Cent. Eur. J. Math. • 11(8) • 2013 • 1392-1415
DOI: 10.2478/s11533-013-0252-6

VERSITA

Central European Journal of **Mathematics**

Adaptive multiscale scheme based on numerical density of entropy production for conservation laws

Research Article

Mehmet Ersoy[1*], Frédéric Golay[1†], Lyudmyla Yushchenko[1‡]

1 IMATH, EA 2134, Université de Toulon, 83957 La Garde Cedex, France

Received 28 March 2012; accepted 9 January 2013

Abstract: We propose a 1D adaptive numerical scheme for hyperbolic conservation laws based on the numerical density of entropy production (the amount of violation of the theoretical entropy inequality). This density is used as an *a posteriori* error which provides information if the mesh should be refined in the regions where discontinuities occur or coarsened in the regions where the solution remains smooth. As due to the Courant–Friedrich–Levy stability condition the time step is restricted and leads to time consuming simulations, we propose a local time stepping algorithm. We also use high order time extensions applying the Adams–Bashforth time integration technique as well as the second order linear reconstruction in space. We numerically investigate the efficiency of the scheme through several test cases: Sod's shock tube problem, Lax's shock tube problem and the Shu–Osher test problem.

MSC: 74S10, 35L60, 74G15

Keywords: Hyperbolic system • Finite volume scheme • Local mesh refinement • Numerical density of entropy production • Local time stepping

1. Introduction

We are interested in the numerical integration of the following one-dimensional nonlinear hyperbolic system:

$$\begin{cases} \dfrac{\partial w}{\partial t} + \dfrac{\partial f(w)}{\partial x} = 0, & (t,x) \in \mathbb{R}^+ \times \mathbb{R}, \\ w(0,x) = w_0(x), & x \in \mathbb{R}, \end{cases} \tag{1}$$

* E-mail: Mehmet.Ersoy@univ-tln.fr
† E-mail: Frederic.Golay@univ-tln.fr
‡ E-mail: Lyudmyla.Yushchenko@univ-tln.fr

where $\boldsymbol{w}\colon \mathbb{R}^+ \times \mathbb{R} \to \mathbb{R}^d$ stands for the vector state and $\boldsymbol{f}\colon \mathbb{R}^d \to \mathbb{R}^d$ for the flux function.

Solving equation (1) with high accuracy is a challenging problem since, as is well known, its solutions can and will break down at finite time, even if the initial data are smooth, and develop complex structures (shock wave interactions). In such situation, uniqueness of the (weak) solution is lost, but it can be recovered by completing the system (1) with the entropy inequality

$$\frac{\partial s(\boldsymbol{w})}{\partial t} + \frac{\partial \psi(\boldsymbol{w})}{\partial x} \leq 0, \tag{2}$$

where (s, ψ) stands for a convex entropy–entropy flux pair. This inequality allows to select the physical relevant solution. Moreover, in the regions where the solution is smooth the entropy inequality becomes an equality (the conservation equation). In simple cases, it can be proved that the term missing in (2) to make it an equality is a Dirac mass.

The numerical approximation of (1)–(2) leads to the so-called numerical density of entropy production which is a measure of the amount of violation of the entropy equation (as a measure of the local residual as in [3, 12, 14, 15]). This density provides the information if the mesh should be refined locally (e.g. if the solution develops discontinuities) or coarsened (e.g. if the solution is smooth and well-approximated), see Puppo [18–20] and Golay [9]. As shown in Puppo [20] (see also references therein), even if the shocks are well-captured on the coarse grid using finite volume scheme such indicator is able not only to provide the efficient *a posteriori* error, but also to reproduce the qualitative structure of the solution and to pilot the adaptive scheme. Unfortunately, this does not occur efficiently on contact discontinuities where only numerical diffusion is active. Nonetheless, the work by Guermond *et al.* [10] provides a remarkable and simple way to solve this problem by making the numerical diffusion to be proportional to the numerical density of entropy production (i.e. adding large numerical dissipation in the shock regions and almost no dissipation in the regions where the solution remains smooth).

Explicit adaptive schemes are well known to be time consuming due to the CFL stability condition. The CPU-time increases rapidly as the mesh is refined since the Courant–Friedrich–Lewy (CFL) stability condition imposes an upper bound on $\delta t/h$ where δt is the time step and h the finest mesh size. Nevertheless, the CPU-time can be significantly reduced using the local time stepping algorithm, see e.g. [2, 16, 25]. In recent years, such numerical schemes have been widely developed for equations which arise in many fluid flows: traffic flows, multi-phase flows, multi-layer flows, *etc.* The local time stepping algorithm has been employed successfully for real two and three dimensional steady and unsteady problems, see for instance [13, 16, 17, 20, 24, 26].

The aim of this paper is to construct a 1D multiscale adaptive numerical scheme for hyperbolic conservation laws. The paper is organized as follows. In Section 2, we recall the finite volume approximation, the time integration methods and fix the notation. Thereafter, in Section 3, we explain how to adapt the space grid using the numerical density of entropy production. Combining both the time integration and mesh refinement, we show how to implement the multiscale adaptive numerical scheme. The general algorithm and two approaches to distribute the flux between two cells of different levels are compared in Section 4. Finally, in Section 5, efficiency of the multiscale adaptive scheme is shown through Sod's shock tube problem, Lax's shock tube problem and the Shu–Osher test case.

2. Numerical approximation for hyperbolic conservation laws

2.1. One-dimensional finite volume formulation

We recall here the well-known construction of the numerical approximation of the following general nonlinear hyperbolic conservation laws:

$$\begin{cases} \dfrac{\partial \boldsymbol{w}}{\partial t} + \dfrac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial x} = 0, & (t,x) \in \mathbb{R}^+ \times \Omega, \\ \boldsymbol{w}(0,x) = \boldsymbol{w}_0(x), & x \in \mathbb{R}, \end{cases} \tag{3}$$

where $\Omega \subset \mathbb{R}$, $\boldsymbol{w} \in \mathbb{R}^d$ denotes the vector state (referred to as the conservative variables, see e.g. equations (13)) and $\boldsymbol{f}$ denotes the flux governing the physical description of the flow.

For the sake of simplicity, we consider here the one-dimensional case. The computational domain is split into control volumes $C_k = (x_{k-1/2}, x_{k+1/2})$ of mesh size h_k with $x_{k\pm1/2} = x_k \pm h_k/2$. The unknowns $\boldsymbol{w}(t,x)$ are approximated by their mean values on the cell C_k at time t,

$$\boldsymbol{w}_k(t) \approx \frac{1}{h_k}\int_{C_k} \boldsymbol{w}(t,x)\,dx.$$

Integrating (3) over each cell and applying the Green formula we obtain

$$h_k \frac{\partial \boldsymbol{w}_k}{\partial t} + \boldsymbol{F}_{k+1/2}(t) - \boldsymbol{F}_{k-1/2}(t) = 0, \tag{4}$$

where $\boldsymbol{F}_{k\pm1/2}$ stands for the numerical flux at the interface $x_{k\pm1/2}$, see for instance [6, 8, 27]. Denote the flux difference between two interfaces of C_k as

$$\delta \boldsymbol{F}_k(t) = \boldsymbol{F}_{k+1/2}(t) - \boldsymbol{F}_{k-1/2}(t),$$

then equation (4) takes the form

$$h_k \frac{\partial \boldsymbol{w}_k(t)}{\partial t} + \delta \boldsymbol{F}_k(t) = 0. \tag{5}$$

For practical purposes, let $\boldsymbol{F} = \boldsymbol{f}(\boldsymbol{R}(0^{\pm}, \boldsymbol{w}_{\mathrm{L}}, \boldsymbol{w}_{\mathrm{R}}))$, where $\boldsymbol{R}(0^{\pm}, \boldsymbol{w}_{\mathrm{L}}, \boldsymbol{w}_{\mathrm{R}})$ is the exact solution of the following Riemann problem associated with the left $\boldsymbol{w}_{\mathrm{L}}$ and the right $\boldsymbol{w}_{\mathrm{R}}$ state:

$$\begin{cases} \dfrac{\partial \boldsymbol{w}}{\partial t} + \dfrac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial x} = 0, \\ \boldsymbol{w}_0(x) = \begin{cases} \boldsymbol{w}_{\mathrm{L}} & \text{if} \quad x < 0, \\ \boldsymbol{w}_{\mathrm{R}} & \text{if} \quad x > 0. \end{cases} \end{cases} \tag{6}$$

Then, the semidiscrete first order numerical scheme reads, see for instance [6, 8, 27],

$$h_k \frac{\partial \boldsymbol{w}_k(t)}{\partial t} + \delta \boldsymbol{F}_k(t) = h_k \frac{\partial \boldsymbol{w}_k(t)}{\partial t} + \boldsymbol{f}\big(\boldsymbol{R}(0^-, \boldsymbol{w}_k(t), \boldsymbol{w}_{k+1}(t))\big) - \boldsymbol{f}\big(\boldsymbol{R}(0^+, \boldsymbol{w}_{k-1}(t), \boldsymbol{w}_k(t))\big).$$

Equation (3) is completed with the entropy inequality (2). As mentioned in the introduction, (2) not only allows to select the physical relevant solution but also can be used to define the relevant mesh refinement criterion (with the help of the so-called *numerical density of entropy production*, see Section 3). To this end, (2) is discretized using the same scheme as applied to equation (3):

$$h_k \frac{\partial s(\boldsymbol{w}_k)}{\partial t} + \delta \psi_k(t) = h_k \frac{\partial s(\boldsymbol{w}_k)}{\partial t} + \psi\big(\boldsymbol{R}(0^-, \boldsymbol{w}_k(t), \boldsymbol{w}_{k+1}(t))\big) - \psi\big(\boldsymbol{R}(0^+, \boldsymbol{w}_{k-1}(t), \boldsymbol{w}_k(t))\big), \tag{7}$$

where $\delta\psi_k(t) = \psi_{k+1/2}(t) - \psi_{k-1/2}(t)$ and $\boldsymbol{R}$ is the solution of the Riemann problem (6).

To achieve the second order approximation in space, the MUSCL reconstruction can be used, which we now recall for the sake of completeness, see for instance [8, 27]. It consists of solving the Riemann problem (6) with the linearly extrapolated initial data

$$\left(\boldsymbol{w}_k + \frac{\mathcal{M}_k}{2},\ \boldsymbol{w}_{k+1} - \frac{\mathcal{M}_{k+1}}{2}\right),$$

where $\mathcal{M}_k$ is the approximation to the slope obtained with the MinMod limiter, i.e.,

$$\mathcal{M}_k = \begin{cases} m \min\left(\Delta \boldsymbol{w}_{k-1/2}, \Delta \boldsymbol{w}_k, \Delta \boldsymbol{w}_{k+1/2}\right) & \text{if} \quad m = \operatorname{sgn} \Delta \boldsymbol{w}_{k-1/2} = \operatorname{sgn} \Delta \boldsymbol{w}_k = \operatorname{sgn} \Delta \boldsymbol{w}_{k+1/2}, \\ 0 & \text{otherwise}, \end{cases}$$

$$\Delta \boldsymbol{w}_{k-1/2} = \boldsymbol{w}_k - \boldsymbol{w}_{k-1}, \qquad \Delta \boldsymbol{w}_{k+1/2} = \boldsymbol{w}_{k+1} - \boldsymbol{w}_k, \qquad \Delta \boldsymbol{w}_k = \frac{\Delta \boldsymbol{w}_{k-1/2} + \Delta \boldsymbol{w}_{k+1/2}}{2}.$$

Thus, we define the numerical flux as

$$\boldsymbol{F}_{k+1/2} = \boldsymbol{f}\left(\boldsymbol{R}\left(0^-,\ \boldsymbol{w}_k + \frac{\mathcal{M}_k}{2},\ \boldsymbol{w}_{k+1} - \frac{\mathcal{M}_{k+1}}{2}\right)\right), \qquad \boldsymbol{F}_{k-1/2} = \boldsymbol{f}\left(\boldsymbol{R}\left(0^+,\ \boldsymbol{w}_{k-1} + \frac{\mathcal{M}_{k-1}}{2},\ \boldsymbol{w}_k - \frac{\mathcal{M}_k}{2}\right)\right).$$

2.2. Time integration

Now we focus on the numerical time integration of equations (5) and (7). First recall the well-known time integration methods of Runge–Kutta and Adams–Bashforth and fix the notation.

2.2.1. Runge–Kutta schemes

Integrating equations (5) and (7) during the time step $]t_n, t_{n+1}[$ of length δt_n and evaluating the numerical fluxes at time t_n, we obtain the well-known Euler scheme,

$$\boldsymbol{w}_k(t_{n+1}) = \boldsymbol{w}_k(t_n) - \frac{\delta t_n}{h_k}\,\delta \boldsymbol{F}_k(t_n), \qquad s(\boldsymbol{w}_k(t_{n+1})) = s(\boldsymbol{w}_k(t_n)) - \frac{\delta t_n}{h_k}\,\delta\psi_k(t_n). \tag{8}$$

In order to increase the accuracy, apply the second order method as follows:

$$\boldsymbol{w}_k(t_{n+1}) = \boldsymbol{w}_k(t_n) - \frac{\delta t_n}{h_k}\,\delta \boldsymbol{F}_k(t_{n+1/2}), \qquad \boldsymbol{w}_k(t_{n+1/2}) = \boldsymbol{w}_k(t_n) - \frac{\delta t_n}{2h_k}\,\delta \boldsymbol{F}_k(t_n),$$

with the time discretization of the entropy the same as the time discretization of conservative variables. Now, as in [20], we can define the *numerical density of entropy production* S_k^n as

$$S_k^n = \frac{s(\boldsymbol{w}_k(t_{n+1})) - s(\boldsymbol{w}_k(t_n))}{\delta t_n} + \frac{\delta\psi_k(t_n)}{h_k} \tag{9}$$

for the first order scheme, and

$$S_k^n = \frac{s(\boldsymbol{w}_k(t_{n+1})) - s(\boldsymbol{w}_k(t_n))}{\delta t_n} + \frac{\delta\psi_k(t_{n+1/2})}{h_k}$$

for the second order scheme. The value

$$\mathcal{P} = \sum_{n,k} S_k^n\,\delta t_n\,h_k \tag{10}$$

call the *total numerical entropy production*.

2.2.2. Adams–Bashforth schemes

In order to avoid intermediate computation and to spare computing time, one can use a *multitime step method* such as the Adams–Bashforth method. The Adams–Bashforth method of order m consists of replacing the numerical flux of equation (5) with a polynomial interpolation of the same order [11], i.e.

$$\boldsymbol{F}_{k+1/2}(t) \approx \sum_{j=0}^{m-1} L_j(t)\,\boldsymbol{F}_{k+1/2}(t_{n-j}),$$

where $L_j(t)$ denotes the coefficients of the Lagrange polynomial. Let us note that this explicit approximation is built with the fluxes previously computed and stored. If the polynomial coefficients are integrated (e.g. by the numerical Gauss integration),

$$b_j(\delta t_n) = \frac{1}{\delta t_n}\int_{t_n}^{t_{n+1}} L_j(t)\,dt,$$

then the integration of equations (5) and (7) leads to

$$\boldsymbol{w}_k(t_{n+1}) = \boldsymbol{w}_k(t_n) - \sum_{j=0}^{m-1}\frac{\delta t_n}{h_k}\,b_j(\delta t_n)\,\delta \boldsymbol{F}_k(t_{n-j}).$$

For example, the second order Adams–Bashforth method reads

$$\boldsymbol{w}_k(t_{n+1}) = \boldsymbol{w}_k(t_n) - \frac{\delta t_n}{h_k}\,\delta \boldsymbol{F}_k(t_n) - \frac{\delta t_n^2}{2\delta t_{n-1}\,h_k}\big(\delta \boldsymbol{F}_k(t_n) - \delta \boldsymbol{F}_k(t_{n-1})\big).$$

The first order method is the same as (8). The Adams–Bashforth methods of the second and third order are stable under the CFL condition with the bound equal to 1 [1]. For the stability purpose, we limit our applications to the second order scheme.

Perform the discretization as in (7) and define the numerical density of the entropy production S_k^n by (9) for the first order scheme and as

$$S_k^n = \frac{s(\boldsymbol{w}_k(t_{n+1})) - s(\boldsymbol{w}_k(t_n))}{\delta t_n} + \frac{\delta \psi_k(t_n)}{h_k} + \frac{\delta t_n}{2\delta t_{n-1} h_k}\big(\delta \psi_k(t_n) - \delta \psi_k(t_{n-1})\big)$$

for the second order scheme. Finally, the total numerical entropy production define as in (10).

3. Mesh refinement

In order to compute the solution of a hyperbolic system with the prescribed accuracy, one can apply a strategy to adapt the mesh automatically following the local error indicator.

3.1. Mesh refinement criterion

Many works studying the *a posteriori* error estimate base on mathematical arguments [28]. But paradoxically, to our knowledge, very few works use a physical criterion [5] — the numerical density of entropy production. Recently, Puppo [18, 19] has proved that the numerical density of entropy production can be used as a discontinuity indicator (whenever solution develops shocks or contact discontinuities) and a local error indicator (whenever the solution remains smooth). As the entropy inequality is mathematically and physically related to the system (3), the numerical density of entropy production can serve (and actually does, as we will see in Section 5) for an accurate and useful mesh refinement criterion. Let us show how to use it in the automatic mesh refinement framework.

3.2. Mesh refinement process

We present here the local mesh refinement procedure driven by the numerical density of entropy production. In order to reduce the time necessary to manage the refinement, we use the so-called "macro cells" which could be refined by generating hierarchical grids. Each cell can be split into two sub-cells. We thus produce a dyadic cells graph, whose numbering (in basis 2) allows a quick computing scan to determine the adjacent cells. We make use of the following notation: let k_b be the index which makes reference to the macro cell numbered k and b be a binary number which contains the hierarchical information of a sub-cell. In particular, the level of a sub-cell C_{k_b} is defined as length$(b) - 1$. For instance, the macro cell C_{k_0} of level 0 will be split into two sub-cells $C_{k_{00}}$ and $C_{k_{01}}$ of level 1. A mesh refinement example is proposed in Figure 1.

Define the mesh refinement procedure as follows. First fix a mesh refinement parameter $\overline{S}$. For instance, it can be the mean value over the domain Ω at time t_n,

$$\overline{S} = \frac{1}{|\Omega|}\sum_{k_b} S_{k_b}^n. \tag{11}$$

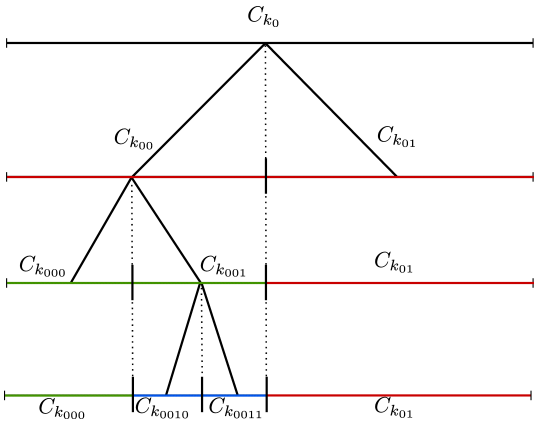


Figure 1. Example of the hierarchical dyadic tree.

Next define two coefficients $0 \leqslant \alpha_{\min} \leqslant \alpha_{\max} \leqslant 1$, which determine the ratio of numerical production of entropy leading to the mesh refinement or mesh coarsening: for each cell C_{k_b},

- if $S^n_{k_b} > \overline{S}\alpha_{\max}$, the mesh is refined and split into two sub-cells $C_{k_{b0}}$ and $C_{k_{b1}}$,
- if $S^n_{k_{b0}} < \overline{S}\alpha_{\min}$ and $S^n_{k_{b1}} < \overline{S}\alpha_{\min}$, the mesh is coarsened into the cell C_{k_b}.

For the first order scheme, if the cell C_{k_b} is split into two sub-cells $C_{k_{b0}}$ and $C_{k_{b1}}$, the averaged values $w^n_{k_b}$ are projected on each sub-cell:

$$\boldsymbol{w}^n_{k_{b0}} = \boldsymbol{w}^n_{k_{b1}} = \boldsymbol{w}^n_{k_b}.$$

In the case of the second order method, as the gradient is available, let

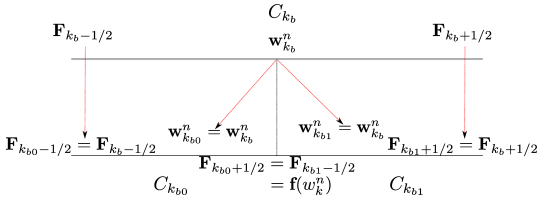
$$\boldsymbol{w}^n_{k_{b0}} = \boldsymbol{w}^n_{k_b} - \frac{h_{k_b}}{4}\frac{\partial \boldsymbol{w}^n_{k_b}}{\partial x}, \qquad \boldsymbol{w}^n_{k_{b1}} = \boldsymbol{w}^n_{k_b} + \frac{h_{k_b}}{4}\frac{\partial \boldsymbol{w}^n_{k_b}}{\partial x}.$$

Define the numerical fluxes between the two sub-cells (see Figure 2(a)) as

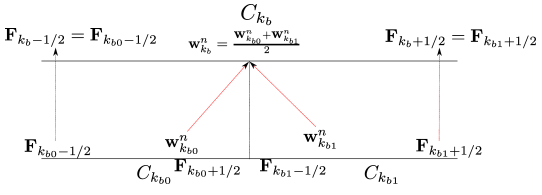
$$\boldsymbol{F}^n_{k_{b0}-1/2} = \boldsymbol{F}^n_{k_b-1/2}, \qquad \boldsymbol{F}^n_{k_{b0}+1/2} = \boldsymbol{F}^n_{k_{b1}-1/2} = \boldsymbol{f}(\boldsymbol{w}^n_{k_b}), \qquad \boldsymbol{F}^n_{k_{b1}+1/2} = \boldsymbol{F}^n_{k_b+1/2}.$$

On the other hand, if two sub-cells $C_{k_{b0}}$ and $C_{k_{b1}}$ are coarsened, initialize the new cell C_{k_b} with, see Figure 2(b),

$$\boldsymbol{w}^n_{k_b} = \frac{\boldsymbol{w}^n_{k_{b0}} + \boldsymbol{w}^n_{k_{b1}}}{2}, \qquad \boldsymbol{F}^n_{k_b-1/2} = \boldsymbol{F}^n_{k_{b0}-1/2}, \qquad \boldsymbol{F}^n_{k_b+1/2} = \boldsymbol{F}^n_{k_{b1}+1/2}.$$



(a) Mesh refinement.



(b) Mesh coarsening.

Figure 2. Mesh refinement and mesh coarsening: construction of the fluxes.

Remark 3.1.
It is hard to define an optimal mesh refinement criterion since the smaller the threshold the more the mesh is unnecessarily refined, and the larger threshold the less the mesh is refined. Therefore, a good compromise is to use the average value of the mesh refinement criterion (11). The other threshold parameters $\alpha_{\min}$ and $\alpha_{\max}$ allow to set the percentage of the mesh refinement and mesh coarsening with respect to the quantity $\overline{S}$. It is not surprising that these settings will deteriorate or improve the accuracy of the numerical solution. For instance, the smaller $\alpha_{\min}$ and $\alpha_{\max}$ the more accurate are the results to the expense of the computational time.

4. The local time stepping method

Unlike the Runge–Kutta method, the Adams–Bashforth time integration can be easily implemented in the *local time stepping* framework in order to reduce the computational cost. We recall the main approach to update averaged quantities and provide the general algorithm.

4.1. The local time stepping algorithm

In this subsection, we combine the local time stepping algorithm, see for instance [2], with the mesh refinement process presented above. For the sake of clarity, we do not use the binary subscript notation introduced in the previous section.

Let $h_{\min}$ be the minimum diameter of the mesh and $h_{\max}$ the maximum diameter of the mesh. All cells of the mesh are sorted in groups, called "levels" with respect to their diameter. The level of refinement L_k of the k^{th} cell C_k is defined as

$$2^{N-L_k} h_{\min} \leqslant h_k < 2^{N+1-L_k} h_{\min},$$

where N stands for the maximum level

$$N = \log_2 \frac{h_{\max}}{h_{\min}} + 1.$$

The coarsest cells are therefore of level 1, while the finest cells are of level N.

Remark 4.1.
For any mesh, the previous inequality holds and in the particular case of the dyadic mesh we have

$$h_k = 2^{N-L_k} h_{\min}.$$

Let δt_n be the minimum time step at time t_n according to the CFL condition associated with the smallest cell. The macro time step Δt_n is then defined as

$$\Delta t_n = 2^{N-1} \delta t_n.$$

We define the level of interface of two adjacent cells $L_{k+1/2}$ as

$$L_{k+1/2} = \max(L_k, L_{k+1}).$$

Assuming that the maximum level of refinement is N at the current time t_n, following [2], the local time stepping algorithm reads

foreach $i \in \{1, 2^{N-1}\}$ **do**
  Let j be the biggest integer such that 2^j divides i
  foreach interface $x_{k+1/2}$ such that $L_{k+1/2} \geqslant N - j$ **do**
    1. compute the integral of $\boldsymbol{F}_{k+1/2}(t)$ on the time interval $2^{N-L_{k+1/2}}\delta t_n$,
    2. distribute $\boldsymbol{F}_{k+1/2}(t_n)$ to the two adjacent cells,
    3. update only the cells of level greater than $N - j$.
  end
end

Remark 4.2.
Note that according to this algorithm, the flow is computed more times in small cells than in the bigger ones.

Remark 4.3.
The minimum time step is defined by the CFL condition associated with the smallest cell at time t_n, i.e.,

$$\delta t_n < \frac{h_{\min}}{\max|v|},$$

where v is the eigenvalue of the convection matrix $D_{\boldsymbol{w}}\boldsymbol{f}(\boldsymbol{w})$.

Remark 4.4.
This scheme uses the Osher and Sanders [17] projection. This point is developed in subsection 4.2.

Let us illustrate for the global time advancement the local time stepping algorithm for the first order scheme (the principle for the second order is essentially the same since we have to use the value of the flux stored in the two previous local time steps). We consider the mesh composed of four cells of levels 1, 2 and 3 respectively, i.e., here $N = 3$, as displayed in Figure 3. Let us denote by $\boldsymbol{w}_m^n$ the state vector at time t_n. At the first stage, $i = 1$, see Figure 3(a), only the small cells are advanced with the step δt^n, i.e. $\boldsymbol{w}_3^{n1} = \boldsymbol{w}_3(t_n + \delta t^n)$ and $\boldsymbol{w}_4^{n1} = \boldsymbol{w}_4(t_n + \delta t^n)$ are computed. While at the second stage, $i = 2$, see Figure 3(b), the the small cells of level L_N and the cells immediately larger at level L_{N-1} are involved. The cells L_N travel with the time step δt_n, while the cells L_{N-1} travel with the time step $2\delta t_n$, *etc.*

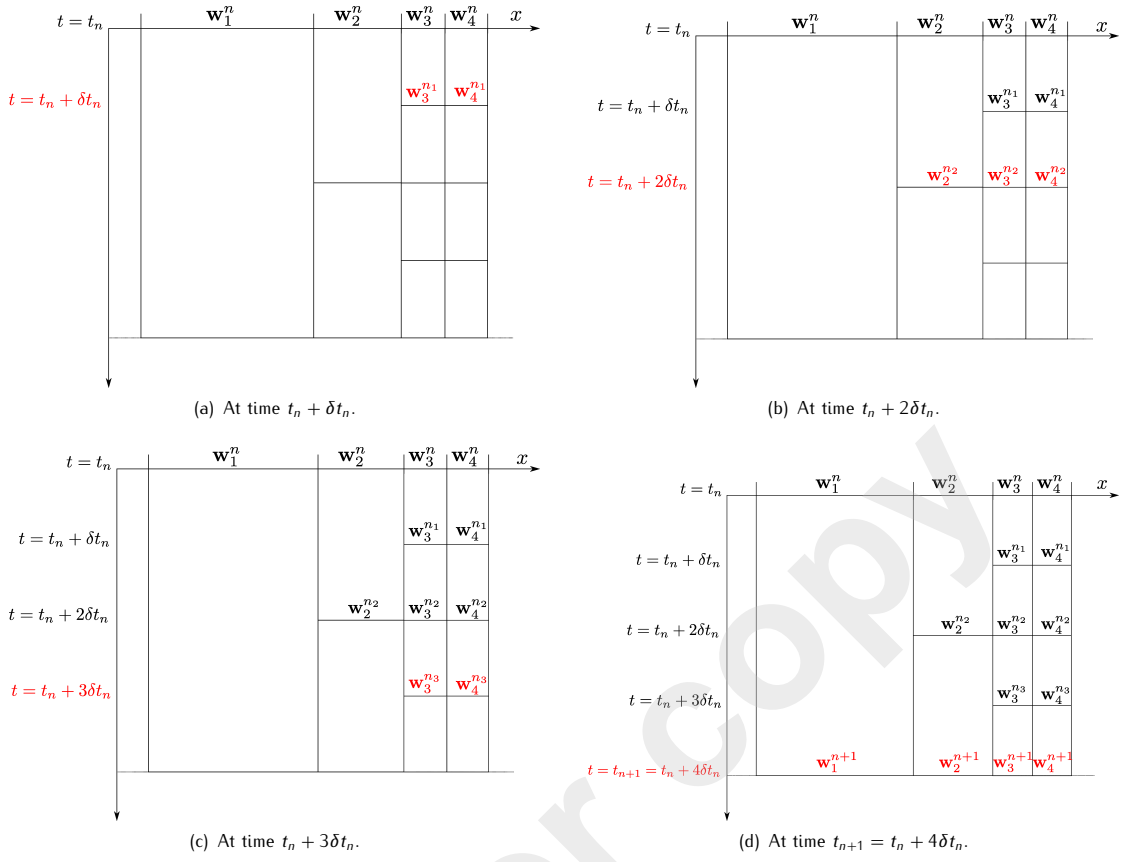
4.2. Computation of the flux

In this subsection we recall the procedure introduced by Osher and Sanders [17], and Tang and Warnecke [26] for computation of the flux at the cell interface of two different levels. For the sake of simplicity, we assume here only two time increments which means that we consider cells of only two different kinds, say $h(x) = h_{k_0} = h_0$ for $x < x_{k_0+1/2}$ and $h(x) = h_{k_0+1} = h_0/2$ for $x > x_{k_0+1/2}$. Thus, for every $k \leqslant k_0$, all cells C_k are of level L_{k_0} and for every $k > k_0$, all cells C_k are of level $L_{k_0} + 1$ (as displayed in Figure 4). Let $\Delta t_n = t_{n+1} - t_n = 2\delta t_n$ be the macro time step. To avoid boundary terms, we consider the example with the initial condition with compact support.

4.2.1. A simple projection method

The Osher and Sanders scheme is defined as follows, see Figure 4(b):

$$\boldsymbol{w}_k^{n+1/2} = \begin{cases} \boldsymbol{w}_k^n & \text{if} \quad k \leqslant k_0, \\ \boldsymbol{w}_k^n - \dfrac{\Delta t_n}{2h_k}\,\delta \boldsymbol{F}_k^n & \text{if} \quad k > k_0, \end{cases} \qquad \text{for all} \quad k \in \mathbb{Z},$$



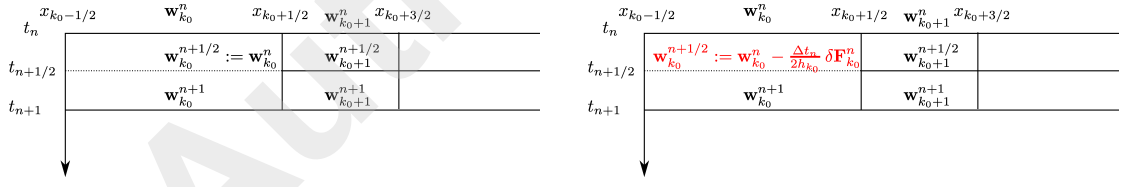
(a) At time $t_n + \delta t_n$.

(b) At time $t_n + 2\delta t_n$.

(c) At time $t_n + 3\delta t_n$.

(d) At time $t_{n+1} = t_n + 4\delta t_n$.

Figure 3. Illustration of the global time advancement for the first order scheme.



(a) The Osher and Sanders projection.

(b) The Tang and Warnecke projection.

Figure 4. Projection methods.

and

$$\boldsymbol{w}_k^{n+1} = \boldsymbol{w}_k^n - \frac{\Delta t_n}{2h_k}\left(\delta \boldsymbol{F}_k^n + \delta \boldsymbol{F}_k^{n+1/2}\right), \qquad k \in \mathbb{Z}. \tag{12}$$

It was pointed by Tang and Warnecke [26] that the present scheme loses the conservative property and is locally inconsistent in the sense of the truncation error. Indeed, at the intermediate time step, one has

$$\sum_{k\in\mathbb{Z}} \boldsymbol{w}_k^{n+1/2} = \sum_{k\in\mathbb{Z}} \boldsymbol{w}_k^n + \frac{\Delta t_n}{2h_{k_0+1}} \boldsymbol{F}_{k_0+1/2}^n(\boldsymbol{w}_{k_0}^n, \boldsymbol{w}_{k_0+1}^n)$$

which shows that this scheme is locally nonconservative. At the next intermediate step, conservativity is preserved, i.e.,

$$\sum_{k\in\mathbb{Z}} \boldsymbol{w}_k^{n+1} = \sum_{k\in\mathbb{Z}} \boldsymbol{w}_k^{n+1/2}$$

but the global conservativity is lost. The fact that the scheme is not consistent in the sense of truncation error can be observed in the simplest case of the one-dimensional transport equation, as done in [26],

$$\partial_t \boldsymbol{w} + c\partial_x \boldsymbol{w} = 0$$

with a positive constant c. In this case, using the Godunov solver, the scheme (12) writes for $k = k_0$ as

$$\frac{\boldsymbol{w}_{k_0+1}^{n+1} - \boldsymbol{w}_{k_0+1}^{n}}{\Delta t_n} - \frac{c}{2h_{k_0}}\left(\boldsymbol{w}_{k_0+1}^{n} - \boldsymbol{w}_{k_0}^{n} + \left(\boldsymbol{w}_{k_0+1}^{n+1/2} - \boldsymbol{w}_{k_0}^{n}\right)\right) = 0.$$

Assuming $\boldsymbol{w}$ smooth enough, writing the Taylor expansion of $\boldsymbol{w}$ at the point x_{k_0+1} at times t_{n+1} and $t_{n+1/2}$, we deduce the following equation:

$$\partial_t \boldsymbol{w} + c\partial_x \boldsymbol{w} = -\frac{c}{4}\frac{\Delta t}{h}\,\partial_t \boldsymbol{w} + \mathcal{O}(h, \Delta t).$$

Thus, if $\Delta t/h$ converges to a constant $d > 0$ then the previous equation does not converge to the initial transport equation.

4.2.2. A natural projection method

Contrary to the Osher and Sanders [17] approach, Tang and Warnecke [26] propose the following scheme:

$$\boldsymbol{w}_k^{n+1/2} = \boldsymbol{w}_k^{n} - \frac{\Delta t_n}{2h_k}\,\delta \boldsymbol{F}_k^{n}, \qquad \boldsymbol{w}_k^{n+1} = \boldsymbol{w}_k^{n+1/2} - \frac{\Delta t_n}{2h_k}\begin{cases}\delta \boldsymbol{F}_k^{n} & \text{if} \quad k \leqslant k_0,\\ \delta \boldsymbol{F}_k^{n+1/2} & \text{if} \quad k > k_0,\end{cases} \qquad k \in \mathbb{Z}.$$

One can easily check that the global conservativity is also lost and

$$\sum_{k\in\mathbb{Z}} \boldsymbol{w}_k^{n+1} = \sum_{k\in\mathbb{Z}} \boldsymbol{w}_k^{n} + \frac{\Delta t_n}{2h_{k_0+1}}\,\boldsymbol{F}_{k_0+1/2}^{n+1/2}(\boldsymbol{w}_{k_0}^{n}, \boldsymbol{w}_{k_0+1}^{n}).$$

Although the scheme is not conservative, they claim that it is consistent in the sense of the truncation error.

Even if the Osher and Sanders projection leads to nonconservative and inconsistent scheme at the space-time grid between two levels of refinement, it has the advantage to be less time consuming than the second one. Let us also note that the apparent loss of consistency on the local truncation error seems not to affect the actual error of the scheme (as we will see in subsection 5.1.1, see also [4, 22] for interesting issues). On the contrary, Puppo and Semplice [20] do not lose consistency and conservativity because their procedure does not produce the mass and the fluxes are well computed at the cell interface between the two levels of refinement. Nevertheless, their approach requires more evaluation of the flux than the Osher and Sanders projection since they use updated neighbourhood state (non-projected as done in the Osher and Sanders approach) .

5. Numerical experiments

We now present some results using the *adaptive multiscale scheme* constructed in Sections 3 and 4. The discussion is limited to studying the robustness, accuracy and gain of the CPU-time of several schemes. To this end, we compare

results to those from the exact solution or a reference solution. These include Sod's shock tube problem, Lax's shock tube problem, and the Shu–Osher test case.

For all test cases, numerical solutions are computed using the following one-dimensional gas dynamics equations for ideal gas:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0, \qquad \frac{\partial \rho u}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} = 0, \qquad \frac{\partial \rho E}{\partial t} + \frac{\partial (\rho E + p) u}{\partial x} = 0, \qquad p = (\gamma - 1)\rho\varepsilon, \tag{13}$$

where ρ, u, p, γ, E are the density, velocity, pressure, ratio of the specific heats (set to 1.4) and the total energy $E = \varepsilon + u^2/2$ (where ε is the internal specific energy), respectively. Using conservative variables $\boldsymbol{w} = (\rho, \rho u, \rho E)^T$, we classically define the entropy by

$$s(\boldsymbol{w}) = -\rho \ln \frac{p}{\rho^\gamma}$$

and the entropy flux by $\psi(\boldsymbol{w}) = u\, s(\boldsymbol{w})$ for equations (13).

Notation and settings

- In what follows, we perform several numerical tests using the first and second order schemes. We will refer as AB1 to the first order scheme, AB2 to the second order Adams–Basforth scheme, and RK2 to the second order Runge–Kutta scheme. AB2 and RK2 use the MUSCL reconstruction. Moreover, all computations are made with a dynamic grid, except if the acronym ends with the capital letter "U" which refers to the fixed uniform grid. All computations are made with the uniform time step, except if the acronym ends with the capital letter "M" which refers to the local time stepping algorithm.
- We will also compare the adaptive numerical solution with the one computed on the fixed uniform grid. To have a coherent support of comparison, the solution on the fixed grid will be computed with $N_{L_{\max}}$ cells. $N_{L_{\max}}$ stands for the average number of cells used during the simulation of the adaptive scheme with the maximum level $L_{\max}$.
- All presented results display the density which is positive. Thus, in all figures the numerical density of entropy production is plotted with the reversed sign to overlap with the density.
- For all numerical tests, the initial mesh is adapted to the initial data during a few iterations (except for the numerical solution computed on the fixed uniform grid) and we have used the following threshold parameters: mesh refinement parameter $\alpha_{\max} = 0.01$, mesh coarsening parameter $\alpha_{\min} = 0.001$, and mesh refinement parameter

$$\overline{S} = \frac{1}{|\Omega|} \sum_{k_b} S^n_{k_b}.$$

- ρ_{ex} will stand for the exact solution of Sod's or Lax's problems.

5.1. Sod's shock tube problem

We first consider the classical Sod's shock tube problem [23] which models the one-dimensional flow resulting from the rupture of a membrane separating air with different density and pressure inside a shock tube. The advantage of this test case is that the exact solution exists which enables a detailed comparison between the approximate results and the exact solution.

The test consists of the one-dimensional Riemann problem with the following Riemann data:

$$x \in [-1, 1], \qquad (\rho, u, p)(0, x) = \begin{cases} (1, 0, 1), & x \leqslant 0, \\ (0.125, 0, 0.1), & x > 0. \end{cases} \tag{14}$$

It consists of two nonlinear waves (one the right shock and one the left rarefaction) connected through a contact discontinuity. This basic test provides information on how well a scheme captures and resolves shocks and contact discontinuities and how well the correct density of the rarefaction wave is reproduced. It is used as the first test.

The computational domain here is $[-1, 1]$ with prescribed free boundary conditions. For each numerical computation, the following parameters have been used:

CFL:	0.25,
Simulation time (s):	0.4,
Initial number of cells:	200,
Maximum level of mesh refinement:	$L_{\max}$.

5.1.1. The Osher–Sanders and the Tang–Warnecke projections

In subsection 4.2, we have recalled the computation of the numerical flux between two levels of refinement. We compare here the Tang–Warnecke (TW) [26] and the Osher–Sanders (OS) [17] projections. In particular, we numerically show that even if the TW projection leads to a consistent scheme, for $L_{\max}$ large enough (here 6), numerical oscillations are still observed, see Figure 5(b) and 5(c). The same behavior is obtained for the OS projection as displayed in Figure 5(a)–5(c). We can reduce these effects using the smoothing grid technique. It means that we prevent two adjacent cells from having a level difference greater than two. This method minimizes the loss of consistency and conservativity error. We have displayed on Figure 5 the density of the solution at time $t = 0.4$ using the first order scheme AB1M (using AB2M leads to the same conclusion). To compare these methods, we have also computed the CPU-time. Without the smoothing effect we have

$$\text{CPU-time for TW projection} = 330.27\,\text{s}, \qquad \text{CPU-time for OS projection} = 85.02\,\text{s},$$

and with the smoothing effect we have

$$\text{CPU-time for TW projection} = 288.21\,\text{s}, \qquad \text{CPU-time for OS projection} = 80.04\,\text{s}.$$

As displayed in Figure 5 and the CPU performance above, even if the results using the TW projection are more accurate than the OS ones, the TW projection is more time consuming because it is necessary to update the fluxes on large cells at small intermediate times if they border a smaller cell. As a consequence, we will use in what follows the OS projection. Let us also note that such a problem appears when $L_{\max}$ is large. In order to avoid such oscillations, we will limit $L_{\max}$ to 5 which is large enough for practical applications.

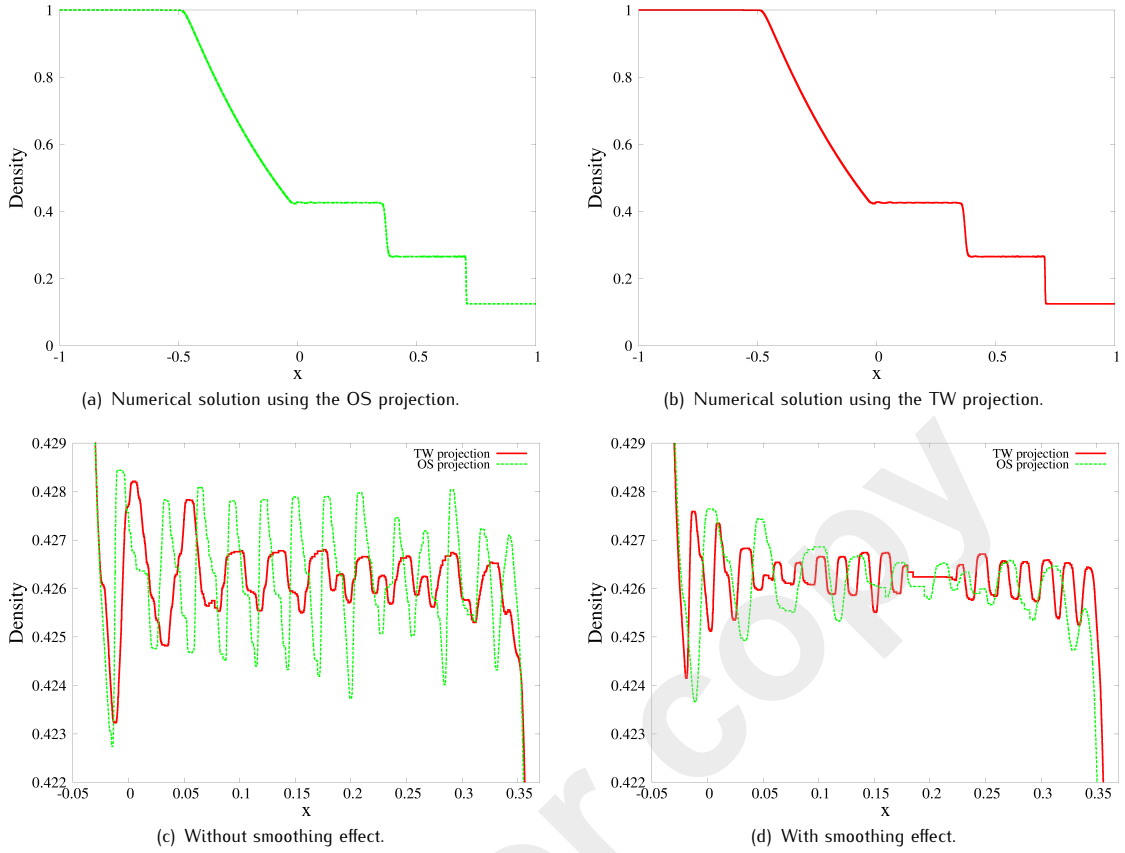
5.1.2. Numerical convergence

We study the numerical convergence of the AB1, AB1M, AB2, AB2M and RK2 schemes. We compare the CPU-time and the accuracy of the numerical solutions for the level $1 \leqslant L_{\max} \leqslant 5$ (i.e. on the fixed and dynamic grid). The analytical solution of the Riemann problem (13) with the Riemann data (14) is computed using 5 000 uniform cells.

The numerical density of entropy production as a discontinuity and error indicator

As emphasized by Puppo [18, 19], the numerical density of entropy production can be used as a discontinuity and a local error indicator (everywhere where the solution remains smooth). Thus, the mesh refinement parameter (11) combined with the numerical density of entropy production, see also Remark 3.1, can provide a useful tool to refine/adapt the mesh around discontinuities and sharply varying regions. In Figures 6–7, we plot the density ρ for levels $L_{\max} = 2$ and 5, which we compare with the numerical solution computed on the fixed grid with $N_{L_{\max}}$ cells which are respectively 254 and 681.

As we observe in Figures 6–7, the numerical density of entropy production for levels $L_{\max} = 2$ and 5 captures the contact discontinuity, see Figures 6(d) and 7(d), and the shock discontinuity, see Figures 6(e) and 7(e). Moreover, the numerical



(a) Numerical solution using the OS projection.

(b) Numerical solution using the TW projection.

(c) Without smoothing effect.

(d) With smoothing effect.

Figure 5. Sod's shock tube problem: numerical oscillations with/without smoothing effect at time $t = 0.4$ with $L_{\max} = 6$. The Tang–Warnecke and the Osher–Sanders projections are used.

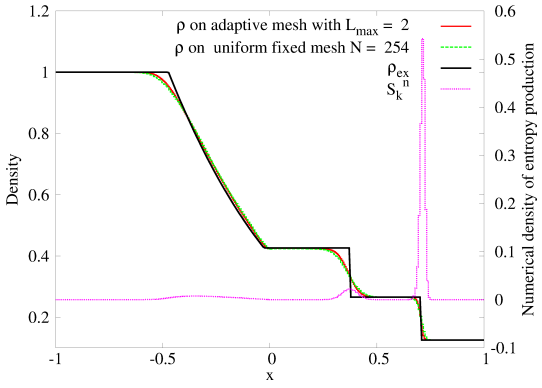
density of entropy production reproduces the behavior of the error made on ρ, as displayed in Figures 6(b) and 7(b), and the mesh is automatically refined in those areas. Far from such regions, the mesh is coarsened since the exact solution and the adaptive one coincide with each other (the precise order and error will be provided in the sequel). The accuracy is, of course, improved when we increase the value of $L_{\max}$ as we can compare in Figures 6 and 7.

The total production of entropy $\mathcal{P}$ (10) corresponds to the discrete $l_t^1 l_x^1$-norm of the numerical density of entropy production. Through several numerical tests, we observe that $\mathcal{P}$ decreases as we refine. For instance, for $L_{\max} = 2$ one has $\mathcal{P} = 0.0098$, while for $L_{\max} = 5$ one has $\mathcal{P} = 0.0073$.

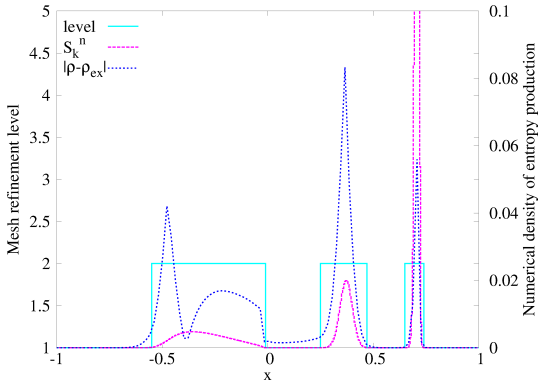
Numerical order

We now perform several numerical tests to compute the rate of convergence of the first and second order adaptive schemes. We use the discrete $l_t^1 l_x^1$-norm of the error on the density, see Figure 8, and the discrete l_x^1-norm of the error on the density at time $t = 0.4$. As the number of cells is not fixed, we consider the average number of cells $N_{L_{\max}}$ for levels $L_{\max} = 1, 2, 3, 4, 5$ used as abscissa in Figure 8(a)–8(d).

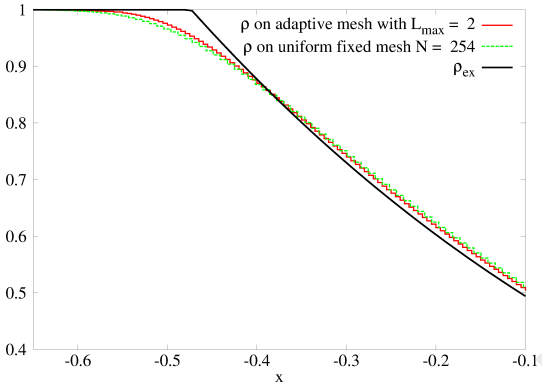
As expected, see for instance [7, 20], the convergence of second order methods is of the first order on the fixed grid due to the presence of discontinuities in the solution, as displayed in Figure 8(b)–8(d). Nevertheless, the adaptive scheme improves the rate of convergence for the first as well as for the second order scheme, see Figure 8(a)–8(d). Moreover, using the local time stepping or the uniform time step, the numerical orders are similar, see Figure 8(a)–8(d), and Table 1.



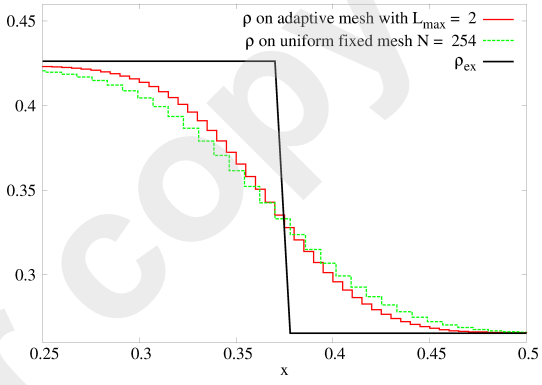
(a) Density and numerical density of entropy production.



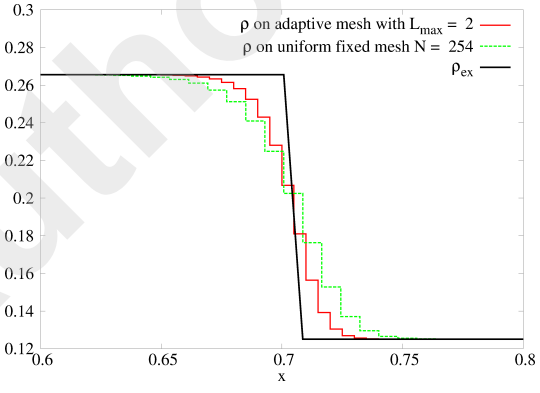
(b) Mesh refinement level, numerical density of entropy production and local error.



(c) Zoom on the rarefaction.



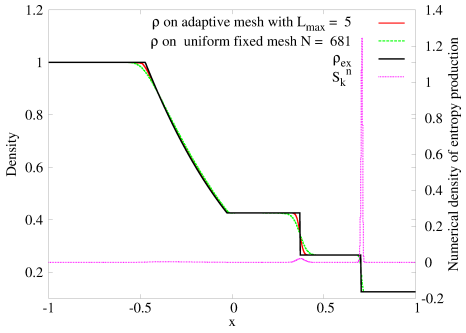
(d) Zoom on the contact discontinuity.



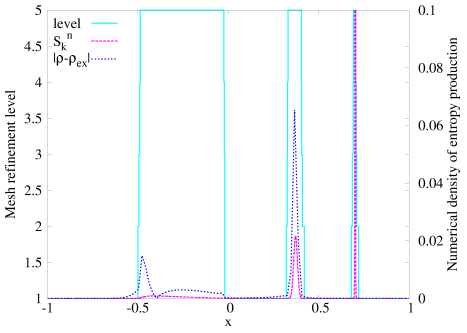
(e) Zoom on the shock.

Figure 6. Sod's shock tube problem: solution at time $t = 0.4$ using the AB1M scheme on the dynamic grid with $L_{max} = 2$ and the AB1 scheme on the fixed uniform grid of 254 cells.

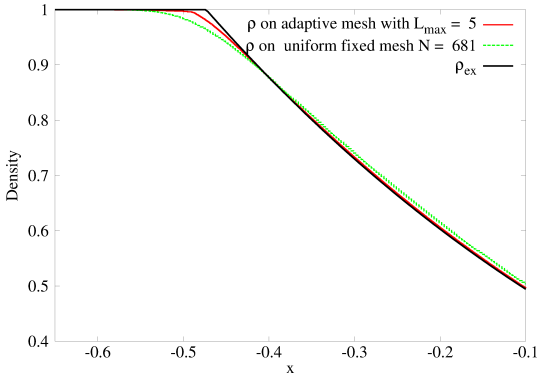
The rate of convergence is considerably increased by the adaptive scheme and, in our experiences, it can be improved by changing the threshold parameters α_{min} and α_{max} at the expense of the CPU-time. With the present threshold parameters, the order of the AB1M scheme is approximately 2.



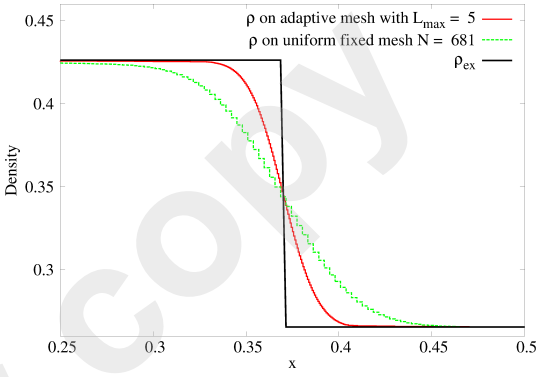
(a) Density and numerical density of entropy production.



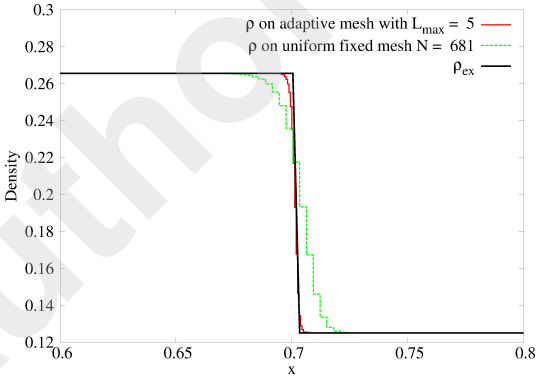
(b) Mesh refinement level, numerical density of entropy production and local error.



(c) Zoom on the rarefaction.

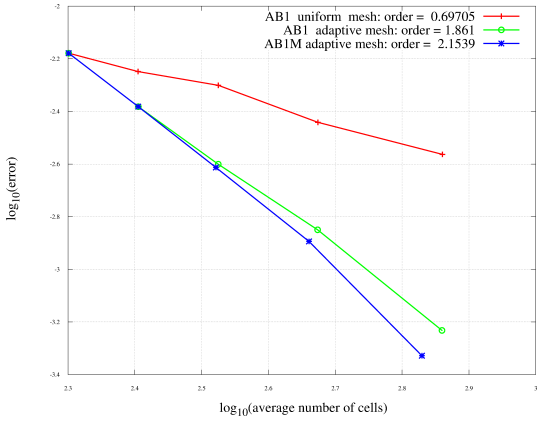


(d) Zoom on the contact discontinuity.

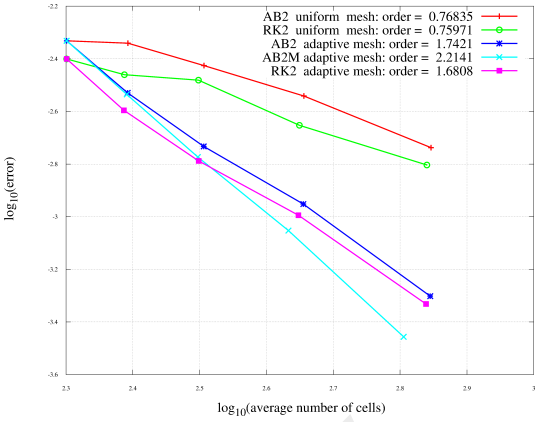


(e) Zoom on the shock.

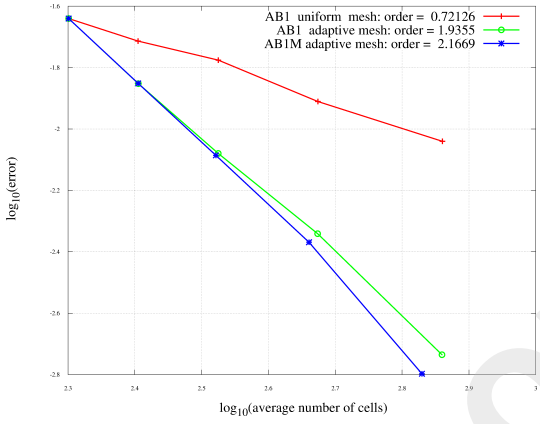
Figure 7. Sod's shock tube problem: solution at time $t = 0.4$ using the AB1M scheme on the dynamic grid with $L_{\max} = 5$ and the AB1 scheme on the fixed uniform grid of 681 cells.



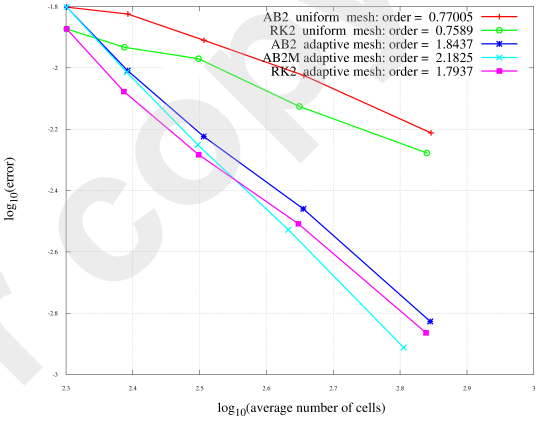
(a) $\|\rho_{ex}-\rho\|_{l_t^1 l_x^1}$ with respect to the averaged number of cells for the schemes of the first order.



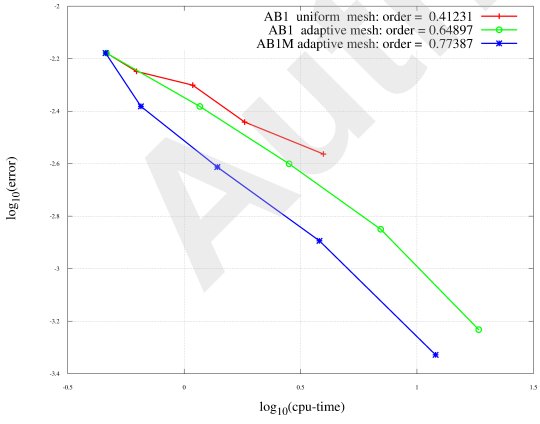
(b) $\|\rho_{ex}-\rho\|_{l_t^1 l_x^1}$ with respect to the averaged number of cells for the schemes of the second order.



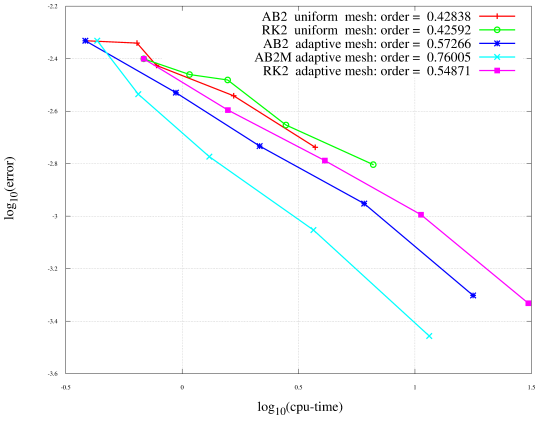
(c) $\|\rho_{ex}-\rho\|_{l_x^1}$ with respect to the averaged number of cells for the schemes of the first order at time $t = 0.4$.



(d) $\|\rho_{ex}-\rho\|_{l_x^1}$ with respect to the averaged number of cells for the schemes of the second order at time $t = 0.4$.



(e) CPU-time-$\|\rho_{ex}-\rho\|_{l_t^1 l_x^1}$ with respect to the CPU-time for the schemes of the first order.



(f) CPU-time-$\|\rho_{ex}-\rho\|_{l_t^1 l_x^1}$ with respect to the CPU-time for the schemes of the second order.

Figure 8. Sod's shock tube problem: numerical order.

Accuracy and CPU-time

Until now, the relevance of the adaptive scheme has been obtained relatively to the number of cells. We now focus on the efficiency of the adaptive scheme with respect to the CPU-time[1]. For the previous numerical tests, we plot the error on the density with respect to the CPU-time, see Figure 8(e)–8(f). As expected, the use of local time stepping algorithm reduces considerably the CPU-time for a given error on the density. For instance, using $L_{\max} = 2$, the AB1 and AB1M schemes use the same number of average cells (254) and compute with the same accuracy ($l_t^1 l_x^1$-norm of the error on the density is 0.0041) but AB1M computes in 0.65 s against 1.17 s for AB1. It means that AB1M computes approximatively two times faster than AB1 which proves the efficiency of the local time stepping algorithm. The same statement is also observed for the second order scheme, see Table 1.

Table 1. Numerical order on the density error using the $l_t^1 l_x^1$-norm and l_x^1-norm at time $t = 0.4$ with respect to the averaged number of cells, and the $l_t^1 l_x^1$-norm on the density error with respect to the CPU-time (in the $\log_{10}$ scale).

Method	$\|\rho - \rho_{\text{ex}}\|_{l_t^1 l_x^1}$ *vs.* $N_{L_{\max}}$	$\|\rho - \rho_{\text{ex}}\|_{l_x^1}$ *vs.* $N_{L_{\max}}$	$\|\rho - \rho_{\text{ex}}\|_{l_t^1 l_x^1}$ *vs.* CPU-time
AB1U	0.70	0.72	0.41
AB1	1.86	1.93	0.64
AB1M	2.15	2.16	0.77
AB2U	0.77	0.77	0.42
AB2	1.74	1.84	0.57
AB2M	2.21	2.18	0.76
RK2U	0.76	0.76	0.42
RK2	1.68	1.80	0.55

5.2. Lax's shock tube problem

Lax's shock tube problem is similar to Sod's shock tube problem, except that the amplitude of the shock is stronger and the initial condition has a discontinuity in the velocity,

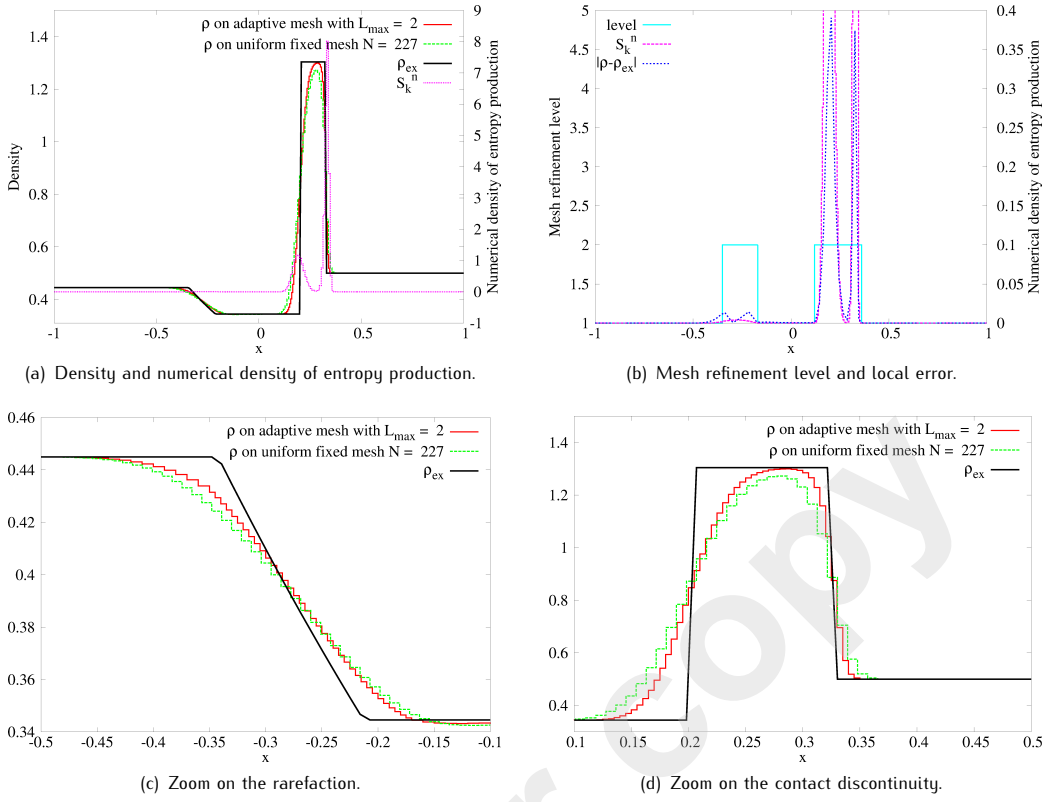
$$x \in [-1, 1], \qquad (\rho, u, p)(0, x) = \begin{cases} (0.445, 0.698, 3.528), & x \leqslant 0, \\ (0.5, 0, 0.571), & x > 0. \end{cases} \tag{15}$$

This second test is used in order to confirm similar conclusion obtained for Sod's shock tube problem. As in the previous case, the exact solution is known and computed from the Riemann problem (13) with the Riemann data (15) using 5 000 uniform cells.

The computational domain here is $[-1, 1]$ with the prescribed free boundary conditions. For each numerical computation, the following parameters have been used:

CFL :	0.5,
Simulation time (s) :	0.13,
Initial number of cells :	200,
Maximum level of mesh refinement :	$L_{\max}$.

[1] *Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz*



(a) Density and numerical density of entropy production.

(b) Mesh refinement level and local error.

(c) Zoom on the rarefaction.

(d) Zoom on the contact discontinuity.

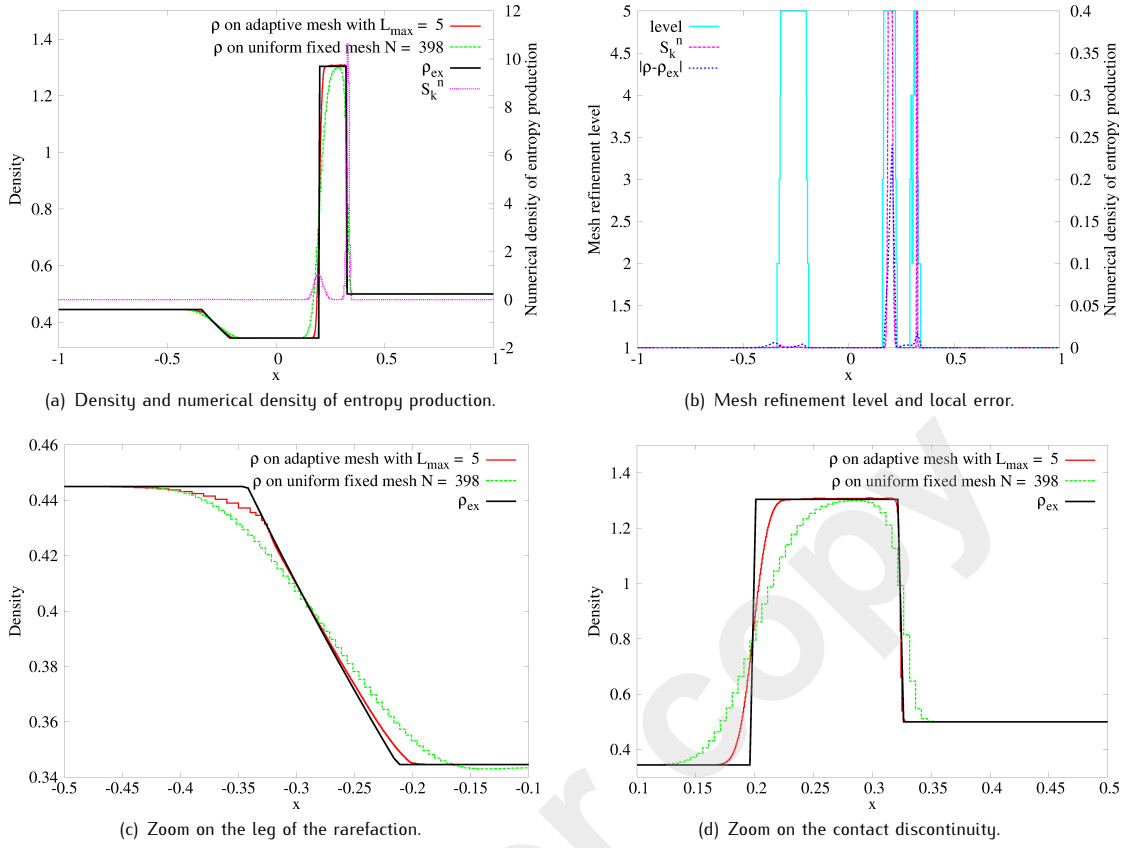
Figure 9. Lax's shock tube problem: solution at time $t = 0.13$ using the AB1M scheme with $L_{\max} = 2$.

As done before, we first represent the numerical solution compared to the exact one for levels $L_{\max} = 2$ and 5. In Figures 9–10, we represent the density of the solution at time $t = 0.13$, plot its numerical density of entropy production, the error made on the density and the level of refinement. We also perform, as done previously, the numerical order of the first and second order schemes in Figure 11(a)–11(d), and compare the accuracy versus the CPU-time in Figure 11(e)–11(f), and Table 2.

Even if Lax's shock tube problem introduces a huge shock wave with respect to Sod's shock tube problem, the same conclusion holds, namely, the numerical density of entropy production reproduces behavior of the error done as we can observe in Figures 9–10 and thus the mesh is automatically refined where it is necessary. Moreover, we observe again an improvement of the numerical approximation when the adaptive scheme is used and exceptional convergence of first order adaptive scheme, see Figure 11(a) and Table 2. Finally, the computational time is drastically reduced when the local time stepping algorithm is used, see Figure 11(e)–11(f), and Table 2.

5.3. The Shu–Osher test case

The next test case concerns the Shu and Osher problem [21] in which the Mach 3 shock moves to the right and collides with an entropy disturbance moving to the left. We focus on the solution at time $t = 0.18$ which develops three shocks in a highly oscillating regime. Thus, it is hard to compute an accurate numerical solution with the fixed uniform grid which requires a large number of cells to precisely capture oscillating regions. In this section we compare adaptive schemes of the first and second order. In particular, we numerically verify that for the given accuracy, the local time stepping algorithm is less time consuming than the standard one.



(a) Density and numerical density of entropy production.

(b) Mesh refinement level and local error.

(c) Zoom on the leg of the rarefaction.

(d) Zoom on the contact discontinuity.

Figure 10. Lax's shock tube problem: solution at time $t = 0.13$ using the AB1M scheme with $L_{\max} = 5$.

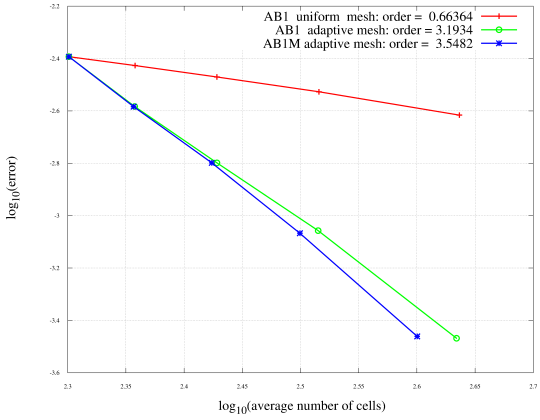
Table 2. Numerical order on the density error using the $l_t^1 l_x^1$-norm and l_x^1-norm at time $t = 0.13$ with respect to the averaged number of cells, and the $l_t^1 l_x^1$-norm on the density error with respect to the CPU-time (in the $\log_{10}$ scale).

Method	$\|\rho - \rho_{ex}\|_{l_t^1 l_x^1}$ *vs.* $N_{L_{\max}}$	$\|\rho - \rho_{ex}\|_{l_x^1}$ *vs.* $N_{L_{\max}}$	$\|\rho - \rho_{ex}\|_{l_t^1 l_x^1}$ *vs.* CPU-time
AB1U	0.66	0.69	0.27
AB1	3.20	3.31	0.66
AB1M	3.55	3.56	0.82
AB2U	0.60	0.62	0.32
AB2	3.24	3.40	0.70
AB2M	3.31	3.11	0.78
RK2U	0.73	0.70	0.38
RK2	3.25	3.43	0.65

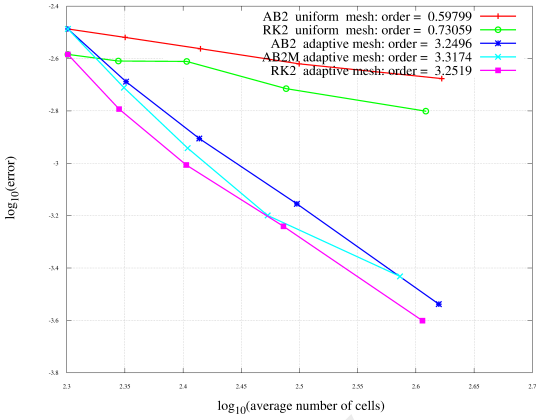
The Shu and Osher initial conditions are

$$(\rho, u, p)(0, x) = \begin{cases} (3.857143, 2.629369, 10.3333), & x \leqslant 0.1, \\ (1 + 0.2\sin(50x), 0, 1), & x > 0.1, \end{cases}$$

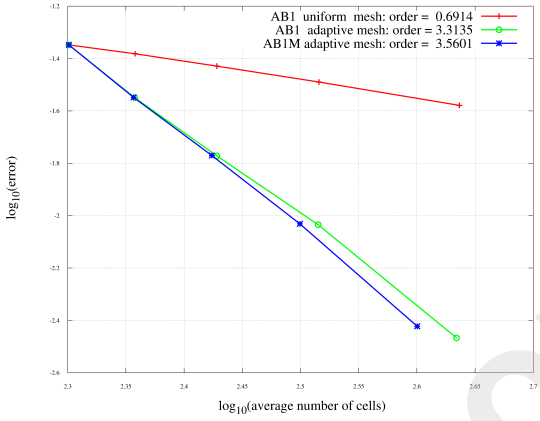
and the computational domain is $[0, 1]$ with the prescribed free boundary conditions.



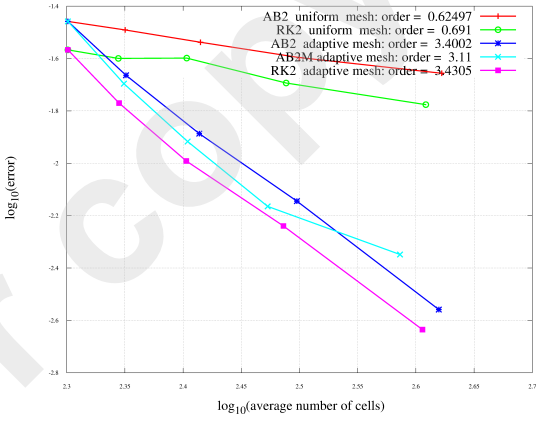
(a) $\|\rho_{ex}-\rho\|_{l_t^1 l_x^1}$ with respect to the averaged number of cells for the schemes of the first order.



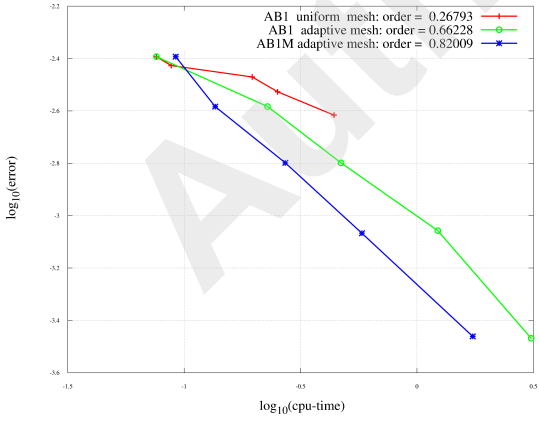
(b) $\|\rho_{ex}-\rho\|_{l_t^1 l_x^1}$ with respect to the averaged number of cells for the schemes of the second order.



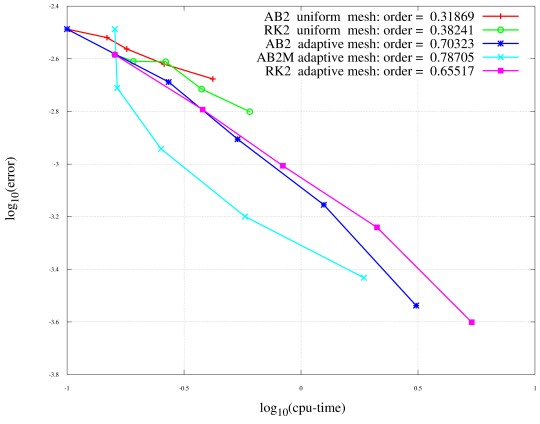
(c) $\|\rho_{ex}-\rho\|_{l_x^1}$ with respect to the averaged number of cells for the schemes of the first order at time $t = 0.13$.



(d) $\|\rho_{ex}-\rho\|_{l_x^1}$ with respect to the averaged number of cells for the schemes of the second order at time $t = 0.13$.



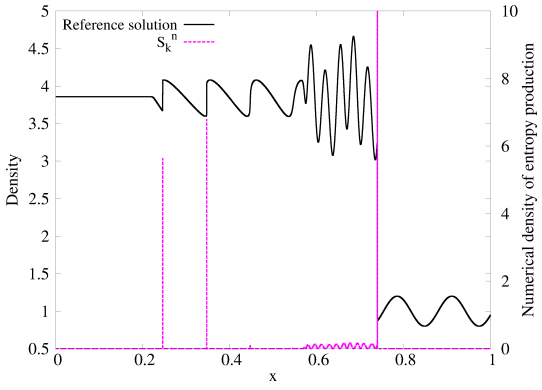
(e) CPU-time-$\|\rho_{ex}-\rho\|_{l_t^1 l_x^1}$ with respect to the CPU-time for the schemes of the first order.



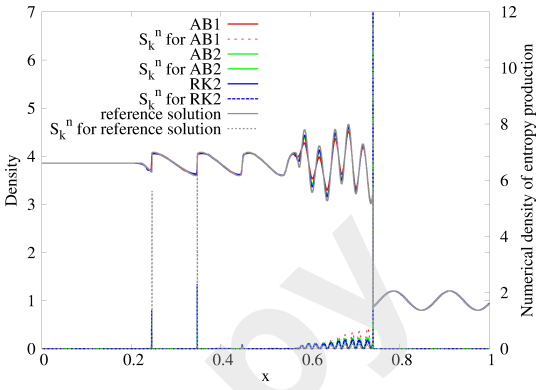
(f) CPU-time-$\|\rho_{ex}-\rho\|_{l_t^1 l_x^1}$ with respect to the CPU-time for the schemes of the second order.

Figure 11. Lax's shock tube problem: numerical order.

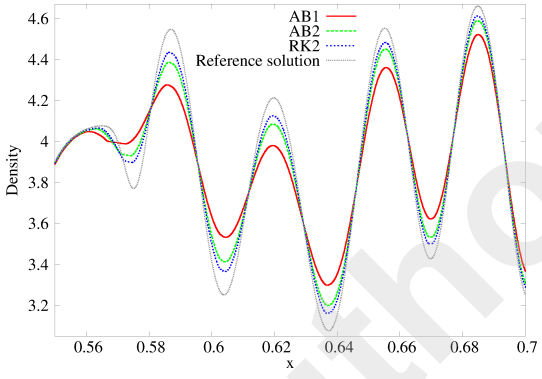
As a reference solution, we compute the solution on the uniform fixed grid (20 000 cells) with the RK2 scheme. The density of the reference solution at time $t = 0.18$ and its numerical density of entropy production are displayed in Figure 12(a). This solution being computed on a very fine fixed grid, as predicted by the theory, the density of entropy production is almost concentrated at the shocks. Even if small productions are present between $0.5 \leqslant x \leqslant 0.75$, one can consider such solution as an "exact" one.



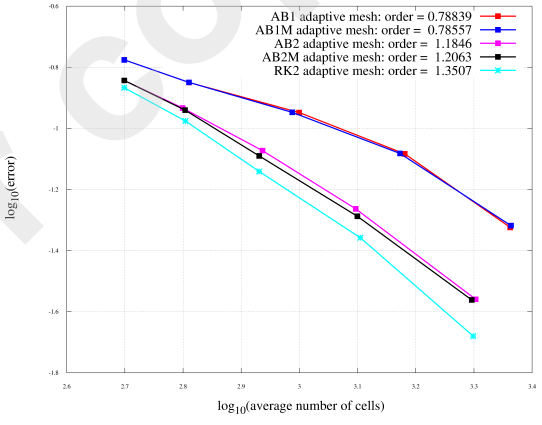
(a) Reference solution.



(b) Density and numerical density of entropy production.



(c) Zoom on oscillating region.



(d) Numerical order: $\|\rho_{ex} - \rho\|_{l_x^1}$ with respect to the averaged number of cells for the schemes of the first and second order at time $t = 0.18$.

Figure 12. The Shu–Osher test case.

For each numerical computation, we have used the following parameters:

CFL :	0.219,
Simulation time (s) :	0.18,
Initial number of cells :	500,
Maximum level of mesh refinement :	$L_{\max} = 4$.

In Figure 12(b), we plot the density of the reference solution, the one by the AB1, AB2 and RK2 schemes and their numerical density of entropy production. Starting from 500 cells, the adaptive schemes lead to very close solutions for each scheme and the numerical density of entropy production vanishes everywhere where the solution is smooth and every solution fits to the reference solution. However, focusing closely on the oscillating area between $0.5 \leqslant x \leqslant 0.7$, one can observe that, as expected, RK2 is better than AB2 which is better than AB1. These results are also confirmed

by the computation of the numerical order displayed in Figure 12(d): it was not true in the two previous test cases, see Figures 8–11, as emphasized in subsection 5.1 (for which the numerical order for AB1, for instance, was greater than the second order methods).

Table 3 summarizes the computation of the total entropy production, the discrete l_x^1-norm of the error on the density, the CPU-time, the average number of cells and the maximum number of cells at the final time $t = 0.18$ for the level $L_{\max} = 4$.

Table 3. Comparison of numerical schemes of the first and second order.

Method	$\mathcal{P}$	$\|\rho - \rho_{\text{ref}}\|_{l_x^1}$	CPU-time	$N_{L_{\max}}$	maximum number of cells
AB1	0.288	$4.74\,10^{-2}$	181	1574	2308
AB1M	0.288	$4.80\,10^{-2}$	120	1572	2314
AB2	0.287	$2.75\,10^{-2}$	170	1391	2023
AB2M	0.286	$2.74\,10^{-2}$	108	1357	1994
RK2	0.285	$2.08\,10^{-2}$	299	1375	2005

As said before, the hierarchy in terms of accuracy is very well respected as we also see in Table 3 for the error on density or the total production of entropy. We see again the case where the total production of entropy is an accuracy indicator.

It is well known that the AB2 scheme is less stable and less accurate than the RK2 scheme. Nonetheless, in the framework of the local time stepping, for almost the same accuracy the AB2M scheme computes three times faster than the RK2 which is a significant gain in time.

6. Concluding remarks and perspectives

In this paper, first and second order methods in space and time are coupled with an adaptive algorithm employing the local time stepping, obtaining the adaptive numerical scheme in which the grid is locally refined or coarsened according to the entropy indicator. Several numerical tests have been performed, they show an impressive improvement with respect to uniform grids even if a large number of cells is used.

All numerical tests also show that the numerical density of entropy production combined with the proposed mesh refinement parameter is a relevant local error indicator (everywhere where the solution remains smooth) and discontinuity detector (large shocks and oscillating solutions are very well-captured). Moreover, we have shown that the implementation of the local time stepping algorithm can significantly reduce the computational time keeping the order of accuracy unchanged.

We considered the Tang–Warnecke and the Osher–Sanders projections. We have shown that even if the Tang–Warnecke projection is consistent, the solution develops numerical instabilities at a large maximum level of refinement. This statement has been numerically reproduced and we have proposed a numerical smoothing grid technique which prevents two adjacent cells from having a level difference greater than two. As a consequence the amplitude of such oscillation is minimized. Nevertheless, since both methods lead to oscillating solutions, the Osher–Sanders projection offers a good compromise between accuracy and computing cost reduction. Finally, we plan to improve the efficiency of the adaptive scheme to capture accurately the contact discontinuities and to extend this work for 2D/3D numerical applications.

Acknowledgements

The authors wish to thank the referees for their careful reading of the initial version of the manuscript and useful remarks.

This work is supported by the “Agence Nationale de la Recherche” through the COSINUS program (ANR CARPEiNTER project No. ANR-08-COSI-002).

References

[1] Allahviranloo T., Ahmady N., Ahmady E., Numerical solution of fuzzy differential equations by predictor–corrector method, Inform. Sci., 2007, 177(7), 1633–1647

[2] Altmann C., Belat T., Gutnic M., Helluy P., Mathis H., Sonnendrücker É., Angulo W., Hérard J.-M., A local time-stepping discontinuous Galerkin algorithm for the MHD system, In: CEMRACS 2008 — Modelling and Numerical Simulation of Complex Fluids, Marseille, July 21–August 29, 2008, ESAIM Proc., 28, EDP Sciences, Les Ulis, 2009, 33–54

[3] Berger M.J., Oliger J., Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys., 1984, 53(3), 484–512

[4] Cockburn B., Gremaud P.-A., A priori error estimates for numerical methods for scalar conservation laws. Part II: Flux-splitting monotone schemes on irregular Cartesian grids, Math. Comp., 1997, 66(218), 547–572

[5] Croisille J.-P., Contribution à l'Étude Théorique et à l'Approximation par Éléments Finis du Système Hyperbolique de la Dynamique des Gaz Multidimensionnelle et Multiespèces, PhD thesis, Université de Paris VI, 1991

[6] Eymard R., Gallouët T., Herbin R., Finite Volume Methods, In: Handbook of Numerical Analysis, VII, Handb. Numer. Anal., VII, North-Holland, Amsterdam, 2000, 713–1020

[7] Gallouët T., Hérard J.-M., Seguin N., Some recent finite volume schemes to compute Euler equations using real gas EOS, Internat. J. Numer. Methods Fluids, 2002, 39(12), 1073–1138

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

[9] Golay F., Numerical entropy production and error indicator for compressible flows, Comptes Rendus Mécanique, 2009, 337(4), 233–237

[10] Guermond J.-L., Pasquetti R., Popov B., Entropy viscosity method for nonlinear conservation laws, J. Comput. Phys., 2011, 230(11), 4248–4267

[11] Hairer E., Nørsett S.P., Wanner G., Solving Ordinary Differential Equations. I, 2nd ed., Springer Ser. Comput. Math., 8, Springer, Berlin, 1993

[12] Houston P., Mackenzie J.A., Süli E., Warnecke G., A posteriori error analysis for numerical approximations of Friedrichs systems, Numer. Math., 1999, 82(3), 433–470

[13] Jenny P., Lee S.H., Tchelepi H.A., Adaptive multiscale finite-volume method for multiphase flow and transport in porous media, Multiscale Model. Simul., 2005, 3(1), 50–64

[14] Karni S., Kurganov A., Local error analysis for approximate solutions of hyperbolic conservation laws, Adv. Comput. Math., 2005, 22(1), 79–99

[15] Karni S., Kurganov A., Petrova G., A smoothness indicator for adaptive algorithms for hyperbolic systems, J. Comput. Phys., 2002, 178(2), 323–341

[16] Müller S., Stiriba Y., Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping, J. Sci. Comput., 2007, 30(3), 493–531

[17] Osher S., Sanders R., Numerical approximations to nonlinear conservation laws with locally varying time and space grids, Math. Comp., 1983, 41(164), 321–336

[18] Puppo G., Numerical entropy production on shocks and smooth transitions, J. Sci. Comput., 2002, 17(1-4), 263–271

[19] Puppo G., Numerical entropy production for central schemes, SIAM J. Sci. Comput., 2004, 25(4), 1382–1415

[20] Puppo G., Semplice M., Numerical entropy and adaptivity for finite volume schemes, Commun. Comput. Phys., 2011, 10(5), 1132–1160

[21] Shu C.-W., Osher S., Efficient implementation of essentially nonoscillatory shock-capturing schemes, J. Comput. Phys., 1988, 77(2), 439–471

[22] Simeoni C., Remarks on the consistency of upwind source at interface schemes on nonuniform grids, J. Sci. Comput., 2011, 48(1-3), 333–338

[23] Sod G.A., A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comput. Phys., 1978, 27(1), 1–31

[24] Sonar T., Hannemann V., Hempel D. Dynamic adaptivity and residual control in unsteady compressible flow computation, In: Theory and Numerical Methods for Initial-Boundary Value Problems, Math. Comput. Modelling, 1994, 20(10-11), 201–213
[25] Tan Z., Zhang Z., Huang Y., Tang T., Moving mesh methods with locally varying time steps, J. Comput. Phys., 2004, 200(1), 347–367
[26] Tang H., Warnecke G., A class of high resolution difference schemes for nonlinear Hamilton–Jacobi equations with varying time and space grids, SIAM J. Sci. Comput., 2005, 26(4), 1415–1431
[27] Toro E.F., Riemann Solvers and Numerical Methods for Fluid Dynamics, 2nd ed., Springer, Berlin, 1999
[28] Zhang X.D., Trépanier J.-Y., Camarero R., A posteriori error estimation for finite-volume solutions of hyperbolic conservation laws, Comput. Methods Appl. Engrg., 2000, 185(1), 1–19