in (1.5). In fact, it contains information on the size of the local truncation error, and it signals the presence of singularities.

Moreover, we will prove that, if the numerical entropy flux is properly chosen, the entropy production built on first order monotone schemes is essentially negative definite, in the sense that positive overshoots in the cell entropy inequality are possible, but only on non monotone data and their amplitude decreases fast under grid refinement for smooth flows.

A local measure of the entropy production is also used in [10] to add artificial dissipation to numerical schemes for conservation laws only in those cells where the entropy dissipation is large. A similar approach can be found in [6], where an entropy conservative scheme is coupled to a local dissipation in an effort to obtain convergence results, and to control the amount of dissipation needed to ensure a non oscillatory behavior of the scheme.

We propose a 1D adaptive algorithm based on a single non-uniform mesh, unlike the multiple overlayed grids that characterize the AMR implementation of CLAWPACK, [1, 19]. Our algorithm makes use of dyadic grids in space, which are stored in a binary tree structure. The storage is similar to the one of [22], although we work in physical space, while they use the space of wavelet coefficients. For the time integration we consider first a uniform time step algorithm, in which Δt is thus dictated by the smallest cell in the grid, as in [5]. This is very simple, but can be extremely penalizing on highly nonuniform grids. Thus we also consider a nonuniform timestep algorithm, where each cell has a fixed CFL number. This requires a local time-stepping strategy, which is not yet a standard procedure. Unlike the approach followed in CLAWPACK, we start the update process from the finest grid, using the smallest time step, and then we iterate up to the

coarsest level. In this fashion, the scheme always uses the most accurate information available. Our method tries to overcome the problems of previous algorithms at the interfaces between cells of different sizes: [24] suffers from lack of accuracy and it is not straightforward to extend the method to higher order time discretizations, while [32] is not exactly conservative at such interfaces. Our approach also allows for a more precise time advancing than the one in [4].

The structure of the paper is as follows. We will work on semidiscrete schemes, based mainly on [28] for the high order case and [17] for a second order scheme, although our framework can be generalized also to other finite volume schemes, such as Discontinuous Galerkin [3]. We start defining the numerical entropy production for semidiscrete schemes in Section 2; we prove the estimated rate of convergence and the sign of the entropy production in Section 3. Next we illustrate the properties of the indicator on a few test problems in Section 4. Finally we describe the adaptive algorithm in Section 5. Here we consider scheme adaptivity first, and next we build the 1D grid adaptive algorithm.

2 Semidiscrete Runge-Kutta schemes

From now on, consider a p^{th} order semidiscrete numerical scheme with a Runge-Kutta time advancement. Such a scheme for the conservation law (1.1) can be written in the form

$$\bar{U}_j^{(i)} = \bar{U}_j^n - \lambda \sum_{k=1}^{i-1} a_{ik} \left(F_{j+\frac{1}{2}}^{(k)} - F_{j-\frac{1}{2}}^{(k)} \right), \quad \bar{U}_j^{n+1} = \bar{U}_j^n - \lambda \sum_{i=1}^v b_i \left(F_{j+\frac{1}{2}}^{(i)} - F_{j-\frac{1}{2}}^{(i)} \right), \quad (2.1)$$

and it is built up from a Runge-Kutta method with Butcher's tableaux (A, b) and a suitable choice of numerical flux $F(U^-, U^+)$ consistent with the exact flux f appearing in (1.1), while $U_{j+1/2}^\pm$ are the boundary extrapolated data computed from the cell averages through an r^{th} order nonoscillatory reconstruction. As usual we denote $F_{j+1/2}^{(i)} = F(U_{j+1/2}^{(i),-}, U_{j+1/2}^{(i),+})$ the numerical flux evaluated at the i^{th} stage of the Runge-Kutta scheme.

Definition 2.1. The *numerical entropy production* in the control volume $V_j^n = [x_{j-1/2}, x_{j+1/2}] \times [t^n, t^{n+1}]$ is

$$S_j^n = \frac{1}{\Delta t} \left\{ \mathcal{Q}(\eta(U^{n+1}))_j - \mathcal{Q}(\eta(U^n))_j + \lambda \sum_{i=1}^v b_i \left(\Psi_{j+\frac{1}{2}}^{(i)} - \Psi_{j-\frac{1}{2}}^{(i)} \right) \right\}, \quad (2.2)$$

where $\Psi_{j+1/2}^{(i)} = \Psi(U_{j+1/2}^{(i),-}, U_{j+1/2}^{(i),+})$ is a numerical entropy flux $\Psi(U^-, U^+)$ consistent with the exact entropy flux ψ appearing in (1.2). In the formula, $\mathcal{Q}(\cdot)_j$ denotes a q^{th} order accurate quadrature rule applied on the interval I_j which uses L quadrature nodes $x_{l;j}$ and weights w_l , namely

$$\mathcal{Q}(f)_j = \frac{1}{h} \left(h \sum_{l=1}^L w_l f(x_{l;j}) \right) = \bar{f}_j + C_q h^q f^{(q)}(x_j) + \mathcal{O}(h^{q+1}).$$

Remark 2.1. We point out that the computation of S_j^n makes use of the same reconstructions that one must compute in order to advance the solution from \bar{U}_j^n to \bar{U}_j^{n+1} , so it does not involve a heavy computational overhead.

3 Properties

In this section we study the rate of convergence and the sign property for the entropy production, as defined in the previous section. The rate of convergence is proved for a general p^{th} order numerical scheme on smooth flows. The sign property is studied for a first order scheme, based on the Lax-Friedrichs flux splitting. A similar result can be proven, with the same technique, for the first order upwind scheme, when the propagation speed does not change sign. The estimate becomes harder for the general Godunov scheme using the entropy fix.

3.1 Rate of convergence

Proposition 3.1. Consider a p^{th} order convergent scheme of the form (2.1) and define the entropy production as in (2.2) with a consistent numerical entropy flux. Assume $\Delta t = \lambda h$, where λ is the mesh ratio. Then if the solution is regular, the numerical entropy production in a fixed time step decays as $\mathcal{O}(\Delta t^p)$ for $\Delta t \rightarrow 0$. In those cells in which the solution is not regular, $S_j \leq C/\Delta t$, for a constant C which does not depend on Δt .

Proof. Let $u(t, x)$ and $U(t, x)$ be the exact and the numerical solution of the conservation law. Assume that at time t^n the cell averages of the exact and the numerical solution coincide (localization assumption).

The cell averages of the entropy must be computed through quadrature; let x_l and w_l be the nodes and weights of the quadrature formula of order q , which must match the accuracy of the underlying scheme. Then

$$\mathcal{Q}(\eta(U^n))_j = \sum_{l=1}^L w_l \eta(U^n(x_{l;j})).$$

Dropping the subscript j , the values $U^n(x_l)$ in the quadrature nodes are computed from the r^{th} order reconstruction \mathcal{R} needed for the boundary extrapolated data, so that on smooth solutions we have

$$U^n(x_l) = \mathcal{R}(\bar{U}^n; x_l) = \mathcal{R}(\bar{u}^n; x_l) = u^n(x_l) + \mathcal{O}(h^{r+1}).$$

Analogously $\mathcal{Q}(\eta(U^{n+1}))_j = \sum_{l=1}^L w_l \eta(U^{n+1}(x_l))$, where

$$\begin{aligned} U^{n+1}(x_l) &= \mathcal{R}(\bar{U}^{n+1}; x_l) = \mathcal{R}(\bar{u}^{n+1}; x_l) + \mathcal{O}(\Delta t^{p+1}) \\ &= u^{n+1}(x_l) + \mathcal{O}(h^{r+1}) + \mathcal{O}(\Delta t^{p+1}). \end{aligned}$$

The time contribution is

$$\begin{aligned} \mathcal{Q}(\eta(U^{n+1}))_j - \mathcal{Q}(\eta(U^n))_j &= \mathcal{Q}(\eta(u^{n+1}))_j - \mathcal{Q}(\eta(u^n))_j \\ &+ \sum_{l=1}^L w_l [\eta'(u^{n+1}(x_l)) - \eta'(u^n(x_l))] \mathcal{O}(h^{r+1}) + \sum_{l=1}^L w_l \eta'(u^{n+1}(x_l)) \mathcal{O}(\Delta t^{p+1}). \end{aligned} \quad (3.1)$$

We note that

$$\eta'(u^{n+1}(x_l)) - \eta'(u^n(x_l)) = \mathcal{O}(\Delta t)$$

due to the smoothness of the exact solution. Further we apply the estimate for the quadrature formula

$$\begin{aligned} &\mathcal{Q}(\eta(U^{n+1}))_j - \mathcal{Q}(\eta(U^n))_j \\ &= \frac{1}{h} \int_{V_j^n} \partial_t \eta(u) dt dx - C_q h^q (\eta^{(q)}(u_j^{n+1}) - \eta^{(q)}(u_j^n)) + \mathcal{O}(\Delta t h^{r+1}) + \mathcal{O}(\Delta t^{p+1}) \\ &= \frac{1}{h} \int_{V_j^n} \partial_t \eta(u) dt dx + \Delta t \mathcal{O}(h^q + h^{r+1} + \Delta t^p). \end{aligned} \quad (3.2)$$

Write the space part as

$$\lambda \sum_{i=1}^v b_i (\Psi_{j+\frac{1}{2}}^{(i)} - \Psi_{j-\frac{1}{2}}^{(i)}) = \lambda \sum_{i=1}^v b_i \Delta_j \Psi^{(i)}.$$

Using the consistency of the numerical entropy flux with the exact entropy flux the difference of the entropy fluxes can be written as

$$\begin{aligned} \Delta_j \Psi^{(i)} &= \Psi(U_{j+\frac{1}{2}}^{(i),+}, U_{j+\frac{1}{2}}^{(i),-}) - \Psi(U_{j-\frac{1}{2}}^{(i),+}, U_{j-\frac{1}{2}}^{(i),-}) \\ &= \psi(U_{j+\frac{1}{2}}^{(i)}) - \psi(U_{j-\frac{1}{2}}^{(i)}) + \left[\nabla_u \Psi \left(\begin{matrix} U^{(i),+} & U^{(i),-} \\ U^{(i),-} & U^{(i)} \end{matrix} \right) \right]_{j+\frac{1}{2}} - \left[\nabla_u \Psi \left(\begin{matrix} U^{(i),+} & U^{(i),-} \\ U^{(i),-} & U^{(i)} \end{matrix} \right) \right]_{j-\frac{1}{2}}, \end{aligned}$$

where $U^{(i)}$ is an intermediate value between $U^{(i),+}$ and $U^{(i),-}$. Using the smoothness of the numerical entropy flux Ψ , the accuracy of the intermediate values $U^{(i),\pm}$ which relies on the accuracy of the underlying numerical scheme, we find that

$$\Delta_j \Psi^{(i)} = \int_{\Delta x} \partial_x \psi(U^{(i)}) dx + \mathcal{O}(h^{r+1}).$$

Finally, applying the time accuracy of the Runge-Kutta quadrature,

$$\lambda \sum_{i=1}^v b_i (\Psi_{j+\frac{1}{2}}^{(i)} - \Psi_{j-\frac{1}{2}}^{(i)}) = \frac{1}{h} \left[\int_{V_j} \partial_x \psi dt dx + \mathcal{O}(h \Delta t^{p+1}) \right] + \lambda \mathcal{O}(h^{r+1}). \quad (3.3)$$

Summing up the time and space contributions (3.2) and (3.3), we get

$$S_j^n = \frac{1}{\Delta t} \left\{ \frac{1}{h} \int_{V_j^n} (\partial_t \eta(u) + \partial_x \psi(u)) dt dx + \Delta t \mathcal{O}(\Delta t^p + h^{r+1} + h^q) \right\}.$$

If the reconstruction is computed with accuracy $r+1 \geq p$ and the quadrature has $q \geq p$, we find that $S_j^n = \mathcal{O}(\Delta t^p)$ for smooth solutions.

Note that in the proof, the accuracy of the underlying numerical scheme was used to assign to each stage value of the boundary extrapolated data the required accuracy.

Finally, for the second part of the proposition, it is sufficient to note that every term in Eq. (1.6) is at least bounded, so $|S_j| \leq C/\Delta t$, where C depends on the mesh ratio λ . For λ fixed, we have that S_j^n is bounded by $\mathcal{O}(1/h)$ terms. \square

Remark 3.1. In the proposition we have supposed that the underlying scheme is convergent, thus the argument of the proof can be repeated at each time step. The localization assumption ensures that the numerical entropy production measures the error introduced in each single time step.

Remark 3.2. For schemes up to second order, one may simply consider the midpoint quadrature rule and replace

$$\mathcal{Q}(\eta(U))_j \equiv \eta(\bar{U}_j) = \overline{\eta(U)}_j + \mathcal{O}(h^2)$$

in Definition 2.1.

3.2 Sign of the numerical entropy production

In this section, we will prove that the numerical entropy production built on the Lax Friedrichs scheme is negative definite on monotone profiles, provided that the numerical entropy flux is properly chosen. We can also prove that it is essentially negative definite on smooth flows in the presence of local extrema, meaning that $S_j^n \leq C\Delta t^q$, for some $q \geq p$. Thus local positive overshoots in the numerical entropy production are possible but their amplitude decreases at least as Δt^p .

In this section we consider the scalar case. For the Lax Friedrichs scheme, the numerical flux is:

$$F_{j+\frac{1}{2}}^n = F(U_{j+1}^n, U_j^n) = \frac{1}{2} [f(U_{j+1}^n) + f(U_j^n) - \alpha(U_{j+1}^n - U_j^n)]. \quad (3.4)$$

The coefficient α is the coefficient of artificial diffusion, chosen at each timestep so that it satisfies $|\alpha(U_j^n)| \leq \alpha \leq \lambda^{-1}$, $\forall j$, where $\alpha(U_j^n) = f'(U_j^n)$.

Proposition 3.2. Consider the first order Lax Friedrichs scheme, with the numerical flux (3.4). Define the numerical entropy flux as:

$$\Psi_{j+\frac{1}{2}}^n = \Psi(U_{j+1}^n, U_j^n) = \frac{1}{2} [\psi(U_{j+1}^n) + \psi(U_j^n) - \alpha(\eta(U_{j+1}^n) - \eta(U_j^n))]. \quad (3.5)$$

Suppose the sequence of grid points $(U_{j-1}^n, U_j^n, U_{j+1}^n)$ is monotone. Then the numerical entropy production (2.2), with the numerical entropy flux (3.5), is negative definite.

Proof. Since the reconstruction for the Lax Friedrichs scheme is piecewise constant, the numerical entropy production can be written as:

$$(\Delta t)S_j^n = \eta(U_j^{n+1}) - \eta(U_j^n) + \lambda[\Psi_{j+\frac{1}{2}}^n - \Psi_{j-\frac{1}{2}}^n].$$

Consider first the space part:

$$\begin{aligned} \Delta\Psi_j^n &= \lambda[\Psi_{j+\frac{1}{2}}^n - \Psi_{j-\frac{1}{2}}^n] \\ &= \frac{\lambda}{2} \left\{ \int_{U_j^n}^{U_{j+1}^n} [\psi'(u) - \alpha\eta'(u)] du + \int_{U_{j-1}^n}^{U_j^n} [\psi'(u) + \alpha\eta'(u)] du \right\} \\ &= \frac{\lambda}{2} \left\{ \int_{U_j^n}^{U_{j+1}^n} \eta'(u)[a(u) - \alpha] du + \int_{U_{j-1}^n}^{U_j^n} \eta'(u)[a(u) + \alpha] du \right\}, \end{aligned}$$

where we have used the compatibility condition $\psi'(u) = \eta'(u)a(u)$. Now consider the time part:

$$\begin{aligned} \Delta T_j^n &= \eta(U_j^{n+1}) - \eta(U_j^n) \\ &= \int_{U_j^n}^{U_j^{n+1}} \eta'(v) dv = \int_{U_j^n}^{U_j^*} \eta'(v) dv + \int_{U_j^*}^{U_j^{n+1}} \eta'(v) dv, \end{aligned}$$

where

$$U_j^* = U_j^n - \frac{\lambda}{2}[f_{j+1}^n - f_j^n - \alpha(U_{j+1}^n - U_j^n)],$$

so that

$$U_j^{n+1} = U_j^* - \frac{\lambda}{2}[f_j^n - f_{j-1}^n + \alpha(U_j^n - U_{j-1}^n)].$$

In the first integral, we change variables choosing

$$v_1(u) = U_j^n - \frac{\lambda}{2}[f(u) - f_j^n - \alpha(u - U_j^n)].$$

In the second integral, we set

$$v_2(u) = U_j^* - \frac{\lambda}{2}[f_j^n - f(u) + \alpha(U_j^n - u)].$$

The time integral becomes:

$$\Delta T_j^n = \frac{\lambda}{2} \left\{ \int_{U_j^n}^{U_{j+1}^n} \eta'(v_1)(\alpha - a(u)) du + \int_{U_j^n}^{U_{j-1}^n} \eta'(v_2)(\alpha + a(u)) du \right\}.$$

We can now drop the time dependence. Summing up the space and time contributions, we find:

$$\begin{aligned}\Delta t S_j &= \Delta T_j + \Delta \Psi_j \\ &= \frac{\lambda}{2} \left\{ \int_{U_j}^{U_{j+1}} [\eta'(v_1(u)) - \eta'(u)] (\alpha - a(u)) du \right. \\ &\quad \left. + \int_{U_{j-1}}^{U_j} [\eta'(u) - \eta'(v_2(u))] (\alpha + a(u)) du \right\} \\ &= \frac{\lambda}{2} \left\{ \int_{U_j}^{U_{j+1}} [v_1(u) - u] \eta''(\xi_1) (\alpha - a(u)) du \right. \\ &\quad \left. + \int_{U_{j-1}}^{U_j} [u - v_2(u)] \eta''(\xi_2) (\alpha + a(u)) du \right\},\end{aligned}$$

where ξ_1 and ξ_2 are intermediate points in the intervals enclosed by $v_1(u)$, u and by $v_2(u)$, u respectively. Substituting v_1 and v_2 , we find

$$\Delta t S_j = \frac{\lambda}{2} \int_{U_j}^{U_{j+1}} \left[u - U_j + \frac{\lambda}{2} (f(u) - f_j - \alpha(u - U_j)) \right] \eta''(\xi_1) (\alpha - a(u)) du \quad (\mathcal{I}_1)$$

$$+ \frac{\lambda}{2} \int_{U_{j-1}}^{U_j} \left[u - U_j - \frac{\lambda}{2} (f(u) - f_j + \alpha(u - U_j)) \right] \eta''(\xi_2) (\alpha + a(u)) du \quad (\mathcal{I}_2)$$

$$+ \frac{\lambda}{2} \int_{U_{j-1}}^{U_j} [U_j - U_j^*] \eta''(\xi_2) (\alpha + a(u)) du. \quad (\mathcal{I}_3)$$

Applying the mean value theorem again, we get:

$$\begin{aligned}\Delta t S_j &= \frac{\lambda}{2} \int_{U_j}^{U_{j+1}} -\frac{1}{2} [u - U_j] [(1 - \lambda \alpha) + (1 + \lambda a(\tilde{u}_1))] \eta''(\xi_1) (\alpha - a(u)) du \\ &\quad + \frac{\lambda}{2} \int_{U_{j-1}}^{U_j} \frac{1}{2} [u - U_j] [(1 - \lambda \alpha) + (1 - \lambda a(\tilde{u}_2))] \eta''(\xi_2) (\alpha + a(u)) du \\ &\quad + \frac{\lambda^2}{4} [f_{j+1} - f_j - \alpha(U_{j+1} - U_j)] \int_{U_{j-1}}^{U_j} \eta''(\xi_2) (\alpha + a(u)) du,\end{aligned}$$

where \tilde{u}_1 and \tilde{u}_2 are intermediate values obtained applying the mean value theorem to the functions $f(u) - f(U_j)$, $u \in (U_j, U_{j+1})$ and $f(u) - f(U_j)$, $u \in (U_{j-1}, U_j)$ respectively. The quantities

$$\frac{1}{2} [(1 - \lambda \alpha) + (1 \pm \lambda a(u))] \eta''(\alpha \mp a(u))$$

are positive, if CFL holds and $|a(u)| < \alpha$, so the sign of the first two integrals coincides with the sign of

$$\int_{U_j}^{U_{j+1}} -[u - U_j] du = -\frac{1}{2} (U_{j+1} - U_j)^2 < 0,$$

in the first case, while in the second case, the sign of the integral is given by the sign of

$$\int_{U_{j-1}}^{U_j} [u - U_j] du = -\frac{1}{2}(U_{j-1} - U_j)^2 < 0.$$

Finally for the last term, we observe that since $\eta''(\xi_2)$ is positive, the sign of the third term coincides with the sign of the following expression, where we note that $\int_{U_{j-1}}^{U_j} (\alpha + a(u)) du$ can be integrated exactly. Thus:

$$\begin{aligned} \text{sign}(\mathcal{I}_3) &= \text{sign}\{ [f_{j+1} - f_j - \alpha(U_{j+1} - U_j)] [f_j - f_{j-1} + \alpha(U_j - U_{j-1})] \} \\ &= -\text{sign}[U_{j+1} - U_j] (\alpha - a(\tilde{u}_1)) (\alpha + a(\tilde{u}_2)) [U_j - U_{j-1}] \\ &= -\text{sign}[U_{j+1} - U_j] [U_j - U_{j-1}]. \end{aligned}$$

Using again the fact that $(\alpha - a(\tilde{u}_1))$ and $(\alpha + a(\tilde{u}_2))$ are positive and the hypothesis on the monotonicity of the solution, we conclude that also $\mathcal{I}_3 \leq 0$. \square

Remark 3.3. Note that the first two integrals (\mathcal{I}_1) and (\mathcal{I}_2) are negative regardless of the shape of the initial profile. It is only the third term which may become positive, if an extremum occurs in the sequence U of grid values. Thus in the general case we will have to combine (\mathcal{I}_3) with (\mathcal{I}_1) and (\mathcal{I}_2) .

Proposition 3.3. Consider the first order Lax Friedrichs scheme, with the numerical flux (3.4). Define the numerical entropy flux as in (3.5). Suppose the sequence of grid points $(U_{j-1}^n, U_j^n, U_{j+1}^n)$ has a point of extremum. Then the numerical entropy production (1.6) is essentially negative definite on smooth flows.

Proof. To fix ideas, we will assume that the extremum occurring in $(U_{j-1}^n, U_j^n, U_{j+1}^n)$ is a local maximum. In particular, we will suppose that the point of extremum is such that $U_{j-1} \leq U_{j+1} \leq U_j$. Rewrite S_j from (\mathcal{I}_1) , (\mathcal{I}_2) and (\mathcal{I}_3) as:

$$\Delta t S_j = \frac{\lambda}{2} \int_{U_j}^{U_{j+1}} [a(u) - \alpha] \eta_1'' \left[\left(1 - \frac{\lambda\alpha}{2}\right)(u - U_j) + \frac{\lambda}{2}(f(u) - f_j) \right] \quad (\mathcal{J}_1)$$

$$+ \frac{\lambda}{2} \int_{U_{j+1}}^{U_j} [a(u) + \alpha] \eta_2'' \left[\left(1 - \frac{\lambda\alpha}{2}\right)(u - U_j) - \frac{\lambda}{2}(f(u) - f_j) \right] \quad (\mathcal{J}_2)$$

$$+ \frac{\lambda^2}{4} \int_{U_{j+1}}^{U_j} [a(u) + \alpha] \eta_2'' [f_{j+1} - f_j - \alpha(U_{j+1} - U_j)] \quad (\mathcal{J}_3)$$

$$+ \frac{\lambda}{2} \int_{U_{j-1}}^{U_{j+1}} [a(u) + \alpha] \eta_2'' \left[\left(1 - \frac{\lambda\alpha}{2}\right)(u - U_j) - \frac{\lambda}{2}(f(u) - f_j) \right] \quad (\mathcal{J}_4)$$

$$+ \frac{\lambda^2}{4} \int_{U_{j-1}}^{U_{j+1}} [a(u) + \alpha] \eta_2'' [f_{j+1} - f_j - \alpha(U_{j+1} - U_j)], \quad (\mathcal{J}_5)$$

where $\eta_i'' = \eta''(\xi_i)$, $i = 1, 2$. We will group together the terms (\mathcal{J}_1) , (\mathcal{J}_2) and (\mathcal{J}_3) , which actually contain the extremum, while (\mathcal{J}_4) and (\mathcal{J}_5) will be bounded separately.

Let $0 \leq m \leq \eta''(u) \leq M$ be the bounds of η'' over the interval (U_{j-1}, U_j) . We already know that (\mathcal{J}_3) is positive, thus it can be bounded as:

$$\begin{aligned}\mathcal{J}_3 &\leq M \frac{\lambda^2}{4} \int_{U_{j+1}}^{U_j} [a(u) + \alpha] [f_{j+1} - f_j - \alpha(U_{j+1} - U_j)] \\ &= -M \frac{\lambda^2}{4} [f_{j+1} - f_j - \alpha(U_{j+1} - U_j)] [f_{j+1} - f_j + \alpha(U_{j+1} - U_j)] \\ &= -M \frac{\lambda^2}{4} [(f_{j+1} - f_j)^2 - \alpha^2(U_{j+1} - U_j)^2].\end{aligned}$$

Now sum (\mathcal{J}_1) and (\mathcal{J}_2) , observing that, since they are both negative, η'' can be bounded by m to get:

$$\begin{aligned}\mathcal{J}_1 + \mathcal{J}_2 &\leq m \frac{\lambda}{2} \int_{U_{j+1}}^{U_j} [\alpha(2 - \lambda\alpha)(u - U_j) - \lambda a(u)(f(u) - f_j)] du \\ &= -\frac{m\lambda\alpha}{2} \left(1 - \frac{\lambda\alpha}{2}\right) (U_{j+1} - U_j)^2 + \frac{m\lambda^2}{4} (f_{j+1} - f_j)^2,\end{aligned}$$

where we have used the fact that the integrals can be evaluated exactly. Adding (\mathcal{J}_3) to the terms above, we have:

$$\mathcal{J}_1 + \mathcal{J}_2 + \mathcal{J}_3 \leq -\frac{m\lambda\alpha}{2} (\Delta U)^2 - \frac{\lambda^2}{4} (M - m) (\Delta F)^2 + \frac{\lambda^2\alpha^2}{4} (m + M) (\Delta U)^2,$$

where we used the notation $\Delta U = (U_{j+1} - U_j)$ and $\Delta F = (f_{j+1} - f_j)$. Now we write $\Delta F = \tilde{a}(U_{j+1} - U_j)$, where \tilde{a} denotes the function $a(\xi)$, evaluated in a point $\xi \in (U_{j+1} - U_j)$. Therefore:

$$\mathcal{J}_1 + \mathcal{J}_2 + \mathcal{J}_3 \leq \left[-\frac{m\lambda\alpha}{2} (1 - \lambda\alpha) + \frac{\lambda^2}{4} (M - m) (\alpha^2 - \tilde{a}^2) \right] (U_{j+1} - U_j)^2. \quad (3.6)$$

Note that $M > m$, $(\alpha^2 - \tilde{a}^2) \geq 0$ and $\lambda\alpha < 1$.

If η'' is constant, then $m = M$, and the entropy production due to these terms is negative. If η'' is not constant, then $M - m > 0$, and the last expression in the square brackets is non-negative. Note however that if the flow is smooth, then $m - M = \mathcal{O}(h^2)$ and $\Delta U^2 = \mathcal{O}(h^4)$ because the x derivative of U vanishes by assumption in the interval (x_{j-1}, x_{j+1}) and thus positive overshoots, if they occur, are extremely small on smooth flows. On the other hand, if the flow is not smooth, then $m - M$ and ΔU can be of order 1.

For the last two terms, we find:

$$\begin{aligned}\mathcal{J}_4 + \mathcal{J}_5 &= \frac{\lambda}{2} \int_{U_{j-1}}^{U_{j+1}} [a(u) + \alpha] \eta_2'' \left[u - U_j - \frac{\lambda\alpha}{2} (u - U_j) - \frac{\lambda\alpha}{2} (U_{j+1} - U_j) - \frac{\lambda}{2} (f(u) - f_{j+1}) \right] \\ &= \frac{\lambda}{2} \int_{U_{j-1}}^{U_{j+1}} [a(u) + \alpha] \eta_2'' \left[(1 - \lambda\alpha)(u - U_j) + \frac{\lambda}{2} (\alpha - \tilde{a})(u - U_{j+1}) \right].\end{aligned}$$

Here the integral is negative, since $(u - U_{j+1}) < 0$ and $(u - U_j) < 0$ for $u \in (U_{j-1}, U_{j+1})$ and $U_j > U_{j+1}$ under the hypothesis that U_j is a local maximum. \square

Remark 3.4. The previous proposition implies that the numerical entropy production is not bounded away from zero in cells containing extrema. This may underestimate the error in these cells when the numerical entropy production is used as an error indicator.

Remark 3.5. The same technique allows to prove that the sign of the entropy production is negative definite for the first order upwind scheme, under the restriction that f' does not change sign and that the CFL condition holds. In the upwind case, the entropy dissipation remains negative even on local extrema.

Remark 3.6. We have proved that the entropy production is negative for a few first order schemes in the scalar case. However we observe numerically that the entropy production due to first order monotone schemes is negative even in the case of systems of equations, such as the Euler system of compressible gas dynamics.

4 Numerical tests on properties of the entropy production

In this section, we show results on the rate of convergence of the entropy production for schemes from second to fifth order accuracy on scalar problems, a comparison with the Karni Kurganov error indicator on a 2D problem and a 1D shock tube problem, and the entropy production for a second order scheme on gas dynamics problems.

4.1 Rate of convergence

We compute the rate of convergence to zero of $|S_j|$ for a smooth flow and the maxima of $|S_j|$ across a shock wave and a contact discontinuity. We consider Burgers equation before and after shock formation, and a scalar linear advection problem on discontinuous data. For Burgers equation the initial data is $u(x, t=0) = 1 + \frac{1}{2} \sin(\pi x)$ in $[-1, 1]$ with periodic boundary conditions. The final times are $T = 0.3$ (before shock formation) and $T = 1.5$ (after shock formation). For the linear advection problem, the equation is $u_t + u_x = 0$ with initial condition

$$u(x, t=0) = \begin{cases} \cos\left(\frac{\pi}{2}x\right), & -1 < x < 0, \\ \sin(\pi x), & 0 \leq x \leq 1, \end{cases}$$

with periodic boundary conditions and final time $T = 1.5$.

The schemes tested are semidiscrete schemes with Runge-Kutta time advancement, and local Lax Friedrichs numerical flux. For the reconstruction, we consider a piecewise linear (SD2), a third order compact WENO reconstruction, [20] (SD3). For the time advancement SD2 implements the TVD second order Heun scheme, while SD3 is integrated in time with the third order TVD Runge-Kutta of [9]. Finally we consider fifth order WENO reconstructions, [28] with a fourth order (SD4) and a fifth order (SD5) Runge-Kutta scheme.

Table 1: Sup norm of $|S_j|$ on a smooth transition (at the final time) for schemes of order from 2 to 5, Burgers equation before shock formation.

	$N=20$	$N=40$	$N=80$	$N=160$	$N=320$	$N=640$
SD2	0.41144	0.996E-01	0.175E-01	0.350E-02	0.773E-03	0.179E-03
SD3	0.78220	0.524E-01	0.535E-02	0.692E-03	0.111E-03	0.158E-04
SD4	0.958E-01	0.142E-01	0.108E-02	0.637E-04	0.347E-05	0.196E-06
SD5	0.489E-01	0.997E-02	0.676E-03	0.271E-04	0.896E-06	0.259E-07

Table 2: Rate of convergence to zero for $|S_j|$ on a smooth transition for schemes of order from 2 to 5 (Burgers equation before shock formation).

	$N=20$	$N=40$	$N=80$	$N=160$	$N=320$
SD2	2.04	2.50	2.32	2.18	2.10
SD3	3.89	3.29	2.95	2.63	2.81
SD4	2.74	3.71	4.08	4.19	4.14
SD5	2.29	3.88	4.63	4.92	5.10

Table 3: Maximum entropy production on a shock for several schemes, under grid refinement. The cells with the maximum $|S_j|$ are those containing the shock (Burgers equation after shock formation).

	$N=20$	$N=40$	$N=80$	$N=160$	$N=320$	$N=640$
SD2	1.1131	2.1399	4.6215	9.0075	17.933	35.573
SD3	1.2336	2.6214	5.1332	10.182	20.316	40.602
SD4	1.1649	3.1572	4.7325	10.451	20.180	36.302
SD5	1.2333	2.3536	4.5667	9.0634	18.124	36.045

Table 4: Maximum entropy production on a contact discontinuity for several schemes, under grid refinement. The cells with the maximum $|S_j|$ are those containing the discontinuity.

	$N=20$	$N=40$	$N=80$	$N=160$	$N=320$	$N=640$
SD2	0.20367	0.24645	0.84641	0.48792	0.32028	0.25642
SD3	0.25252	0.32165	0.06848	0.07275	0.04870	0.03929
SD4	0.06423	0.06859	0.08822	0.06656	0.10110	0.12494
SD5	0.33048	0.14757	0.07947	0.05416	0.04986	0.05215

Tables 1 and 2 show the actual $\max_j |S_j^n|$ of the entropy production on Burgers' solution before shock formation and its rate of convergence to zero respectively, under grid refinement. The rate observed is consistent with the theory.

Table 3 contains the values of the maximum dissipation of entropy on a flow with a shock. Here the entropy doubles, as expected, as the mesh size is halved. On a contact discontinuity, the entropy is not as large (due also to the smearing of the profile induced by the scheme), and the entropy remains approximately constant under grid refinement, see Table 4. Note also that the orders of magnitude of these values of $|S_j|$ are much larger than in the smooth case. For this reason, the entropy production is an effective indicator to select discontinuities in the flow.

Moreover, comparing the data in Table 1 and those in Table 3 we see that the spurious entropy production on smooth flows decreases fast for high order schemes, and it is orders of magnitude smaller than in the shocked case. Thus the entropy production is an effective indicator especially for high order schemes.

4.2 A 2D example

We consider a simple 2D problem. A rectangular patch is rotated with constant angular velocity on the domain $[0,1] \times [0,1]$.

The equation is

$$\partial_t u + \partial_x \left[-\left(y - \frac{1}{2}\right) \frac{\pi}{2} u \right] + \partial_y \left[\left(x - \frac{1}{2}\right) \frac{\pi}{2} u \right] = 0,$$

with initial condition

$$u(x, y, t=0) = \begin{cases} 1.5, & (x, y) \in [0.3, 0.7] \times [0.2, 0.8], \\ 0.5, & \text{elsewhere.} \end{cases}$$

The underlying scheme is the second order SD2 method. Since the solutions exhibit contact discontinuities, limiters are needed to avoid spurious oscillations. Fig. 1 shows the solution with and without limiters. The second row shows the corresponding entropy production. It is clear that the entropy signals the presence of the discontinuities in both cases, but the strong positive overshoots in the case without limiters reveal also the presence of spurious oscillations. On the other hand the entropy production for the solution with limiters remains negative definite. We thus see an extension of Proposition 3.2 and 3.3 to the case of a nonsmooth problem for a high order scheme in two space dimensions.

The entropy production on the two solutions are clearly distinct. The last row shows for comparison the results obtained with the error indicator proposed by [14]. Here too the presence of the singularity is detected, but there is no clear way to distinguish the two solutions.

4.3 Gas-dynamics

We integrate the Euler equations for an ideal gas with the second order semidiscrete scheme SD2. The unknowns are $u = [\rho, \rho v, E]$, where ρ is the density, v the velocity and $E = \rho v^2/2 + \rho e$ is the total energy. The flux function is $f(u) = [\rho v, \rho v^2 + p, v(E + p)]$, where the equation of state $p = \rho e(\gamma - 1)$ links the internal energy and the pressure. In this paper we consider $\gamma = 1.4$, which is appropriate for air. For this system we consider the entropy pair defined by the entropy function $\eta(u) = -\rho \log(\rho e / \rho^\gamma)$ and the entropy flux $\psi(u) = -v\eta(u)$ (see [8]).

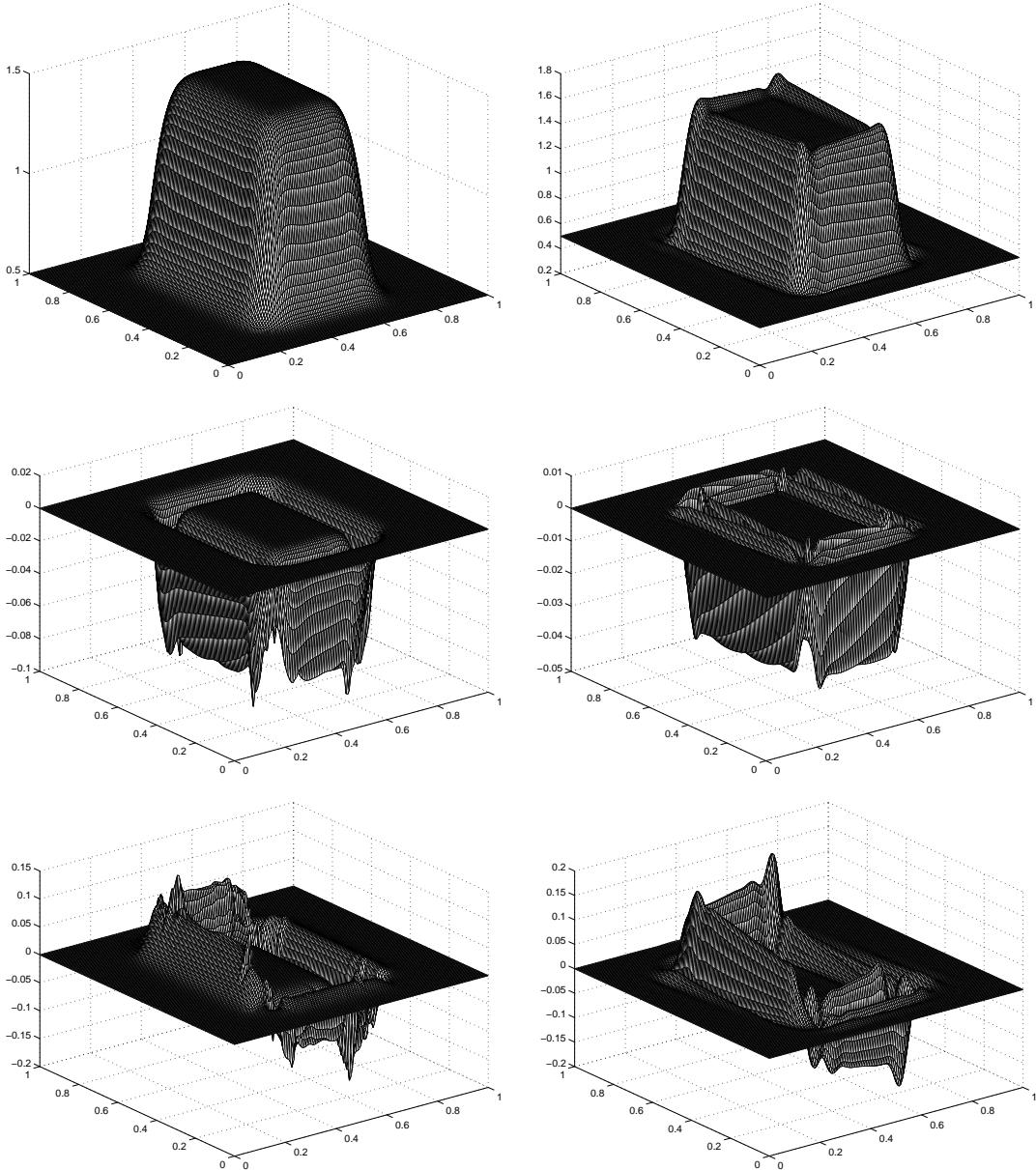


Figure 1: Rotation of a rectangular patch. Top: solution with (left) and without (right) limiters which results in spurious oscillations. Middle: corresponding entropy production. Bottom: corresponding Karni-Kurganov error indicator.

Left-moving rarefaction wave. We evolved the initial state

$$u = \begin{cases} u_L, & x \leq 0.8, \\ u_R, & x > 0.8, \end{cases} \quad \begin{bmatrix} \rho \\ v \\ p \end{bmatrix}_L = \begin{bmatrix} 1.02222 \\ -0.6179 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \rho \\ v \\ p \end{bmatrix}_R = \begin{bmatrix} \rho^* \\ v^* \\ 0.4 \end{bmatrix},$$

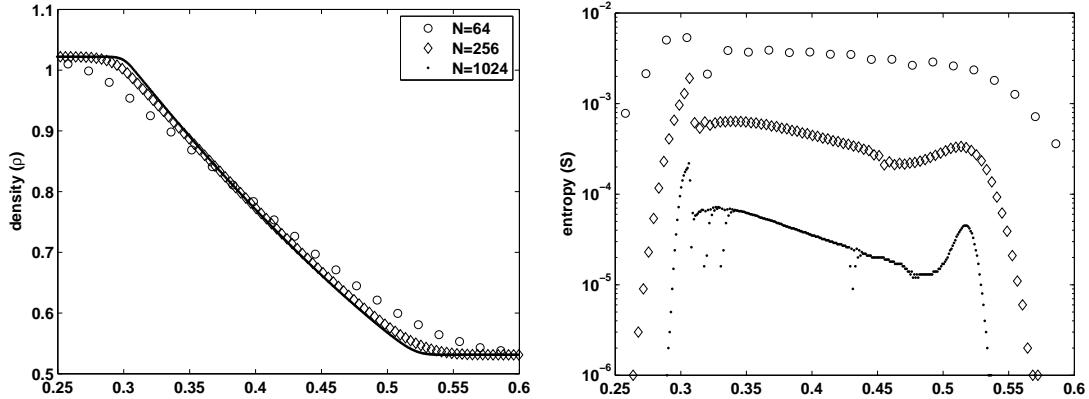


Figure 2: Euler equations, single left moving rarefaction wave. Solutions with different grid sizes (left) and entropy production in logarithmic scale (right).

where $\rho^* = \rho_L (p_R/p_L)^{1/\gamma}$, $v^* = v_L + 2/(\gamma-1)(a_R - a_L)$ and the sound speed is defined by $a = \sqrt{\gamma p/\rho}$. This Riemann problem gives rise to a single left-moving rarefaction wave that starts at $x=0.8$, which satisfies the CFL condition with $\lambda=0.5$. In Fig. 2 we show on the left a blow-up of the region occupied by the rarefaction at $t=0.25$. The right panel contains the corresponding entropy production, showing the improved resolution of the corners of the rarefaction and the convergence to zero of the spurious entropy production under grid refinement.

Right-moving contact discontinuity. Next we consider the initial data

$$u = \begin{cases} u_L, & x \leq 0.2, \\ u_R, & x > 0.2, \end{cases} \quad \begin{bmatrix} \rho \\ v \\ p \end{bmatrix}_L = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \rho \\ v \\ p \end{bmatrix}_R = \begin{bmatrix} 0.4 \\ 1 \\ 1 \end{bmatrix},$$

which give rise to a single right-moving contact discontinuity.

In Fig. 3 we show the solutions obtained at $t=0.25$, applying the SD2 scheme with $\lambda=0.25$. In the left panel we compare the solutions on different grids. The graph on the right clearly shows that the numerical entropy production over a contact wave is approximately constant across a wide range of cell sizes and it is localized on the cells crossed by the discontinuity. We performed the test with N from 32 up to 1024, but for clarity only a few plots are reported in the graph. Note that the entropy production has a completely different order of magnitude with respect to the case of the rarefaction wave, although the data have approximately the same magnitude.

Shock wave. In this test we set the initial data to

$$u = \begin{cases} u_L, & x \leq 0.2, \\ u_R, & x > 0.2, \end{cases} \quad \begin{bmatrix} \rho \\ v \\ p \end{bmatrix}_L = \begin{bmatrix} 1 \\ 0.8276 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} \rho \\ v \\ p \end{bmatrix}_R = \begin{bmatrix} 0.5313 \\ 0.1 \\ 0.4 \end{bmatrix}.$$

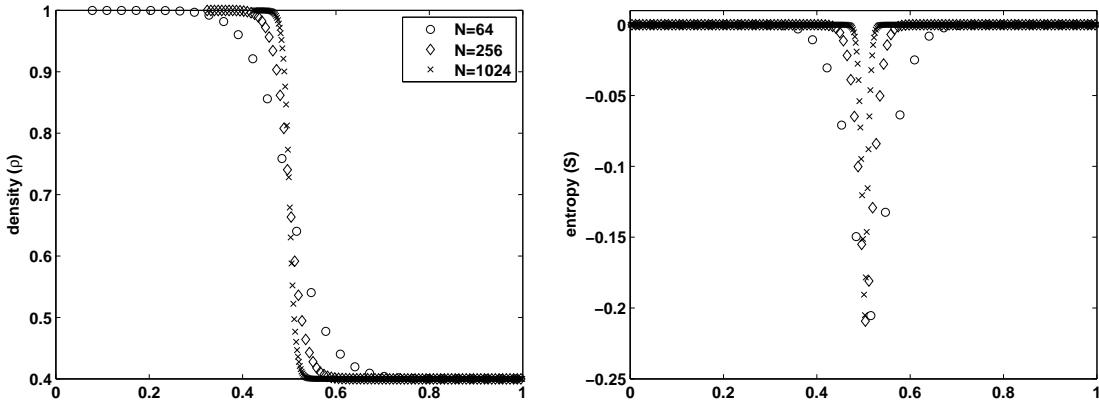


Figure 3: Euler equations, single right moving contact wave. Solutions with different grid sizes (left) and entropy production (right).

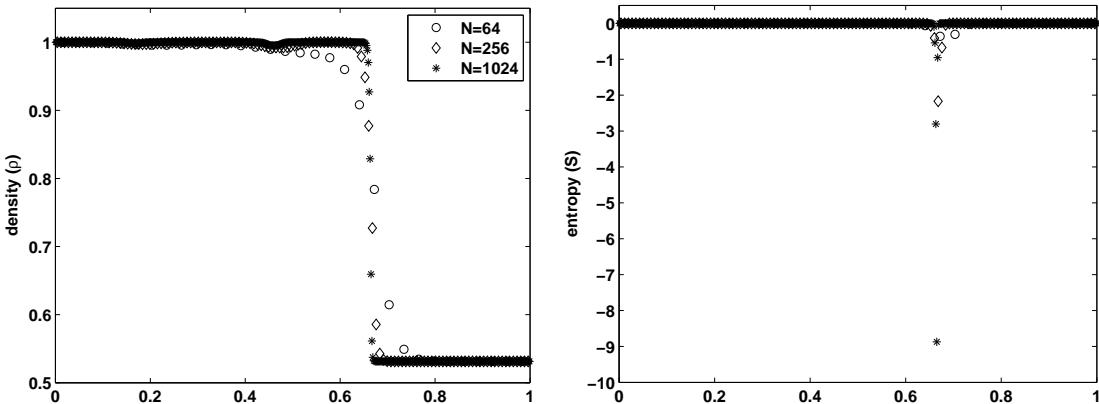


Figure 4: Euler equations, single right moving shock. Solutions with different grid sizes and entropy production.

This Riemann problem develops a right-moving shock wave, initially at $x = 0.2$.

As shown in Fig. 4, the entropy production on the shock increases as $1/h$ when the grid is refined. Note again the higher order of magnitude of the entropy production with respect to the previous cases.

Finally, we compare the entropy production on Lax' shock tube problem with the results obtained with Karni-Kurganov error indicator [14], see Fig. 5. The scheme is the third order SD3. Both indicators clearly separate the main structures of the flow, which can be easily identified comparing the plot of the indicator with the corresponding density profile in Fig. 13. However the Karni-Kurganov indicator for the third order scheme involves three time levels, as opposed to the entropy indicator that uses only two of them. This is clearly an important difference if one wants to modify locally the computation in response to the indicator values.

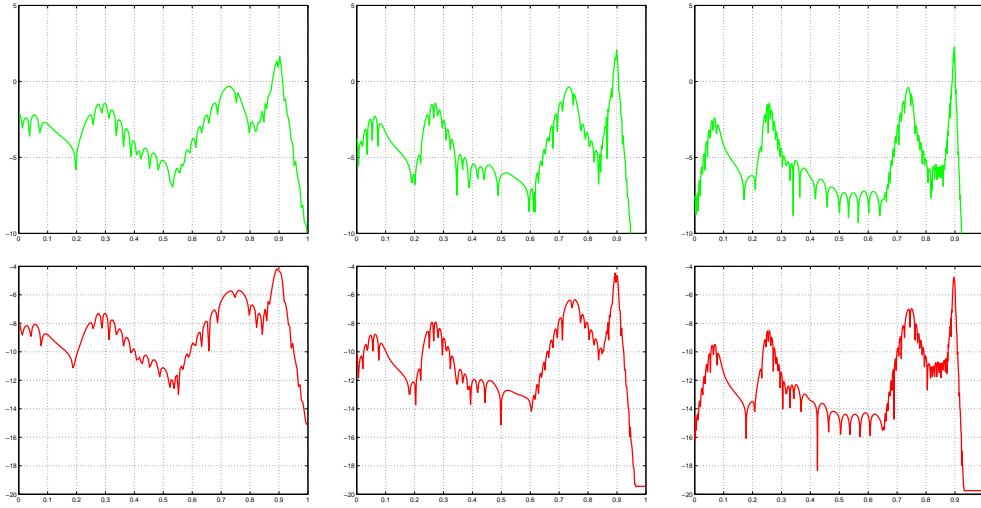


Figure 5: Lax' shock tube problem. Top: entropy production with $N = 200$, $N = 400$ and $N = 800$ grid points. Bottom: corresponding error indicator by Karni-Kurganov. Logarithmic scale.

5 Adaptivity

In this section, we exploit S_j^n as an indicator to locally modify the scheme and/or the grid to improve the accuracy of the solution.

We start with a description of a strategy to produce a non uniform diadic grid obtained by iterative halvening of a single cell initially occupying the whole computational domain. The entropy indicator is then used to refine and coarsen the computational cells in an adaptive mesh algorithm.

Start with a grid consisting of a single cell of width h_0 that may coincide with the whole domain. This cell is said to be of *level 0*. This may be split into two adjacent cells of level 1, of width $h_1 = h_0 2^{-1}$. Each of these may be further split into two cells of width $h_2 = h_0 2^{-2}$, and so on. Thus a diadic grid consists of cells with level $k \in \mathbb{N}$ and width $h_k = h_0 2^{-k}$. Note that we impose no limitation on the relative size of two neighbouring cells. Two cells that originated from the same mother cell are called *sister cells*. During the time integration we allow splitting any cell into two *daughter cells* or coarsen the grid by replacing two sister cells with their common *mother cell*. To this end, it is convenient to store the grid in a binary tree structure, thus retaining the information on the grid hierarchy. An example is shown in Fig. 6. There, the leaves of the tree, forming the active computational grid, are represented with black circles, while the other nodes are denoted with empty circles. The level 3 cells were obtained by refining the level $k = 2$ gray cell. On the other hand, coarsening the two sister cells of level 3 would replace both of them with the gray leaf and dashed link of their common mother cell of level 2.

At any time t^n , the solution is represented by the cell averages \bar{U}_j^n on the j^{th} cell, which has size $h_j = h_0 2^{-l_j}$ where l_j is the level of the cell.

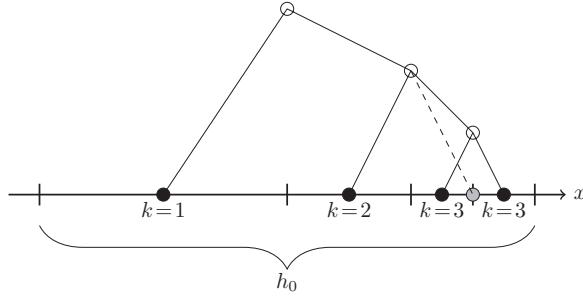


Figure 6: An example of diadic grid and its tree representation. On the rightmost cells, we show the tree representation of the refining/coarsening operation.

Time-stepping. We devised two different time-stepping procedures. In the first one, that we call *Δt -mode*, we choose a time step Δt valid for every cell. Thus, in order to satisfy the CFL condition we must choose the time step according to the stability requirement of the finest grid. Thus:

$$\Delta t = \lambda \min_j h_j = \lambda h_0 2^{-\max_j l_j}. \quad (5.1)$$

Hence the effective CFL number in each cell is

$$\frac{\lambda_j}{\lambda} = \frac{\Delta t}{\lambda h_j} = \frac{\min_i h_i}{h_j} = 2^{l_j - \max_i l_i}.$$

Note that, with this approach, it is important to use a semidiscrete scheme that does not lose accuracy at low CFL numbers. Moreover, even if there are a few small cells, one is forced to advance the solution with very small time steps, because the stability is dictated by the finest cells.

In order to avoid this problem, the second procedure, that we name *CFL-mode*, implements a local time-stepping procedure similar to the one described in [4]. It insists on using the same CFL number on each cell and thus the mesh ratio λ is constant throughout the grid. Thus the j^{th} cell advances with timestep of size

$$\Delta t_j = \lambda h_j = \lambda h_0 2^{-l_j}. \quad (5.2)$$

In a grid with cells of level $l_{\min} \leq l \leq l_{\max}$, the j^{th} cell must advance by $2^{l_j - l_{\min}}$ time steps of size Δt_j in order to match a single time advancement of the largest cells in the grid, which is $\Delta t = \Delta t_{\max} = \lambda h_{\max} = \lambda h_0 2^{-l_{\min}}$.

The procedure to advance the solution for a single global time step in *CFL-mode* is depicted in Fig. 7 for the case of a grid with cells of 3 different levels and a 1 stage Runge Kutta scheme. First (top left panel) compute reconstructions and fluxes (arrows) everywhere at time t^n . With these assemble $\bar{U}_j(t^n + \Delta t/4)$ in all small cells (solid circles) and their first neighbours (empty circles). Then (top right) compute reconstructions and fluxes at the boundaries of small cells at time $t^n + \Delta t/4$, and calculate $\bar{U}_j(t^n + \Delta t/2)$

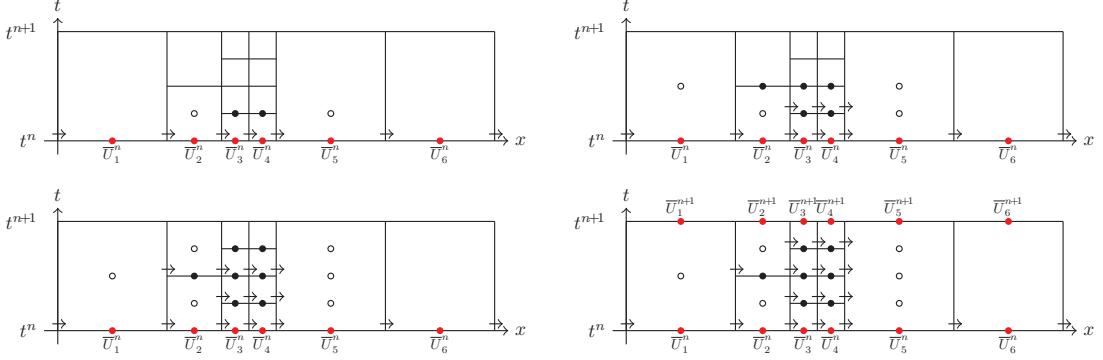


Figure 7: One timestep in *CFL-mode* for a grid with cells of 3 levels and a 1 stage Runge-Kutta.

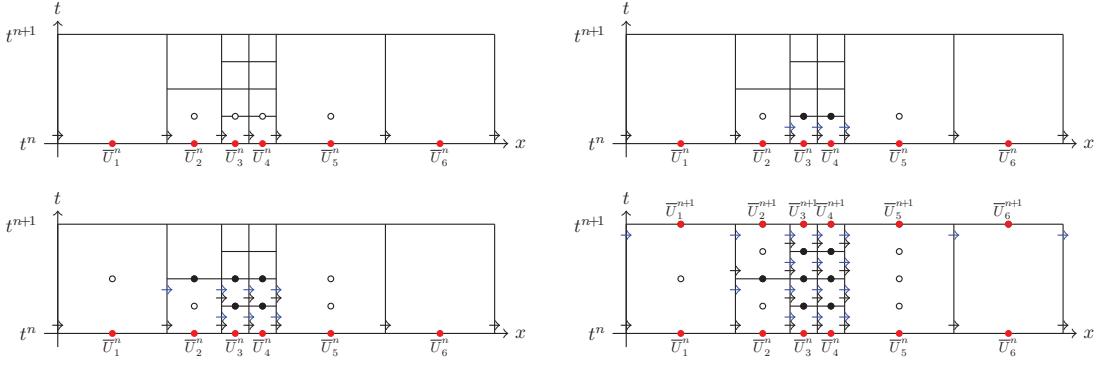


Figure 8: One timestep in *CFL-mode* for a grid with cells of 3 levels and a 2 stage Runge-Kutta.

on small and medium cells and their first neighbours. Next (bottom left) compute reconstructions and fluxes at the boundaries of small and medium cells at time $t^n + \Delta t/2$, and get $\bar{U}_j(t^n + 3\Delta t/4)$ on small cells and their first neighbours. Finally (bottom right) compute reconstructions and fluxes at the boundaries of small and medium cells at time $t^n + 3\Delta t/4$ and finally assemble \bar{U}_j^{n+1} on the whole grid.

Fig. 8 depicts the procedure when using a 2 stage Runge-Kutta. First (top left) compute the first stage fluxes $F_{j+1/2}^{(1)}$ on all boundaries (black arrows) using the reconstructions computed from \bar{U}_j^n and use them to obtain $\bar{U}_j^{(2)}(t^n + \Delta t/4)$ on small cells and first neighbours (empty circles). Reconstructing boundary values from $\bar{U}_j^{(2)}(t^n + \Delta t/4)$, compute (top right) the second stage fluxes $F_{j+1/2}^{(2)}$ (blue arrows) and update $\bar{U}_j(t^n + \Delta t/4)$ on small cells and first neighbours (solid circles and empty circles). Similarly (bottom left) compute $\bar{U}_j(t^n + \Delta t/4)$ on small, medium cells and their first neighbours. Repeating the above procedure (bottom right) get $\bar{U}_j^{(2)}(t^{n+1})$ and with a final reconstruction obtain the computation of $F^{(2)}$ everywhere and $\bar{U}_j^{n+1} = \bar{U}_j(t^{n+1})$ on every cell.

Remark 5.1. The procedure illustrated ensures that the fluxes are functions of the cell boundary and thus no mass is created at cell interfaces. This is the main difference with respect to the scheme [32] and ensures conservation without losing neither consistency nor accuracy.

This procedure differs from the one in [4] due to the computation of the auxiliary values in the neighbouring cells (empty circles in Figs. 7 and 8): the strategy proposed here is more accurate and does not require a smoothly varying cell size h_j .

Indicator and h -adaptivity. The entropy indicator is computed at the end of a complete time step Δt , both in the case of the *CFL-mode* and the Δt -mode. At this stage, the grid is modified following the behaviour of the indicator. In particular, during the time integration, we compute not only the fluxes $F_{j+1/2}^{(\cdot)}(t^n + c_k \Delta t)$ but also the numerical entropy fluxes $\Psi_{j+1/2}^{(\cdot)}(t^n + c_k \Delta t)$, exploiting the same boundary value reconstructions that were computed for updating the solution. When time integration reaches t^{n+1} we have also all the information to approximate $\int_{t^n}^{t^{n+1}} \psi(t, x_{j+1/2}) dt$ using the numerical entropy fluxes and thus we can compute S_j^n using formula (2.2).

Now we can compare the numerical entropy production in each computational cell with a threshold S_{ref} and mark cells for refinement. We then split the marked cells into two halves and initialize them setting \bar{U}_j^n equal to the value of the mother cell. (Alternatively, one may set \bar{U}_j^n on the daughter cells using the reconstruction computed at time level t^n). Then only the values \bar{U}_j^{n+1} affected by this change of initial data are recomputed and another check on the entropy indicator is performed. This procedure is iterated until no more cells need refining. In order to prevent an infinite refining loop around shocks, we impose a lower bound h_{\min} on the cell size and do not refine further when this minimum is reached, regardless of the entropy production.

Before proceeding to the next time step, we perform a coarsening pass on the grid: if the sum of the entropy production on two sister cells is lower than the coarsening threshold S_{coa} , the cells are merged and replaced with their common mother cell and \bar{U}_j^{n+1} is set to the average of the values of the merged cells. In order to avoid coarsening and refining the same cells in subsequent time steps, we choose the coarsening threshold lower than the refining threshold. In the numerical tests we employed $S_{\text{coa}} = \frac{1}{4}S_{\text{ref}}$.

Remark 5.2. Shocks are well resolved by finite volume schemes, and therefore on shocks strong refinement is not necessary. When refinement occurs, one could store the values of the entropy indicator before and after refinement. If these values show that entropy increases under grid refinement, than the corresponding cell can be labelled as containing a shock. In this case, refinement is not carried further.

The ability of recognizing shocked cells is an added value of the entropy indicator and it can be used to prevent excessive and needless refinement.

5.1 Scheme adaptivity

Here we consider a simple example of scheme adaptivity. It is well known that the application of a high order scheme in a componentwise fashion may result in the production of spurious oscillations, whose amplitude diminishes under grid refinement. These oscillations can be avoided if the reconstructions needed to compute the numerical fluxes are carried out along characteristic directions, see [27]. However this procedure is costly and computationally demanding. In this paragraph, we compute the characteristic projection only in those cells where the production of entropy signals a non smooth flow. The adaptive strategy is

$$\begin{cases} \text{if } |S_j| > S_{\text{ref}}, & \text{use characteristic projection,} \\ \text{else,} & \text{use component-wise reconstruction,} \end{cases}$$

with $S_{\text{ref}} = 1$.

Fig. 9 shows the results. The top row contains the solutions obtained with the third order SD3 scheme on Lax' Riemann problem, for several grid sizes. The figure shows a detail of the density peak, with the contact discontinuity on the left followed by the shock. The whole density profile can be seen in Fig. 13. The most oscillatory blue solution is obtained with the componentwise application of the reconstruction in all cells. Spurious oscillations are clearly visible, especially on the coarse grid. The remaining profiles are obtained applying the characteristic projection in all cells and with the adaptive strategy, respectively. This latter is slightly less diffusive than the one obtained with the characteristic projection. The second row of the figure contains the number of cells on which the characteristic projection was actually used by the adaptive strategy, as a function of time, for the three grids used in this test, namely, $N = 100$, $N = 200$ and $N = 400$. Note that the number of points selected remains approximately constant in time, after the waves separate, and it is almost fixed under grid refinement.

5.2 Adaptive grid tests

We start assessing the quality of the time advancing scheme on non uniform grids, both in *Δt -mode* and *CFL-mode*. In particular, in this section we will consider the second order semidiscrete scheme of [17]. First, we wish to show that the accuracy of the scheme is not modified by the presence of a non uniform grid, and that the entropy indicator maintains the expected rate of convergence. The aim of the first tests is to assess the robustness of the scheme and of the indicator as the solution crosses a discontinuity in the mesh spacing, before proceeding to the development of the fully adaptive and unsteady algorithm.

In these tests we compare the solution obtained on a regular grid, with the solution obtained with a non uniform grid, where the mesh size is a step function, namely:

$$h_j = \begin{cases} h, & x_j < \frac{1}{2}, \\ \frac{h}{2}, & x_j \geq \frac{1}{2}. \end{cases}$$

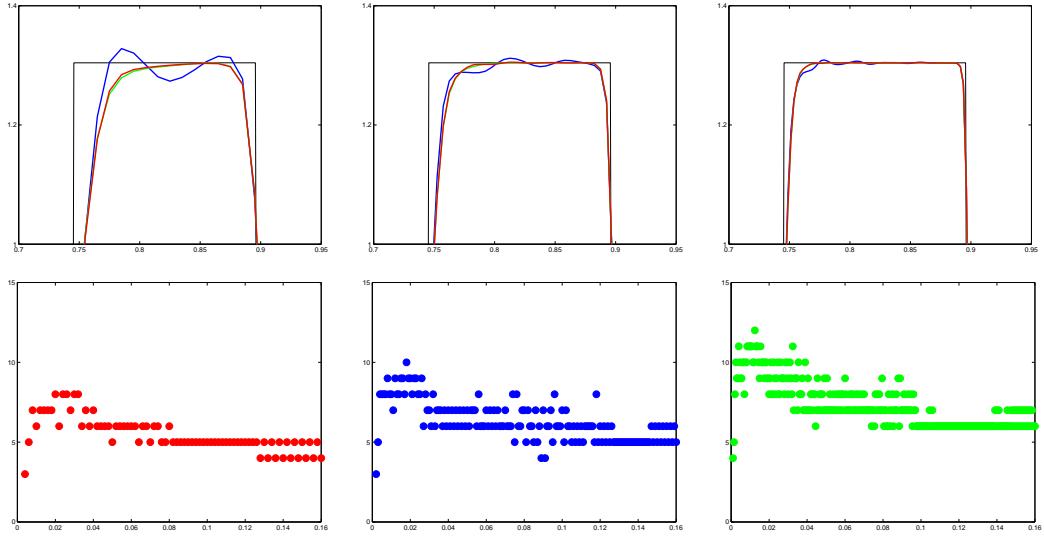


Figure 9: Adaptive application of characteristic projection. Top: detail of the density peak at the final time, with increasing number of cells from left to right. Bottom: number of cells selected for characteristic projection versus time.

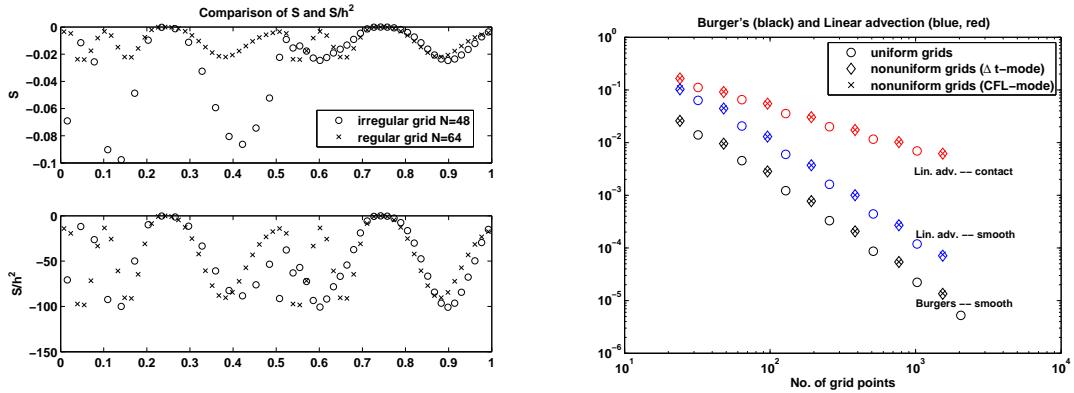


Figure 10: Left: entropy indicator for the transport of a sinusoidal wave. Right: L_1 norm of the error on 3 tests with uniform meshes and non-uniform meshes.

We consider the initial condition $u_0(x) = \sin(2\pi x)$ on $[0,1]$ with periodic boundary conditions for the linear advection equation (left panel of Fig. 10) and in Fig. 11 the linear advection of a solution containing an angular point and a contact discontinuity (on the left) and finally a double-step profile evolving with Burgers' equation (on the right).

For the regular profile (left panel of Fig. 10) we compare the entropy production in the last time step on a regular grid with 64 cells and an irregular grid with $h = 1/32$, i.e., 48 cells: 16 larger cells in $[0,0.5]$ and 32 cells on $[0.5,1]$. The top graph is S_j and the bottom graph shows the normalized quantity S_j/h_j^2 . As expected, the entropy scales as h^2 .

The right panel of Fig. 10 contains, on a logarithmic scale, the history of convergence

of the L^1 norm of the error for two smooth problems initialized with a sine wave (linear advection and Burgers equation with $u_0(x) = \sin(2\pi x)$ before shock formation). For comparison we also add the L^1 norm of the error for a linear advection problem with a contact discontinuity (red markers). For each test we plot the errors when using uniform meshes (circles) and the nonuniform meshes described above (diamonds for the Δt -mode and crosses for the CFL-mode). The series corresponding to each test are distinguished by the color of the markers and the labels in the graph. As expected, the convergence is of second order in the smooth cases (blue and black) and of first order when a discontinuity is present in the solution (red). It is important to note that the irregularity of the grid does not influence the rate of convergence of the error: in fact the symbols for each test appear perfectly aligned in the graphs. Moreover, note that the errors when using the Δt -mode and the CFL-mode are almost identical. This is a strong point in favour of the CFL-mode that is computationally much less expensive when cells of very different sizes are present.

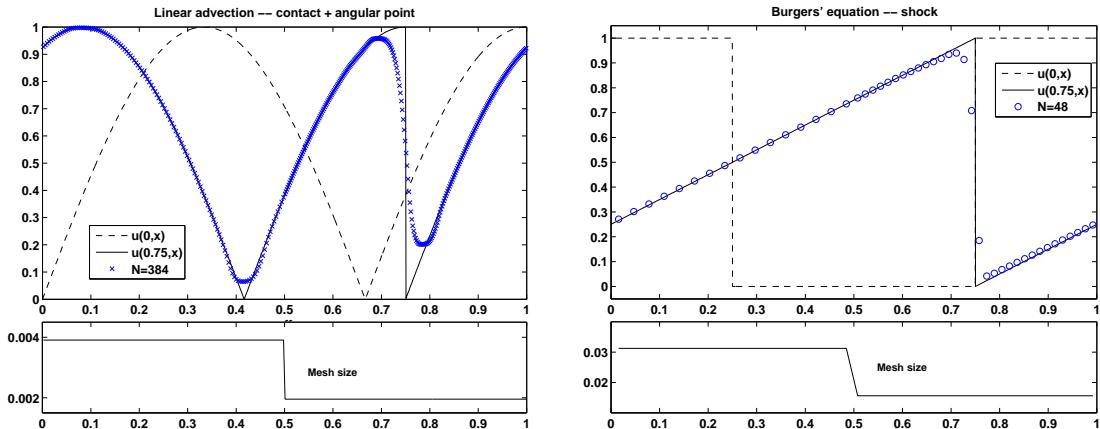


Figure 11: Solutions on non uniform meshes. Left: linear advection of a profile with a corner and a jump discontinuity. Right: shock on Burgers' equation.

Fig. 11 demonstrates that also contact discontinuities and shocks can traverse a discontinuity in the mesh-size without being reflected or deformed. Both panels show the initial datum (dashed line), the exact solution (solid line) and the numerical solution (markers). The bottom graph shows the mesh size function. On the left, the initial profile $u_0(x) = |\sin(3/2\pi x)|$ was evolved in the domain $x \in [0,1]$ with periodic boundary conditions until $t=1$, so that the contact discontinuity moves from the initial position in $x=0$ to $x=3/4$. In the right panel we consider the case of Burgers' equation with a double step initial condition integrated until $t=1$; the shock initially at $x=1/4$ moves to $x=3/4$ and the corner of the rarefaction that starts at $x=3/4$ has reached the shock when $t=1$. Note that no deformation of the profiles has occurred when the discontinuities crossed the jump in cell sizes located at $x=1/2$. We obtain the same results both in the CFL-mode and in the Δt -mode.

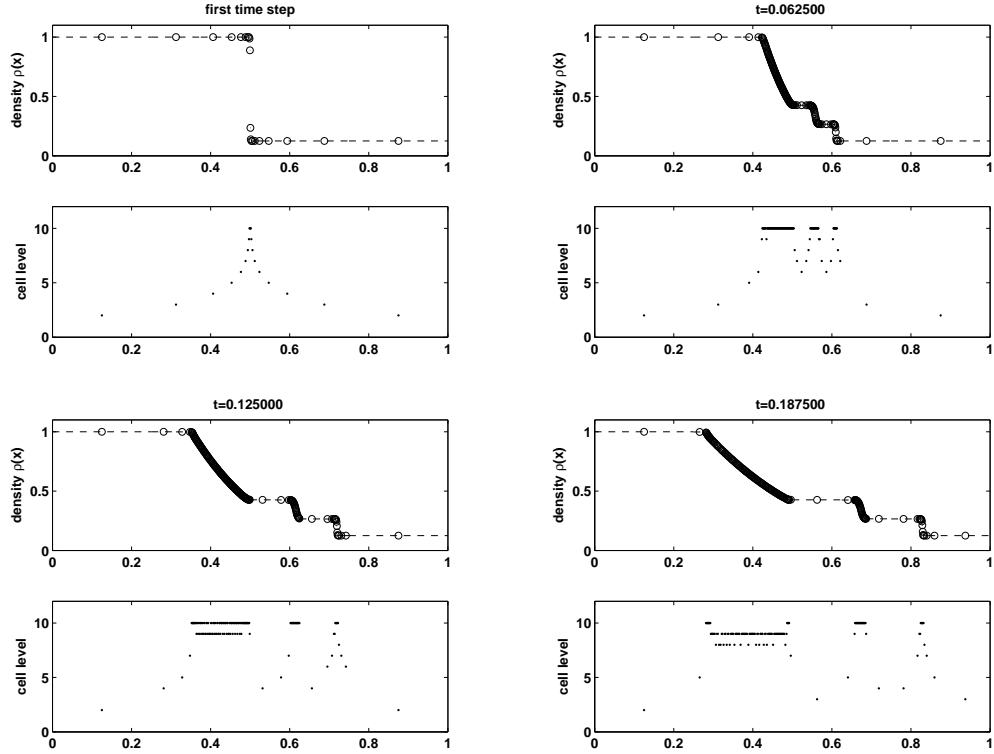


Figure 12: Sod problem with adaptive grid. Initial and minimum cell refinement level is 1, the maximum level is 10, i.e., $h_{\min} = 2^{-10}$. $S_{\text{ref}} = 10^{-3}$ was used in this run.

We now study the performance of the adaptive grid algorithm, considering the Euler equations of gas dynamics. We show plots obtained with the grid evolving in time in Δt -mode, but very similar results are obtained in CFL -mode.

Fig. 12 depicts the time evolution of the grid and of the density ρ of the gas for the Sod shock tube problem. The figure shows the solution and the corresponding grid at four different times. The density is shown with circles, while the grid levels are drawn with stars. For this test we set up an initial grid consisting of only 2 cells, the minimum needed to represent the initial data. During the first timestep (top left panel), the adaptive algorithm refines the grid until the cells around $x = 0.5$ are drawn to the minimum mesh size $h = 2^{-l_{\max}} = 2^{-11}$. Later on in the evolution the rarefaction, contact and shock separate and we can see that the algorithm starts coarsening the grid first on flat areas (top right) and then also inside the rarefaction (bottom panels). Note also that the areas with refined cells move along the x -axis following the features of the solution.

Finally we consider the Lax problem, setting up the grid as in the Sod test. At final time ($t = 0.16$), the adaptive grid consists of 374 cells with refinement level up to 11. In Fig. 13 we compare the solution obtained with the adaptive algorithm and the density computed with two uniform grids with 256 and 512 cells respectively. The grid for the

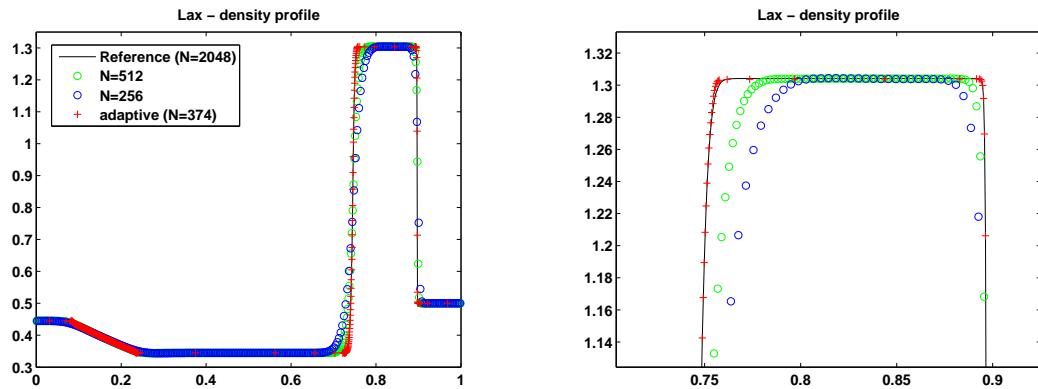


Figure 13: Lax problem with adaptive grid. Initial and minimum cell refinement level is 1, the maximum level is 11., i.e., $h_{\min} = 2^{-11}$. $S_{\text{ref}} = 10^{-3}$ was used in this run.

reference solution has 2048 cells of level 11. We show the complete density profile (left) and a zoom on the contact discontinuity and the shock (right). The solution given by the adaptive grid has an even better resolution than the reference profile obtained with more than 5 times as many points. The profiles obtained with a comparable number of grid points perform much more poorly than the adaptive solution on all important features of the flow. Moreover the adaptive algorithm achieves its resolution by selecting small cells only around the troublesome features of the solution, coarsening as much as is allowed by the dyadic grid structure in the other areas.

6 Conclusions

In this work, we have proposed an a posteriori error/smoothness indicator based on the numerical production of entropy to estimate the local error for finite volume schemes integrating systems of conservation laws. The indicator can be built on an underlying finite volume scheme, provided a consistent numerical entropy flux is given. In the paper, we have proposed how to define such a numerical entropy flux for several semidiscrete finite volume schemes.

We have proved that, on a smooth flow, the indicator decays with the same rate of the local truncation error of the scheme. Moreover, we have proved that for standard first order schemes the numerical production of entropy as defined here is essentially negative definite for scalar conservation laws.

Further, we propose an algorithm to construct non uniform time advancement on non uniform one dimensional grids. The algorithm proposed has a simple data structure, it is consistent and conservative. A few test problems on the second order scheme of [17] show that accuracy is maintained even on non uniform grids, and no spurious effects such as ghost reflection or diffraction at discontinuities in the grid spacing seem to appear.

Finally, the entropy indicator and the second order scheme [17] on non uniform grids are coupled, obtaining an adaptive algorithm in which the grid is locally refined or coarsened according to the behavior of the entropy indicator. The tests included show an impressive improvement in resolution with respect to uniform grids, even with a much larger number of nodes.

We plan to extend these results to construct a third order adaptive algorithm with a compact stencil, following the reconstruction in [20]. We also plan to extend the results of this paper to the two-dimensional setting. Preliminary results using adaptive cartesian meshes with the Δt -mode for two-dimensional gas-dynamics problems with cells of three possible sizes have already appeared in [21]. Future work will be dedicated to implement more levels of grid refinement and use the more efficient CFL -mode time advancement algorithm. Moreover, we plan to apply these ideas to other systems of conservation laws with an entropy, such as MHD.

References

- [1] M. J. Berger and R. J. Leveque, Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems, *SIAM J. Numer. Anal.*, 35(6) (1998), 2298–2316.
- [2] G. Chiavassa and R. Donat, Point value multiscale algorithms for 2D compressible flows, *SIAM J. Sci. Comput.*, 23(3) (2001), 805–823.
- [3] B. Cockburn, An introduction to the discontinuous Galerkin method for convection-dominated problems, in *Advanced numerical approximation of nonlinear hyperbolic equations* (Cetraro, 1997), Volume 1697, Lect. Notes. Math., 151–268, Springer, Berlin, 1998.
- [4] C. Dawson and R. Kirby, High resolution schemes for conservation laws with locally varying time steps, *SIAM J. Sci. Comput.*, 22(6) (2001), 2256–2281.
- [5] A. Dedner, C. Makridakis and M. Ohlberger, Error control for a class of Runge-Kutta Discontinuous Galerkin methods for non linear conservation laws, *SIAM J. Numer. Anal.*, 45 (2007), 514–538.
- [6] U. Fjordholm, S. Mishra and E. Tadmor, Arbitrarily high-order accurate entropy stable essentially non-oscillatory schemes for systems of conservation laws, SAM Report 2001-39, ETH Zurich, 2011.
- [7] M. B. Giles and E. Süli, Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality, *Acta. Numer.*, 11 (2002), 145–236.
- [8] E. Godlewski and P. A. Raviart, *Numerical approximation of hyperbolic systems of conservation laws*, Volume 118, Applied Mathematical Sciences, Springer-Verlag, New York, 1996.
- [9] S. Gottlieb, C.-W. Shu and E. Tadmor, Strong stability-preserving high-order time discretization methods, *SIAM Rev.*, 43(1) (2001), 89–112.
- [10] J.-L. Guermond, R. Pasquetti and B. Popov, Entropy viscosity method for nonlinear conservation laws, *J. Comput. Phys.*, 230(11) (2011), 4248–4267.
- [11] A. Harten, The artificial compression method for computation of shocks and contact discontinuities I: single conservation laws, *Commun. Pure Appl. Math.*, 30(5) (1977), 611–638.
- [12] A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, *Commun. Pure Appl. Math.*, 48(12) (1995), 1305–1342.
- [13] R. Hartmann and P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, *J. Comput. Phys.*, 183(2) (2002), 508–532.

- [14] S. Karni and A. Kurganov, Local error analysis for approximate solutions of hyperbolic conservation laws, *Adv. Comput. Math.*, 22(1) (2005), 79–99.
- [15] S. Karni, A. Kurganov and G. Petrova, A smoothness indicator for adaptive algorithms for hyperbolic systems, *J. Comput. Phys.*, 178(2) (2002), 323–341.
- [16] D. Kröner and M. Ohlberger, A posteriori error estimates for upwind finite volume schemes for nonlinear conservation laws in multidimensions, *Math. Comput.*, 69(229) (2000), 25–39.
- [17] A. Kurganov and E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations, *J. Comput. Phys.*, 160(1) (2000), 241–282.
- [18] R. J. LeVeque, Numerical methods for conservation laws, *Lectures in Mathematics ETH Zürich*. Birkhäuser Verlag, Basel, second edition, 1992.
- [19] R. J. LeVeque, CLAWPACK Version 4.3 Users Guide, 2006. <http://www.amath.washington.edu/~claw/>.
- [20] D. Levy, G. Puppo and G. Russo, Compact central WENO schemes for multidimensional conservation laws, *SIAM J. Sci. Comput.*, 22(2) (2000), 656–672.
- [21] D. Marobin and G. Puppo, An Error Indicator for Semidiscrete Schemes, Springer, editor, *Computational Fluid Dynamics 2004*, pages 103–108, 2006.
- [22] S. Müller and Y. Stiriba, Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping, *J. Sci. Comput.*, 30(3) (2007), 493–531.
- [23] M. Ohlberger, A review of a posteriori error control and adaptivity for approximations of non-linear conservation laws, *Int. J. Numer. Methods Fluids*, 59(3) (2009), 333–354.
- [24] S. Osher and R. Sanders, Numerical approximations to nonlinear conservation laws with locally varying time and space grids, *Math. Comput.*, 41(164) (1983), 321–336.
- [25] G. Puppo, Numerical entropy production on shocks and smooth transitions, in *Proceedings of the Fifth International Conference on Spectral and High Order Methods (ICOSAHOM-01)* (Uppsala), 17 (2002), 263–271.
- [26] G. Puppo, Numerical entropy production for central schemes, *SIAM J. Sci. Comput.*, 25(4) (2003), 1382–1415.
- [27] J. Qiu and C.-W. Shu, On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes, *J. Comput. Phys.*, 183(1) (2002), 187–209.
- [28] C.-W. Shu, High order ENO and WENO schemes for computational fluid dynamics, In *High-order methods for computational physics*, Volume 9, *Lect. Notes Comput. Sci. Eng.*, pages 439–582, Springer, Berlin, 1999.
- [29] C.-W. Shu and S. Osher, Efficient implementation of essentially nonoscillatory shock-capturing schemes, *J. Comput. Phys.*, 77(2) (1988), 439–471.
- [30] E. Tadmor, The numerical viscosity of entropy stable schemes for systems of conservation laws I, *Math. Comput.*, 49(179) (1987), 91–103.
- [31] E. Tadmor, Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems, *Acta. Numer.*, 12 (2003), 451–512.
- [32] H. Z. Tang and G. Warnecke, High resolution schemes for conservation laws and convection-diffusion equations with varying time and space grids, *J. Comput. Math.*, 24(2) (2006), 121–140.
- [33] R. Verfürth, *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Teubner Verlag and J. Wiley, Stuttgart, 1996.