



**HAL**  
open science

# Adaptive multi scale scheme based on numerical density of entropy production for conservation laws

Mehmet Ersoy, Frederic Golay, Lyudmyla Yushchenko

## ► To cite this version:

Mehmet Ersoy, Frederic Golay, Lyudmyla Yushchenko. Adaptive multi scale scheme based on numerical density of entropy production for conservation laws. *Central European Journal of Mathematics*, 2013, 11 (8), pp.1392-1415. 10.2478/s11533-013-0252-6 . hal-01338176

**HAL Id: hal-01338176**

**<https://hal.science/hal-01338176v1>**

Submitted on 28 Jun 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Adaptive multi scale scheme based on numerical density of entropy production for conservation laws.

Mehmet Ersoy<sup>\*1</sup>, Frédéric Golay<sup>†1</sup>, and Lyudmyla Yushchenko<sup>‡1</sup>

<sup>1</sup> Université de Toulon, IMATH, EA 2134, 83957 La Garde Cedex, France.

January 8, 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). Thus it is used as an *a posteriori* error which provides information on the need to refine the mesh in the regions where discontinuities occur and to coarsen the mesh in the regions where the solution remains smooth. Nevertheless, due to the CFL stability condition the time step is restricted and leads to time consuming simulations. Therefore, we propose a local time stepping algorithm. We also use high order time extensions using Adams-Bashforth time integration technique as well as a second order linear reconstruction in space. We numerically investigate the efficiency of the scheme through several test cases: the Sod's shock tube problem, the Lax's shock tube problem and the Shu-Osher test problem.

*Keywords:* Hyperbolic system, finite volume scheme, local mesh refinement, numerical density of entropy production, local time stepping.

## 1 Introduction

In this paper, we are interested in numerical integration of one dimensional non linear hyperbolic systems of conservation laws of the form

$$\begin{cases} \frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{w})}{\partial x} = 0, & (t, x) \in \mathbb{R}^+ \times \mathbb{R} \\ \mathbf{w}(0, x) = \mathbf{w}_0(x), & x \in \mathbb{R}. \end{cases} \quad (1)$$

where  $\mathbf{w} : \mathbb{R}^+ \times \mathbb{R} \rightarrow \mathbb{R}^d$  stands for the vector state and  $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  the flux function.

Solving Equation (1) with high accuracy is a challenging problem since it is well-known that solutions can and will breakdown at a finite time, even if the initial data are smooth, and develop complex structure (shock wave interactions). In such a situation, the uniqueness of the (weak) solution is lost and is recovered by completing the system (1) with an entropy inequality of the form:

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

where  $(s, \psi)$  stands for a convex entropy-entropy flux pair. This inequality allows to select the physical relevant solution. Moreover, the entropy satisfies a conservation equation only in regions where the solution

---

<sup>\*</sup> *Mehmet.Ersoy@univ-tln.fr*

<sup>†</sup> *Frederic.Golay@univ-tln.fr*

<sup>‡</sup> *Lyudmyla.Yushchenko@univ-tln.fr*

is smooth and an inequality when the solution develops shocks. In simple cases, it can be proved that the term missing in (2) to make it an equality is a Dirac mass.

Numerical approximation of Equations (1) with (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, 15, 14]). As a consequence, the numerical density of entropy production provides information on the need to locally refine the mesh (e.g. if the solution develops discontinuities) or to coarsen the mesh (e.g. if the solution is smooth and well-approximated) as already used by Puppo [18, 19, 20] and Golay [9]. Even if the shocks are well-captured on coarse grid using finite volume scheme such indicator is able (as shown in Puppo [20] and the reference therein) not only to provide an efficient *a posteriori* error, but also to reproduce the qualitative structure of the solution and to pilot the adaptive scheme. Let us also mention that this does not occur efficiently on contact discontinuities where only numerical diffusion is active. Nonetheless, we mention the work by Guermond *et al.* [10] as 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 a 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 a CFL stability condition. The cpu-time increases rapidly as the mesh is refined since the CFL stability condition imposes an upper bound on  $\frac{\delta t}{h}$  where  $\delta t$  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. [16, 25, 2]). 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 [17, 24, 13, 26, 16, 20]).

The aim of this paper is to construct a 1D multi scale 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 we fix notations. Thereafter, we explain how to adapt the space grid using the numerical density of entropy production in Section 3. Combining both time integration and mesh refinement, we show how to implement an efficient multi scale 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, the efficiency of the multi scale adaptive scheme is shown through Sod's shock tube problem, Lax's shock tube problem and 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 a numerical approximation of the following general non linear hyperbolic conservation laws

$$\begin{cases} \frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{w})}{\partial x} = 0, (t, x) \in \mathbb{R}^+ \times \Omega \\ \mathbf{w}(0, x) = \mathbf{w}_0(x), x \in \mathbb{R}. \end{cases} \quad (3)$$

where  $\Omega \subset \mathbb{R}$ ,  $\mathbf{w} \in \mathbb{R}^d$  denotes the vector state (referred as the conservative variables, e.g. Equations (14)–(17)) and  $\mathbf{f}$  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  $x_{k\pm 1/2} = x_k \pm h_k/2$ . The unknowns  $\mathbf{w}(t, x)$  are approximated by their mean values on the cell  $C_k$  at time  $t$ :

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

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

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

where  $\mathbf{F}_{k\pm 1/2}$  stands for the numerical flux at the interface  $x_{k\pm 1/2}$  (see for instance [8, 27, 6]).

For the convenience of the reader, noting the flux difference

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

between two interfaces of a cell  $C_k$ , Equation (4) is also written:

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

For practical purpose, in our case,  $\mathbf{F} = \mathbf{f}(\mathbf{R}(0^\pm, \mathbf{w}_L, \mathbf{w}_R))$  is determined from the exact solution  $\mathbf{R}(0^\pm, \mathbf{w}_L, \mathbf{w}_R)$  of the Riemann problem associated with the left  $\mathbf{w}_L$  and the right  $\mathbf{w}_R$  state:

$$\begin{cases} \frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{w})}{\partial x} = 0, \\ \mathbf{w}_0(x) = \begin{cases} \mathbf{w}_L & \text{if } x < 0, \\ \mathbf{w}_R & \text{if } x > 0. \end{cases} \end{cases} \quad (6)$$

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

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

Equation (3) is completed with an entropy inequality of the form:

$$\frac{\partial s(\mathbf{w})}{\partial t} + \frac{\partial \psi(\mathbf{w})}{\partial x} \leq 0, \quad (7)$$

where  $(s, \psi)$  stands for a convex entropy-entropy flux pair. This inequality not only allows to select the physical relevant solution but will be used to define a relevant mesh refinement criterion (the so-called *numerical density of entropy production*, see Section 3). To this end, Equation (7) is discretized using the same scheme as Equation (3). Thus, the semi discrete formulation for the entropy inequality reads:

$$h_k \frac{\partial s(\mathbf{w}_k)}{\partial t} + \delta \psi_k(t) := h_k \frac{\partial s(\mathbf{w}_k)}{\partial t} + \psi(\mathbf{R}(0^-, \mathbf{w}_k(t), \mathbf{w}_{k+1}(t))) - \psi(\mathbf{R}(0^+, \mathbf{w}_{k-1}(t), \mathbf{w}_k(t))) \quad (8)$$

where

$$\delta \psi_k(t) := \psi_{k+1/2}(t) - \psi_{k-1/2}(t)$$

and  $\mathbf{R}$  is the solution of the Riemann problem (6).

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

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

where  $\mathcal{M}_k$  is an approximation to the slope obtained with the MinMod limiter :

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

and

$$\Delta \mathbf{w}_{k-1/2} := \mathbf{w}_k - \mathbf{w}_{k-1}, \quad \Delta \mathbf{w}_{k+1/2} := \mathbf{w}_{k+1} - \mathbf{w}_k, \quad \Delta \mathbf{w}_k := \frac{\Delta \mathbf{w}_{k-1/2} + \Delta \mathbf{w}_{k+1/2}}{2}.$$

Thus as already described before, we define the numerical flux as

$$\begin{aligned} \mathbf{F}_{k+1/2} &:= \mathbf{f} \left( \mathbf{R} \left( 0^-, \mathbf{w}_k + \frac{\mathcal{M}_k}{2}, \mathbf{w}_{k+1} - \frac{\mathcal{M}_{k+1}}{2} \right) \right), \\ \mathbf{F}_{k-1/2} &:= \mathbf{f} \left( \mathbf{R} \left( 0^+, \mathbf{w}_{k-1} + \frac{\mathcal{M}_{k-1}}{2}, \mathbf{w}_k - \frac{\mathcal{M}_k}{2} \right) \right). \end{aligned}$$

## 2.2 Time integration

Now we focus on the numerical time integration of Equations (5) and (8). The well-known time integration method of Runge-Kutta and Adams-Bashforth are recalled to fix notations.

### 2.2.1 Runge Kutta schemes

Integrating Equation (5) and (8) during the time step  $]t_n, t_{n+1}[$  of length  $\delta t_n$  and evaluating the numerical fluxes at time  $t_n$ , we obtain the well-known Euler's scheme:

$$\begin{aligned} \mathbf{w}_k(t_{n+1}) &= \mathbf{w}_k(t_n) - \frac{\delta t_n}{h_k} \delta \mathbf{F}_k(t_n) , \\ s(\mathbf{w}_k(t_{n+1})) &= s(\mathbf{w}_k(t_n)) - \frac{\delta t_n}{h_k} \delta \psi_k(t_n) . \end{aligned} \quad (9)$$

In order to increase the accuracy, a second order method can be used as follows

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

where

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

and the time discretization of the entropy is the same as the time discretization of the conservative variables.

Then, as in [20], we define the *numerical density of entropy production*  $S_k^n$ :

$$S_k^n := \frac{s(\mathbf{w}_k(t_{n+1})) - s(\mathbf{w}_k(t_n))}{\delta t_n} + \frac{\delta \psi_k(t_n)}{h_k} \quad (10)$$

for the first order scheme and

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

for the second order scheme.

We note

$$\mathcal{P} = \sum_{n,k} S_k^n \delta t_n h_k \quad (11)$$

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 *multi time step method* such as the Adams-Bashforth method. The Adams-Bashforth method of order  $m$  consists in replacing the numerical flux of Equation (5) by a polynomial interpolation of the same order [11]:

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

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 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 equation (5) and (8) leads to:

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

For example, the second order Adams-Bashforth method is:

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

The first order method is exactly as Equations (9). The Adams-Bashforth methods of order 2 and 3 are stable for Courant-Friedrich-Levy condition (CFL) less than one [1]. Practically, for stability purpose, we limit our applications to the second order scheme.

We also perform the same discretization above for Equation (8). As done in the case of the Runge-Kutta scheme, we define the numerical density of entropy production  $S_k^n$  by (10) for the first order scheme and

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

for the second order scheme. Finally, the total numerical entropy production is given by (11).

### 3 Mesh refinement

In order to compute with accuracy the solution of hyperbolic systems, a strategy can be to adapt automatically the mesh following a local error indicator (defined later on).

#### 3.1 Mesh refinement criterion

Many works concentrate on *a posteriori* error estimate, which are constructed from mathematical arguments [28]. But paradoxically, to our knowledge, very few works use a physical criterion as [5]: the numerical density of entropy production. More recently, Puppo [18, 19] has proved that the numerical density of entropy production can be used as a discontinuity indicator and a local error indicator. Indeed, the entropy inequality being related mathematically and physically to the system (3), the numerical density of entropy production should be (and actually is, as we will see in Section 5) an accurate and a useful mesh refinement criterion. Let us show how to use it in an automatic mesh refinement framework.

The mesh refinement criterion is the numerical density of entropy production  $S_k^n$  (for instance, defined by (10) for a first order scheme, see Section 2.2 for further details) obtained from the discretization of the entropy inequality

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

where  $\psi$  is the entropy flux with

$$\nabla_{\mathbf{w}}^T \psi = \nabla_{\mathbf{w}}^T s(\mathbf{w}) f'(\mathbf{w}) .$$

As pointed out before in [18, 19], the previous inequality is useful for numerical application both as a local error indicator (whenever it fails to be zero for smooth flow) and discontinuity detector (whenever solution develops shocks or contact discontinuities). Thus it will be used here as a mesh refinement criterion.

#### 3.2 Mesh refinement process

We present here a 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 “macro cells” which could be refined by generating hierarchical grids. Each cell can be split in 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. For the convenience of the reader, we make use of the following notations: let  $k_b$  be the index which makes reference to the macro cell numbered  $k$  and  $b$  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 the length( $b$ )  $- 1$ . For instance, a 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.

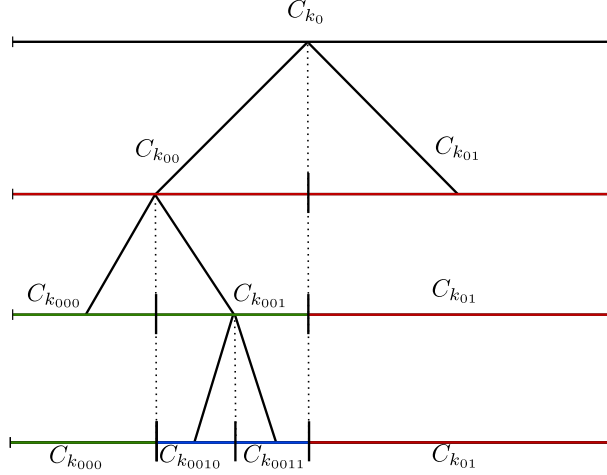


Figure 1: Example of hierarchical dyadic tree.

The mesh refinement procedure can be simply expressed as follows. We first set a mesh refinement parameter  $\bar{S}$ . For instance, it can be the mean value over the domain  $\Omega$  at time  $t_n$ :

$$\bar{S} = \frac{1}{|\Omega|} \sum_{k_b} S_{k_b}^n . \quad (12)$$

We then define two coefficients  $0 \leq \alpha_{\min} \leq \alpha_{\max} \leq 1$ , which determine the ratio of numerical production of entropy leading to mesh refinement or mesh coarsening.

Thus, for each cell  $C_{k_b}$ :

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

On one hand, if a cell  $C_{k_b}$  is split into two sub-cells  $C_{k_{b0}}$  and  $C_{k_{b1}}$ , averaged values at time  $t_n$  are projected on each sub-cell (for the first order scheme):

$$\mathbf{w}_{k_{b0}}^n = \mathbf{w}_{k_{b1}}^n = \mathbf{w}_{k_b}^n$$

and the numerical fluxes between the two sub-cells (see FIGURE 2(a)) are defined as:

$$\begin{cases} \mathbf{F}_{k_{b0}-1/2}^n = \mathbf{F}_{k_b-1/2}^n , \\ \mathbf{F}_{k_{b0}+1/2}^n = \mathbf{F}_{k_{b1}-1/2}^n = \mathbf{f}(\mathbf{w}_{k_b}^n) , \\ \mathbf{F}_{k_{b1}+1/2}^n = \mathbf{F}_{k_b+1/2}^n . \end{cases}$$

In the case of second order method, as the gradient is available, we also use:

$$\begin{cases} \mathbf{w}_{k_{b0}}^n = \mathbf{w}_{k_b}^n - \frac{h_{k_b}}{4} \frac{\partial \mathbf{w}_{k_b}^n}{\partial x} , \\ \mathbf{w}_{k_{b1}}^n = \mathbf{w}_{k_b}^n + \frac{h_{k_b}}{4} \frac{\partial \mathbf{w}_{k_b}^n}{\partial x} . \end{cases}$$

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

$$\mathbf{w}_{k_b}^n = \frac{1}{2} (\mathbf{w}_{k_{b0}}^n + \mathbf{w}_{k_{b1}}^n)$$

and

$$\begin{cases} \mathbf{F}_{k_b-1/2}^n = \mathbf{F}_{k_{b0}-1/2}^n , \\ \mathbf{F}_{k_b+1/2}^n = \mathbf{F}_{k_{b1}+1/2}^n . \end{cases}$$

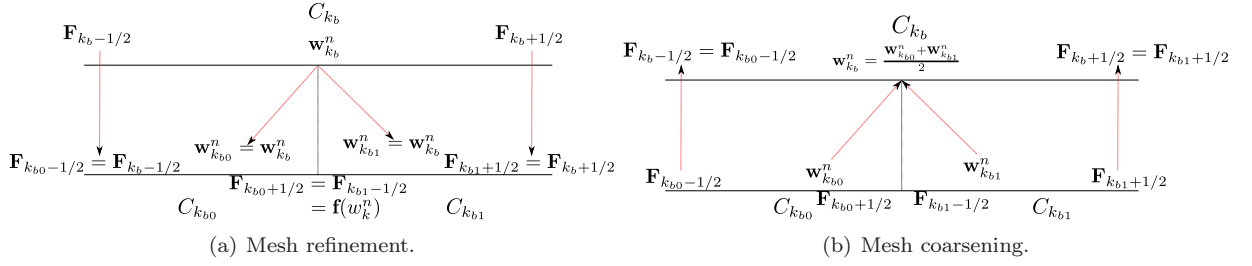


Figure 2: Mesh refinement and mesh coarsening: construction of the fluxes.

**Remark 3.1** *It is hard to define an optimal mesh refinement criterion since the less the threshold is small, the more the mesh is refined unnecessarily and the more the threshold is large, the less the mesh is refined. Therefore, a good compromise is to use the average value of the mesh refinement criterion (12). The others threshold parameters  $\alpha_{\min}$  and  $\alpha_{\max}$  allow to set a percentage of mesh refinement and mesh coarsening with respect to the quantity  $\bar{S}$ . It is not surprising that these settings will deteriorate or improve the accuracy of the numerical solution. For instance, the more  $\alpha_{\min}$  and  $\alpha_{\max}$  are small, 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. The main approach to update averaged quantities will be recalled and the general algorithm will be provided.

### 4.1 The local time stepping algorithm

In this section, 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 "level", based on their diameter. Thus, the level of refinement  $L_k$  of the  $k^{\text{th}}$  cell  $C_k$  is defined by:

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



where  $N$  stands for the maximum level

$$N = \log_2 \left( \frac{h_{\max}}{h_{\min}} \right) + 1.$$

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 a dyadic mesh, we have:*

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

Let  $\delta t_n$  be the minimum time step at time  $t_n$  according to the CFL condition associated with the smallest cell. The macro time step  $\Delta t_n$  is then defined by:

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

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

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

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

```

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  $L_{k+1/2} \geq N - j$  do
    1. compute the integral of  $\mathbf{F}_{k+1/2}(t)$  on the time interval  $2^{N-L_{k+1/2}} \delta t_n$ ,
    2. distribute  $\mathbf{F}_{k+1/2}(t_n)$  to the two adjacent cells,
    3. update only the cells of level greater than  $N - j$ .
  end
end

```

**Remark 4.2** *We can notice that according to this algorithm, the flow is computed more times in the 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*

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

where  $\nu$  is an eigenvalue of the convection matrix  $D_{\mathbf{w}} \mathbf{f}(\mathbf{w})$ .

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

Let us illustrate for a global time advancement the local time stepping algorithm for a first order scheme (the principle for a 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 a mesh composed of 4 cells of levels 1, 2 and 3 respectively, i.e. here  $N = 3$ , as displayed on FIGURE 3. Let us note  $\mathbf{w}_m^n$  the state vector at time  $t_n$ . At the first stage (see FIGURE 3(a)),  $i = 1$ , only the small cells are advanced with a step  $\delta t^n$ , i.e.  $\mathbf{w}_3^{n+1} := \mathbf{w}_3(t_n + \delta t^n)$  and  $\mathbf{w}_4^{n+1} := \mathbf{w}_4(t_n + \delta t^n)$  are computed. While  $i = 2$  (see FIGURE 3(b)) concerns the small cells of level  $L_N$  and the cells immediately larger at level  $L_{N-1}$ . The cells  $L_N$  travel with a time step  $\delta t_n$ , while the cells  $L_{N-1}$  travel with a time step  $2\delta t_n$ , etc.

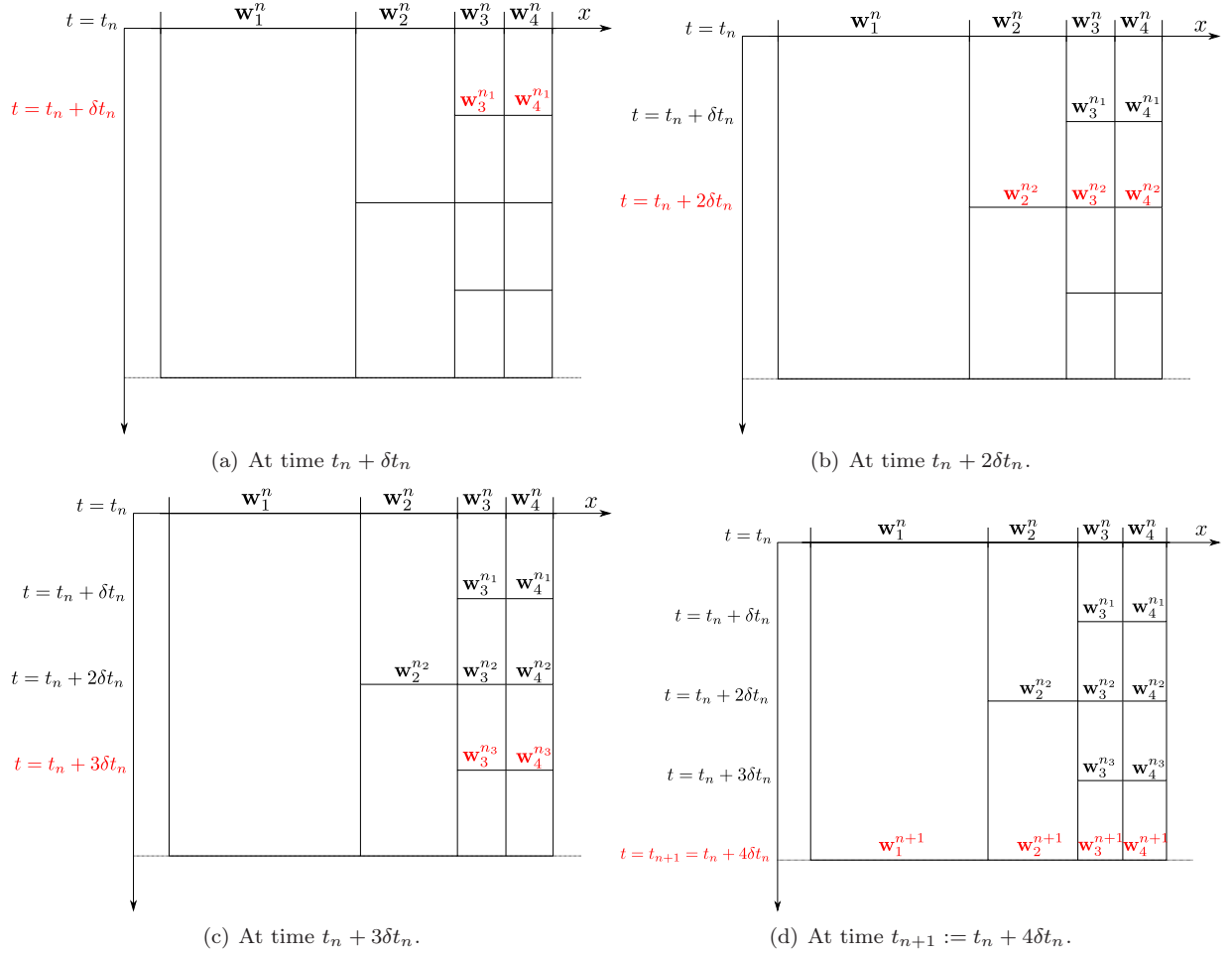
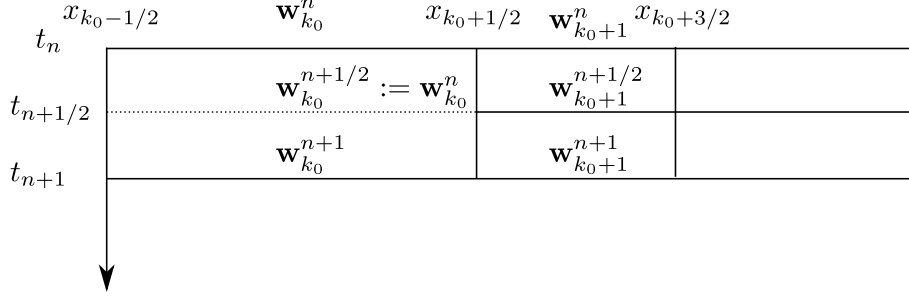


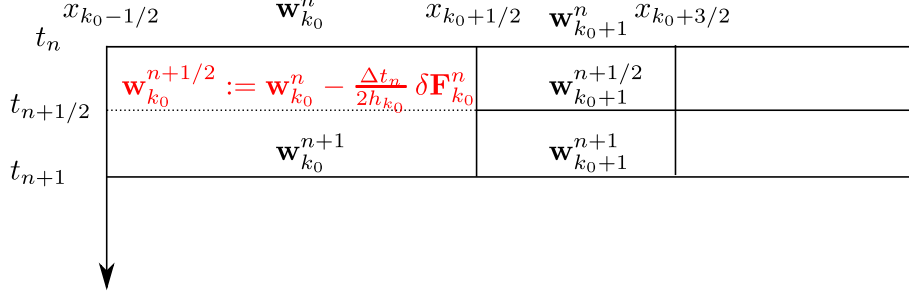
Figure 3: Illustration of a global time advancement with a first order scheme.

## 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 the 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  $k \leq k_0$ , 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 on 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 an initial condition with compact support.



(a) Osher and Sanders projection.



(b) Tang and Warnecke projection.

Figure 4: Projection's methods.

#### 4.2.1 A simple projection method

The Osher and Sanders scheme is defined as follows (see FIGURE 4(a)):

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

and

$$\forall k \in \mathbb{Z}, \mathbf{w}_k^{n+1} = \mathbf{w}_k^n - \frac{\Delta t_n}{2h_k} \left( \delta \mathbf{F}_k^n + \delta \mathbf{F}_k^{n+1/2} \right). \quad (13)$$

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

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

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

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

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]):

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

with a positive constant  $c$ . In that case, using the Godunov solver, the scheme (13) writes for  $k = k_0$ :

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

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

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

Thus, if  $\frac{\Delta t}{h}$  converge 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:

$$\forall k \in \mathbb{Z}, \mathbf{w}_k^{n+1/2} = \mathbf{w}_k^n - \frac{\Delta t_n}{2h_k} \delta \mathbf{F}_k^n$$

and

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

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

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

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 non conservative and non consistent 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 Section 5.1.1 and see also for instance [4, 22] for interesting issues). On the contrary, Puppo and Semplice [20] do not loose the consistency and the conservativity because their procedure does not produce mass and the fluxes are well computed at the cell interface between two levels of refinement. Nevertheless, their approach requires more evaluations 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 multi scale scheme* constructed in Section 3 and Section 4. The discussion is limited on studying the robustness, accuracy and the gain of cpu-time of several schemes. To this end, we compare results to those from the exact solution or a reference solution. These include the Sod's shock tube problem, the Lax's shock tube problem, and the Shu-Osher test case.

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

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0 \tag{14}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} = 0 \tag{15}$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial (\rho E + p) u}{\partial x} = 0 \tag{16}$$

$$p = (\gamma - 1) \rho \varepsilon \tag{17}$$

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

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

and the entropy flux by

$$\psi(\mathbf{w}) = u s(\mathbf{w})$$

for Equations (14)–(17).

## Notations and settings

- In what follows, we perform several numerical tests using first and second order schemes. Thus, we will refer to AB1 as the first order scheme, AB2 as the second order Adams-Basforth scheme, RK2 as the second order Runge-Kutta scheme. AB2 and RK2 use a MUSCL reconstruction. Moreover, all computations are made with a dynamic grid except if the acronym ends with the capital letter “U” which refers to a uniform fixed grid. All computations are made with an 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 to the one computed on a uniform fixed 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 a simulation of an adaptive scheme with a maximum level  $L_{\max}$ .
- All of the presented results display the density which is positive. Thus, for the convenience of the reader, 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 few iterations (except for the numerical solution computed on uniform fixed grid) and we have used the following threshold parameters:

$$\begin{aligned} \text{Mesh refinement parameter } \alpha_{\max} & : 0.01 , \\ \text{Mesh coarsening parameter } \alpha_{\min} & : 0.001 , \\ \text{Mesh refinement parameter } \bar{S} & : \frac{1}{|\Omega|} \sum_{k_b} S_{k_b}^n . \end{aligned}$$

- $\rho_{\text{ex}}$  will make reference to the exact solution of the Sod’s problem or the Lax’s problem.

## 5.1 The 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 an exact solution exists which enables a detailed comparison between the approximate results and the exact solution.

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

$$x \in [-1, 1], \quad (\rho, u, p)(0, x) = \begin{cases} (1, 0, 1), & x \leq 0, \\ (0.125, 0, 0.1), & x > 0 . \end{cases} \quad (18)$$

It consists of two non linear waves (one right shock and one 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 and therefore it is used as a 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 Section 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 observed (see FIGURE 5(b)-FIGURE 5(c)). The same behavior is also obtained for the OS projection as displayed on FIGURE 5(a)-FIGURE 5(c). We can reduce these effects using a 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 the 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.

We have without smoothing effect:

cpu-time for TW projection	=	330.27 ( $s$ ),
cpu-time for OS projection	=	85.02 ( $s$ ),

and with smoothing effect:

cpu-time for TW projection	=	288.21 ( $s$ ),
cpu-time for OS projection	=	80.04 ( $s$ ).

As displayed on FIGURE 5 and cpu performance above, even if the results using the TW projection are more accurate than the OS one: the TW projection is 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 appear when  $L_{\max}$  is large. In order to avoid such oscillations, we will limit to  $L_{\max} \leq 5$  which is large enough for practical applications.

### 5.1.2 Numerical Convergence

We study the numerical convergence of AB1, AB1M, AB2, AB2M and RK2 schemes. We compare the cpu-time and the accuracy of the numerical solutions for level  $1 \leq L_{\max} \leq 5$  (i.e. on fixed and dynamic grid). The analytical solution of the Riemann problem (14)–(17) with Riemann data (18) 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 (12) 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  $\rho$  for levels  $L_{\max} = 2$  and 5, that we compare to the numerical solution computed on a fixed grid with  $N_{L_{\max}}$  cells which are respectively 254 and 681.

The adaptive solution for  $L_{\max} = 2$  requires  $N_{L_{\max}} = 254$  cells while for the level  $L_{\max} = 5$  uses  $N_{L_{\max}} = 681$  cells.

As we observe on FIGURE 6-FIGURE 7, the numerical density of entropy production for levels  $L_{\max} = 2$  and 5 captures the contact discontinuity (see FIGURES 6(d)-7(d)) and the shock discontinuity (see FIGURES 6(e)-7(e)). Moreover, the numerical density of entropy production reproduces the behavior of the error made on  $\rho$  (as displayed on FIGURES 6(b)-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 (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 on FIGURE 6-FIGURE 7.

The total production of entropy  $\mathcal{P}$  (11) 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 scheme. We use the discrete  $l_t^1 l_x^1$  norm of the error on the density (see FIGURE 8) and the discrete  $l_x^1$  on 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 on FIGURES 8(a)–8(d).

As expected (see for instance [7, 20]), the convergence of second order methods are of first order on fixed grid due to the presence of discontinuities in the solution (as displayed on FIGURES 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 FIGURES 8(a)–8(d)). Moreover, using local time stepping or uniform time step, the numerical order are similar (see FIGURES 8(a)–8(d) and TABLE 1).

The rate of convergence is considerably increased by the adaptive scheme and, in our experiences, it can be improved by changing the threshold parameters  $\alpha_{\min}$  and  $\alpha_{\max}$  at the expense of the cpu-time.

Let us also emphasize that with the present threshold parameters, the order of the AB1M scheme is approximately 2.

### Accuracy and cpu-time.

Until now, the relevance of the adaptive scheme have been obtained relatively to the number of cells. We now focus on the efficiency of the adaptive scheme with respect to the cpu-time<sup>1</sup>. For the previous numerical tests, we plot the error on the density with respect to the cpu-time (see FIGURES 8(e)-8(f)). As expected, the use of the 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 scheme uses the same number of average cells (254) and computes with the same accuracy ( $l_t^1 l_x^1$  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 approximately 2 times faster than AB1 which proves the efficiency of the local time stepping algorithm. The same statement is also observed with the second order scheme (see TABLE 1).

	$\ \rho - \rho_{ex}\ _{l_t^1 l_x^1}$ 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	0.70	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

Table 1: Numerical order on the density error using  $l_t^1 l_x^1$  norm and  $l_x^1$  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  $\log_{10}$  scale).

<sup>1</sup>Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz

## 5.2 The Lax’s shock tube problem

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

$$x \in [-1, 1], \quad (\rho, u, p)(0, x) = \begin{cases} (0.445, & 0.698, & 3.528), & x \leq 0, \\ (0.5, & 0, & 0.571), & x > 0. \end{cases} \quad (19)$$

This second test is used in order to confirm similar conclusion obtained for the Sod’s shock tube problem. As in the previous case, the exact solution is known and computed from the Riemann problem (14)–(17) with Riemann data (19) using 5 000 uniform cells.

The computational domain here is  $[-1, 1]$  with 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	:	$L_{\max}$ .

As done before, we first represent the numerical solution compared to the exact one for levels  $L_{\max} = 2$  and 5. On 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 first and second order schemes on FIGURES 11(a)–11(d) and compare the accuracy versus cpu-time on FIGURES 11(e)–11(f) and TABLE 2.

Even if the Lax’s shock tube problem introduce a huge shock wave with respect to the Sod’s shock tube problem the same conclusion holds, namely, the numerical density of entropy production reproduces the behavior of the error done as we can observe on 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 FIGURES 11(e)–11(f) and TABLE 2).

	$\ \rho - \rho_{ex}\ _{l_t^1 l_x^1}$ 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

Table 2: Numerical order on the density error using  $l_t^1 l_x^1$  norm and  $l_x^1$  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  $\log_{10}$  scale).

## 5.3 The Shu-Osher test case

The next test case concerns the Shu and Osher’s problem [21] in which a Mach 3 shock moves to the right and collides with an entropy disturbance moving to the left. We focus on the solution at time  $t = 0.18$  s which develops 3 shocks in a highly oscillating regime. Thus, it is hard to compute an accurate numerical solution with a fixed uniform grid which requires a large number of cells to precisely capture oscillating regions. In this section we compare adaptive schemes of order 1 and 2. Especially, we numerically verify that for a given accuracy, the local time stepping algorithm is less time consuming than the standard one.



The Shu and Osher’s initial conditions are

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

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

As a reference solution, we compute the solution on a uniform fixed grid (20 000 cells) with the RK2 scheme. The density of the reference solution at time  $t = 0.18$  s and its numerical density of entropy production are displayed on 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  $0.5 \leq x \leq 0.75$ , one can consider such a solution as an “exact” one.

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	:	$L_{\max} = 4$ .

On FIGURE 12(b), we plot the density of the reference solution, the one by 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 fit to the reference solution. However, focusing closely on the oscillating area between  $0.5 \leq x \leq 0.7$ , one can observe that, as expected, RK2 is better than AB2 which is better than AB1. Those results are also confirmed by the computation of the numerical order displayed on FIGURE 12(d): it was not true in the two previous test cases (see FIGURES 8-11) as emphasized in Section 5.1 (for which the numerical order for AB1, for instance, was greater than 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 final time  $t = 0.18$  s for a level  $L_{\max} = 4$ .

As said before, the hierarchy in terms of accuracy is very well respected as we also see on TABLE 3 for the error on the density or the total production of entropy. We see again a 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 3 times faster than the RK2 which is a significant gain in time.

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

Table 3: Comparison of numerical schemes of order 1 and 2

## 6 Concluding remarks and perspectives

In this paper, first and second order methods in space and time are coupled with an adaptive algorithm employing local time stepping, obtaining an adaptive numerical scheme in which the grid is locally refined or coarsened according to the entropy indicator. Several numerical tests have been performed and 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 same order of accuracy.

We have recalled the Tang-Warnecke projection and the Osher-Sanders projection. In particular, even if the Tang-Warnecke projection is consistent, the solution develops numerical instabilities for 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

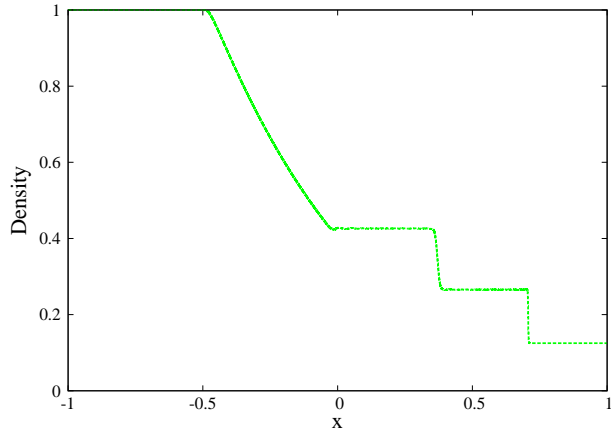
The authors wish to thank the referees for their careful reading of the previous version of the manuscript and useful remarks.

This work is supported by the "Agence Nationale de la Recherche" through the COSINUS program (ANR CARPEiNTER project n° ANR-08-COSI-002).

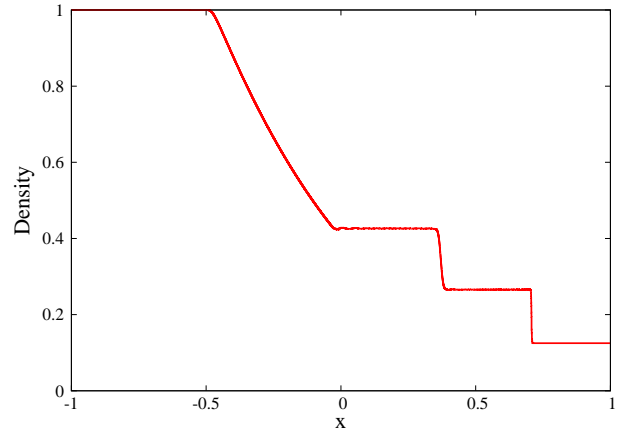
## References

- [1] Allahviranloo T., Ahmady N., Ahmady E., Numerical solution of fuzzy differential equations by predictor–corrector method. *Inform. Sci.*, 177(7):1633–1647, 2007.
- [2] Altmann C., Belat T., Gutnic M., Helluy P., Mathis H., Sonnendrücker E., 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*, volume 28 of *ESAIM Proc.*, pages 33–54. EDP Sci., Les Ulis, 2009.
- [3] Berger M.J., Olinger J., Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.*, 53(3):484–512, 1984.
- [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.*, 66(218):547–572, 1997.
- [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 multi-espèce*. PhD thesis, Université de Paris VI, 1990.
- [6] Eymard R., Gallouët T., Herbin R., Finite volume methods. In *Handbook of numerical analysis, Vol. VII*, Handb. Numer. Anal., VII, pages 713–1020. North-Holland, Amsterdam, 2000.
- [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*, 39(12):1073–1138, 2002.
- [8] Godlewski E., Raviart P.A., *Numerical approximation of hyperbolic systems of conservation laws*, volume 118 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1996.
- [9] Golay F., Numerical entropy production and error indicator for compressible flows. *C.R. Mécanique*, 337:233–237, 2009.

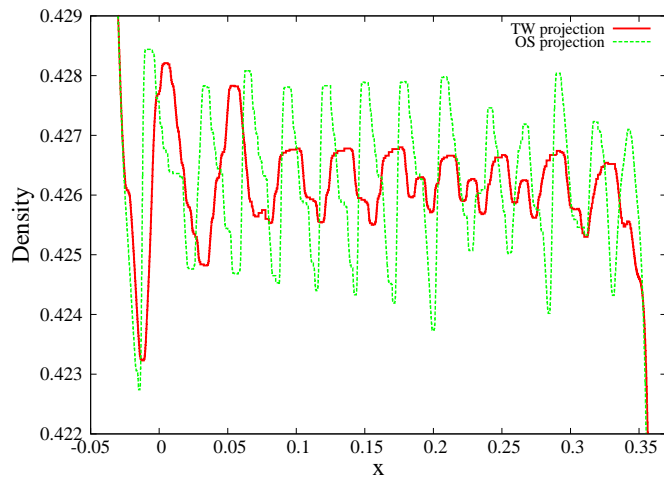
- [10] Guermond J.L., Pasquetti R., Popov B., Entropy viscosity method for nonlinear conservation laws. *J. Comp. Phys.*, 230(11):4248–4267, 2011.
- [11] Hairer E., P. Nørsett S., Wanner G., *Solving ordinary differential equations. I*, volume 8 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1993. Nonstiff problems.
- [12] Houston P., Mackenzie J.A., Süli E., Warnecke G., A posteriori error analysis for numerical approximations of Friedrichs systems. *Numer. Math.*, 82(3):433–470, 1999.
- [13] Jenny P., Lee S.H., Tchelepî H.A., Adaptive multiscale finite-volume method for multiphase flow and transport in porous media. *Multiscale Model. Simul.*, 3(1):50–64, 2005.
- [14] Karni S., Kurganov A., Local error analysis for approximate solutions of hyperbolic conservation laws. *Adv. Comput. Math.*, 22(1):79–99, 2005.
- [15] Karni S., Kurganov A., Petrova G., A smoothness indicator for adaptive algorithms for hyperbolic systems. *J. Comp. Phys.*, 178(2):323–341, 2002.
- [16] Müller S., Stiriba Y., Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping. *SIAM J. Sci. Comput.*, 30(3):493–531, 2007.
- [17] Osher S., Sanders R., Numerical approximations to nonlinear conservation laws with locally varying time and space grids. *Math. Comp.*, 41(164):321–336, 1983.
- [18] Puppo G., Numerical entropy production on shocks and smooth transitions. In *Proceedings of the Fifth International Conference on Spectral and High Order Methods (ICOSAHOM-01) (Uppsala)*, volume 17(1-4), pages 263–271, 2002.
- [19] Puppo G., Numerical entropy production for central schemes. *SIAM J. Sci. Comput.*, 25(4):1382–1415 (electronic), 2003/04.
- [20] Puppo G., Semplice M., Numerical entropy and adaptivity for finite volume schemes. *Commun. Comput. Phys.*, 10(5):1132–1160, 2011.
- [21] Shu C. W., Osher S., Efficient implementation of essentially nonoscillatory shock-capturing schemes. *J. Comput. Phys.*, 77(2):439–471, 1988.
- [22] Simeoni C., Remarks on the consistency of upwind source at interface schemes on nonuniform grids. *SIAM J. Sci. Comput.*, 48(1):333–338, 2011.
- [23] Sod G.A., A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.*, 27(1):1–31, 1978.
- [24] Sonar T., Hannemann V., Hempel D. Dynamic adaptivity and residual control in unsteady compressible flow computation. *Math. Comput. Modelling*, 20(10-11):201–213, 1994.
- [25] Tan Z., Zhang Z., Huang Y., Tang T., Moving mesh methods with locally varying time steps. *J. Comp. Phys.*, 200(1):347–367, 2004.
- [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.*, 26(4):1415–1431, 2005.
- [27] Toro E.F., *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, Berlin, second edition, 1999. A practical introduction.
- [28] Zhang X.D., Trépanier J.Y., Camarero R., A posteriori error estimation for finite-volume solutions of hyperbolic conservation laws. *Comp. Meth. App. Eng.*, 185:1–19, 2000.



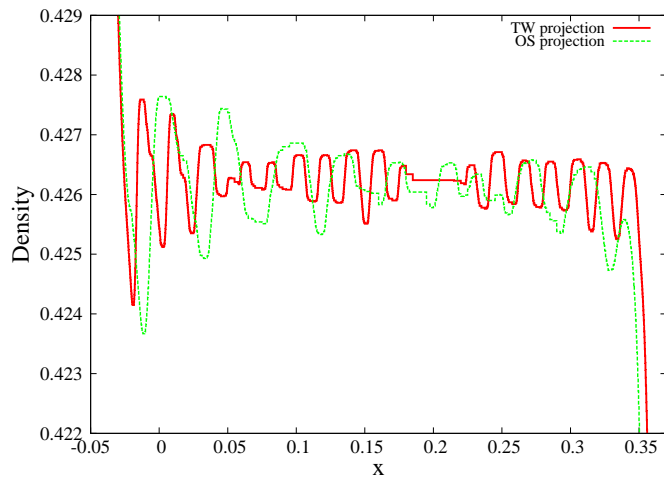
(a) Numerical solution using the OS projection.



(b) Numerical solution using the TW projection.

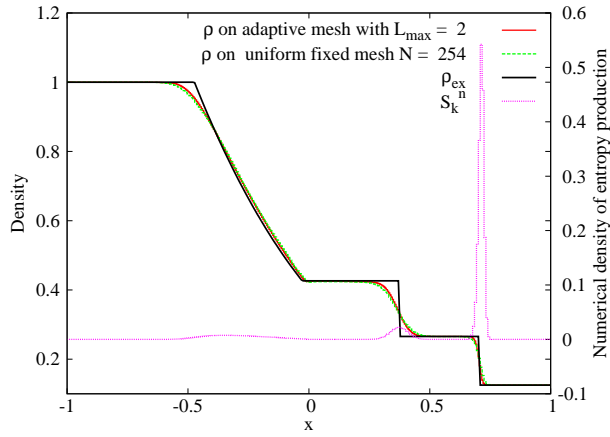


(c) Without smoothing effect.

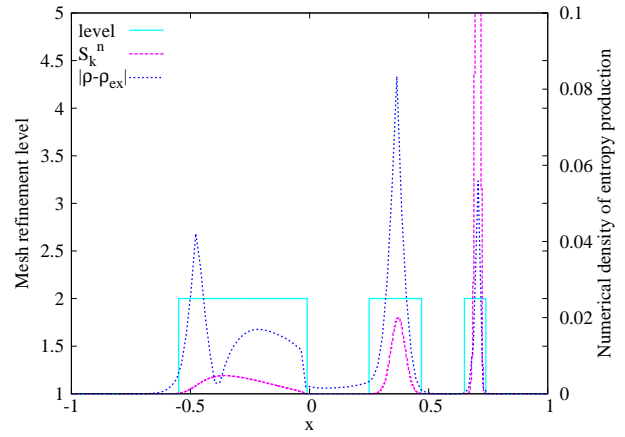


(d) With smoothing effect.

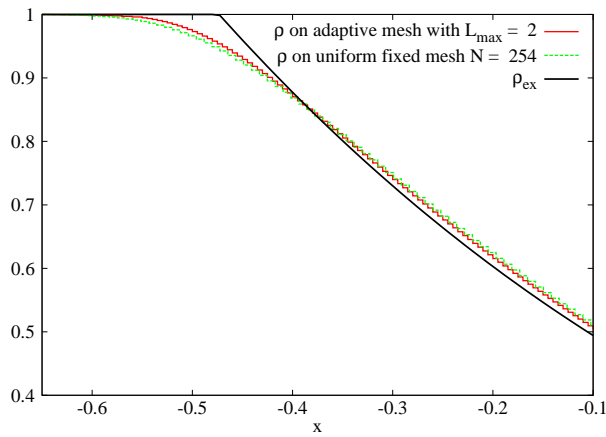
Figure 5: Sod's shock tube problem: numerical oscillations with/without smoothing effect at time  $t = 0.4$  s with  $L_{\max} = 6$ . The Tang-Warnecke (TW) and the Osher-Sanders (OS) projections are used.



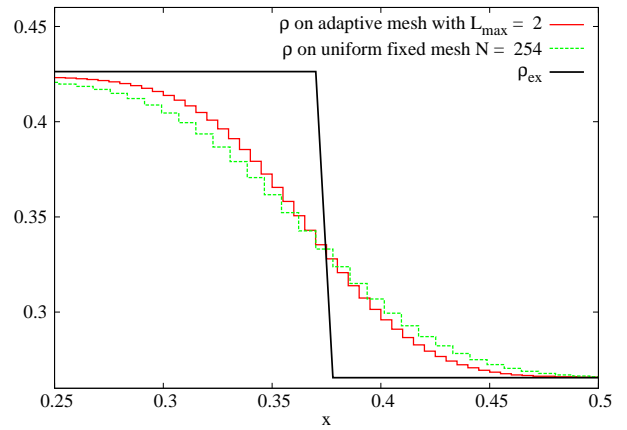
(a) Density and numerical density of entropy production.



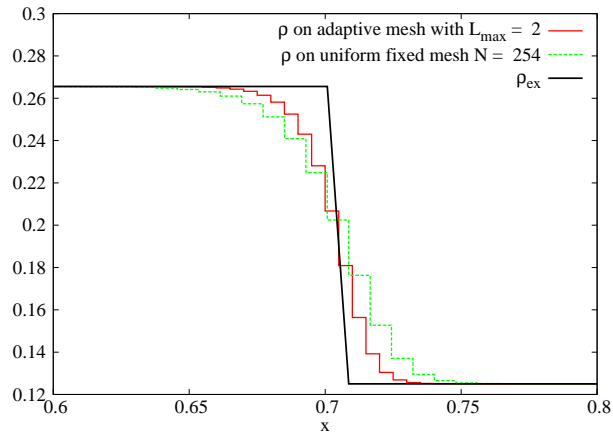
(b) Mesh refinement level, numerical density of entropy production and local error.



(c) Zoom on the rarefaction.

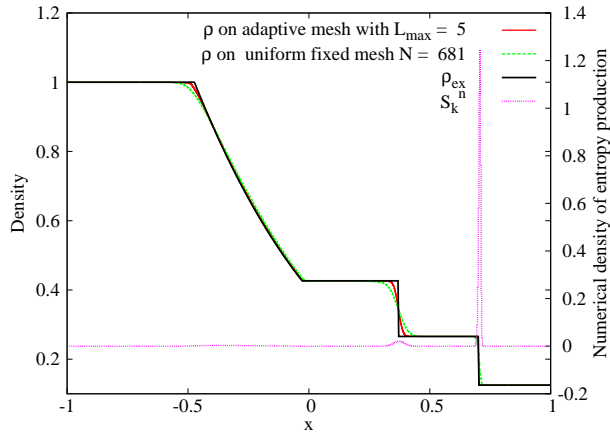


(d) Zoom on the contact discontinuity.

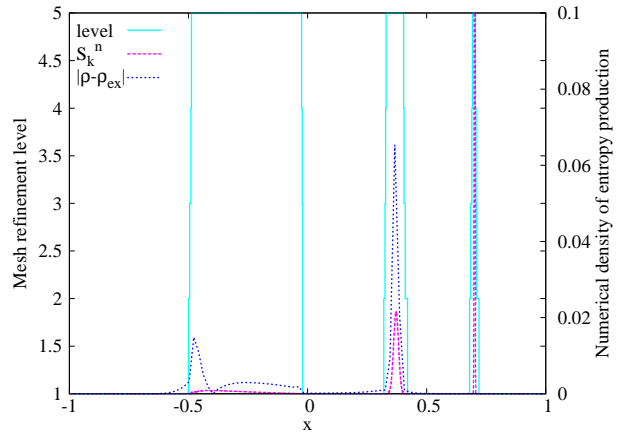


(e) Zoom on the shock.

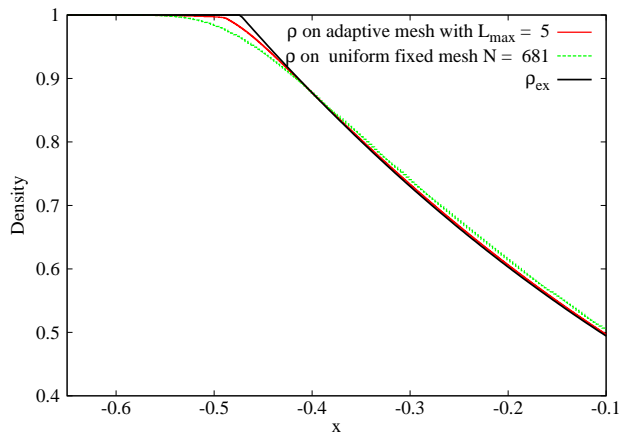
Figure 6: Sod's shock tube problem: solution at time  $t = 0.4$  s using the AB1M scheme on a dynamic grid with  $L_{\max} = 2$  and the AB1 scheme on a uniform fixed grid of 254 cells.



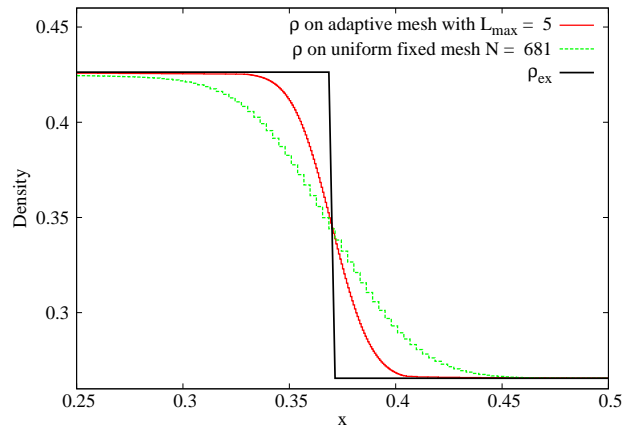
(a) Density and numerical density of entropy production.



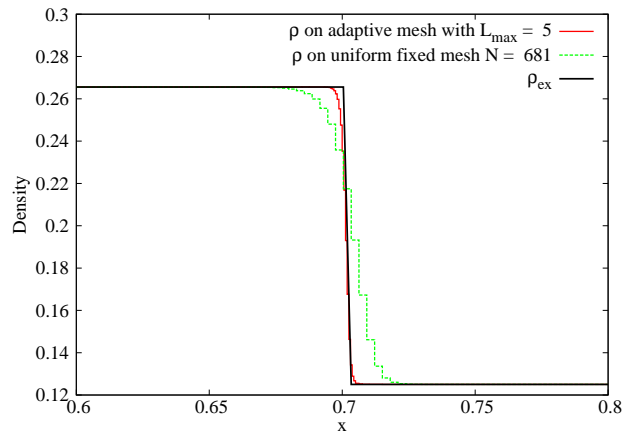
(b) Mesh refinement level, numerical density of entropy production and local error.



(c) Zoom on the rarefaction.

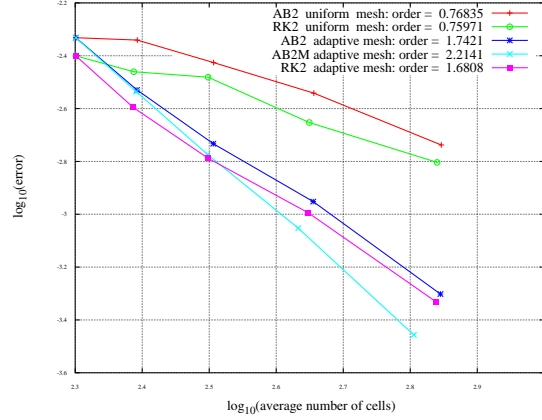
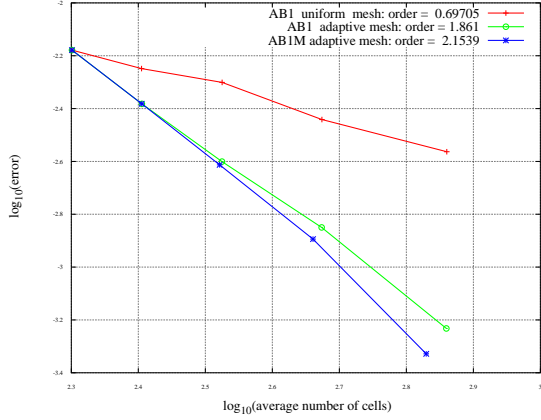


(d) Zoom on the contact discontinuity.

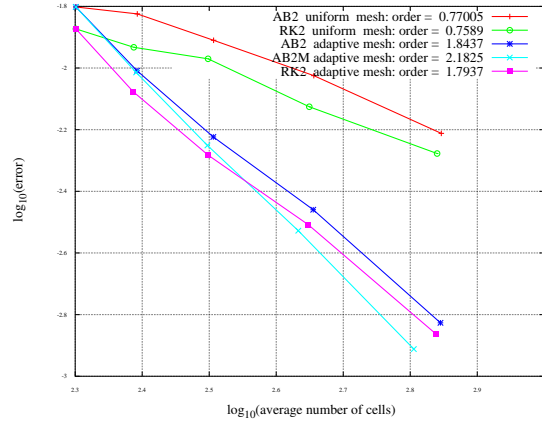
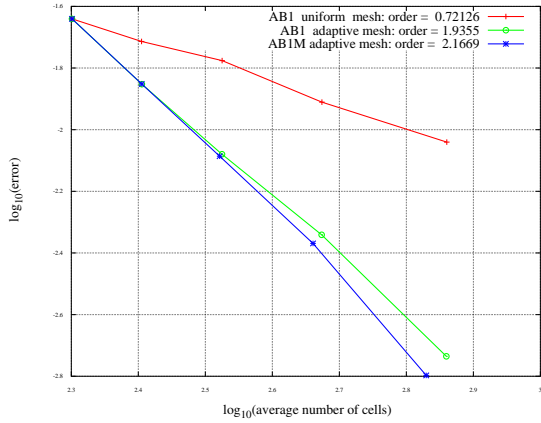


(e) Zoom on the shock.

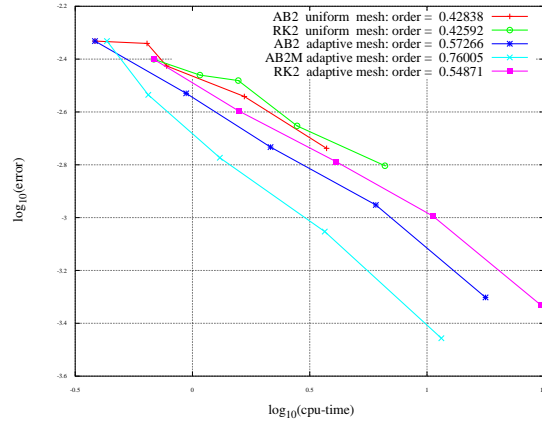
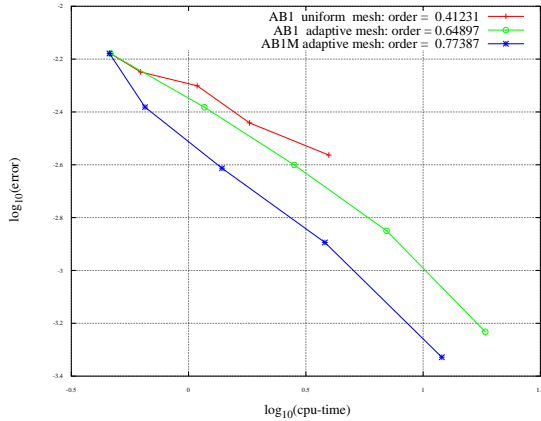
Figure 7: Sod's shock tube problem: solution at time  $t = 0.4$  s using the AB1M scheme on a dynamic grid with  $L_{\max} = 5$  and the AB1 scheme on a uniform fixed grid of 681 cells.



(a)  $\|\rho_{ex} - \rho\|_{l_t^1 l_x^1}$  with respect to the averaged number of cells for the schemes of order 1. (b)  $\|\rho_{ex} - \rho\|_{l_t^1 l_x^1}$  with respect to the averaged number of cells for the schemes of order 2.

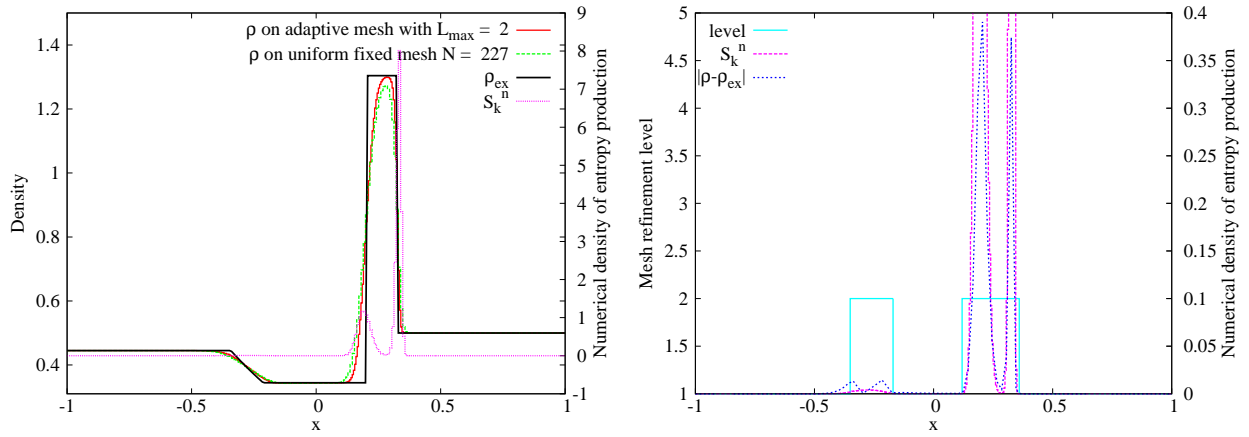


(c)  $\|\rho_{ex} - \rho\|_{l_x^1}$  with respect to the averaged number of cells for the schemes of order 1 at time  $t = 0.4$  s. (d)  $\|\rho_{ex} - \rho\|_{l_x^1}$  with respect to the averaged number of cells for the schemes of order 2 at time  $t = 0.4$  s.



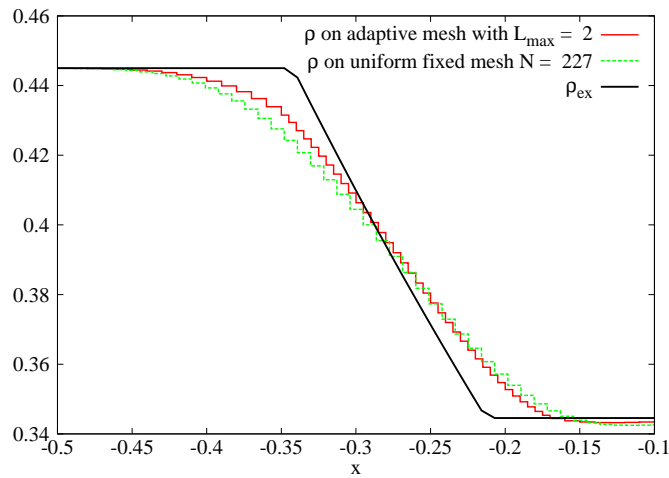
(e)  $\text{cpu-time}-\|\rho_{ex} - \rho\|_{l_t^1 l_x^1}$  with respect to the cpu-time for the schemes of order 1. (f)  $\text{cpu-time}-\|\rho_{ex} - \rho\|_{l_t^1 l_x^1}$  with respect to the cpu-time for the schemes of order 2.

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

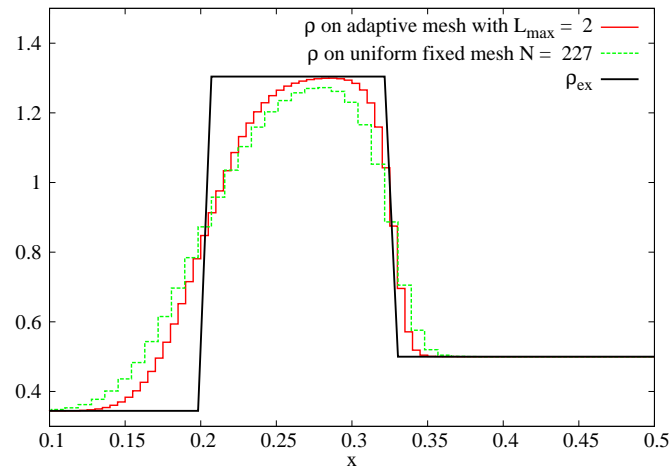


(a) Density and numerical density of entropy production.

(b) Mesh refinement level and local error.



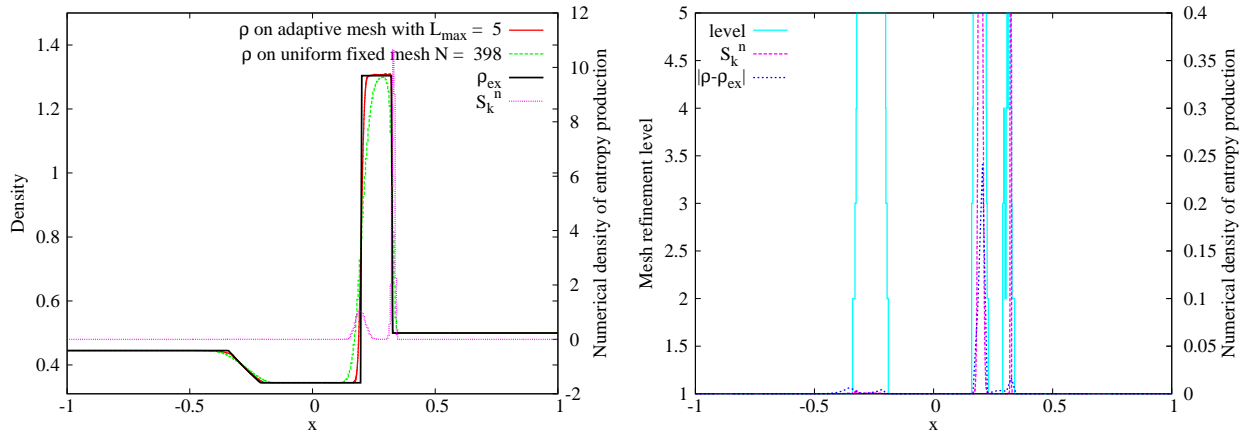
(c) Zoom on the rarefaction.



(d) Zoom on the contact discontinuity.

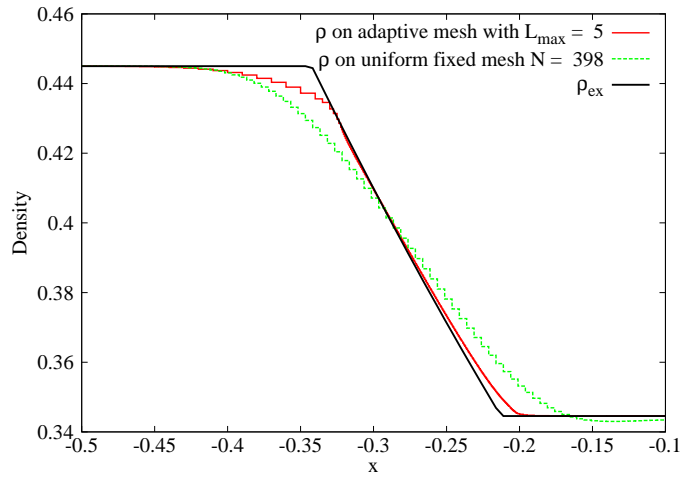
Figure 9: Lax's shock tube problem: solution at time  $t = 0.13$  s using AB1M scheme with  $L_{\max} = 2$ .



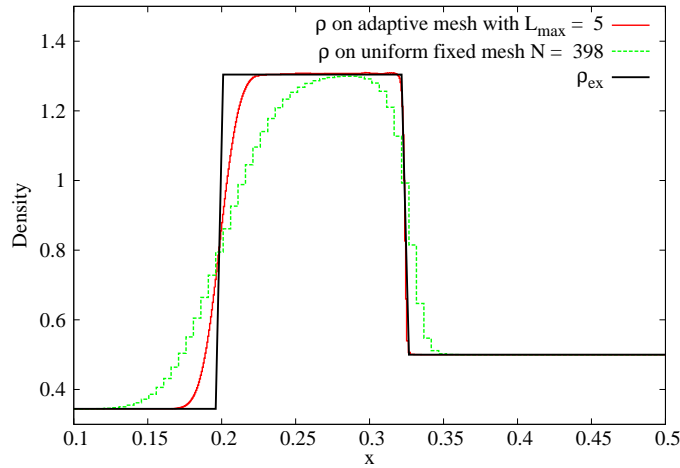


(a) Density and numerical density of entropy production.

(b) Mesh refinement level and local error.

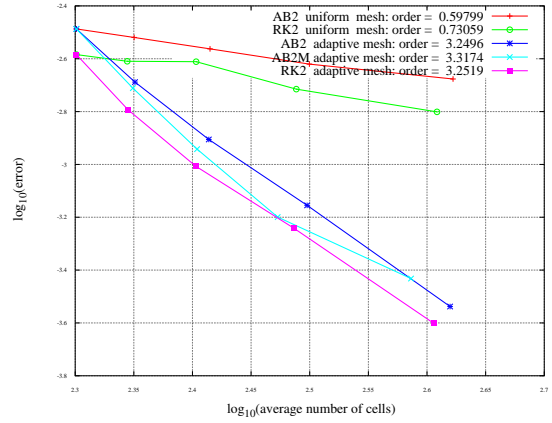
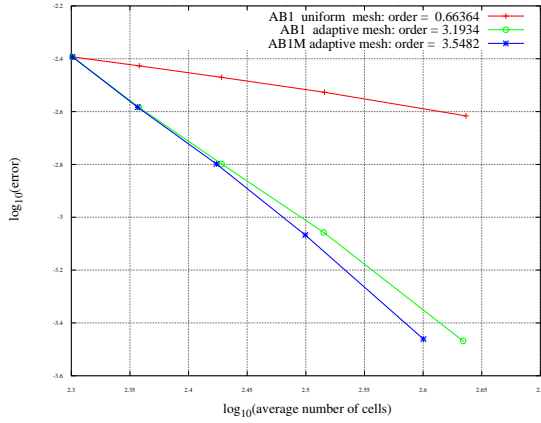


(c) Zoom on the leg of the rarefaction.

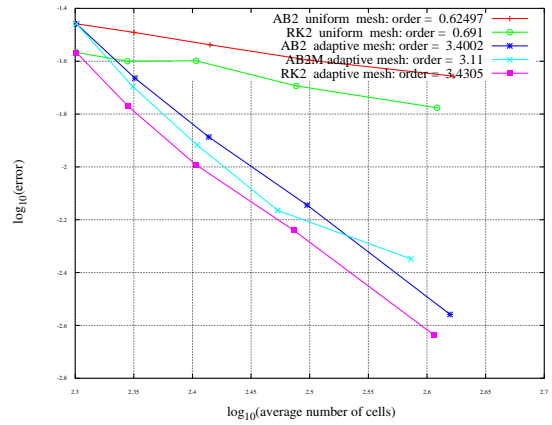
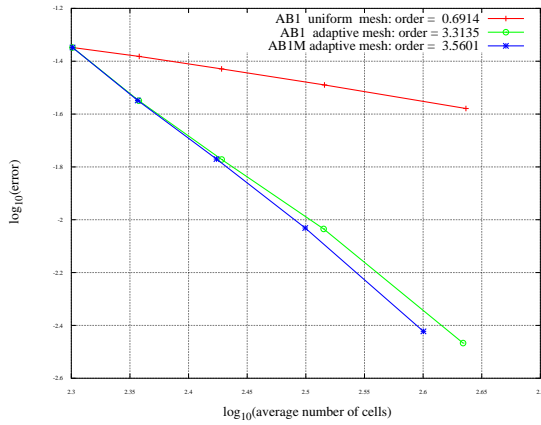


(d) Zoom on the contact discontinuity.

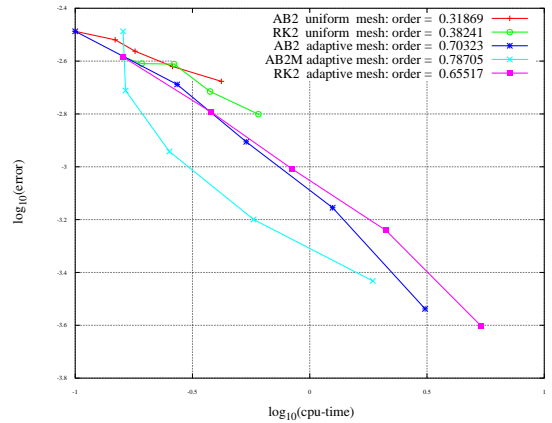
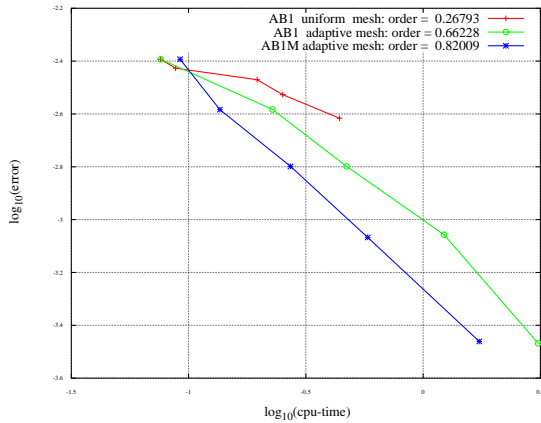
Figure 10: Lax's shock tube problem: solution at time  $t = 0.13$  s using AB1M scheme with  $L_{\max} = 5$ .



(a)  $\|\rho_{ex} - \rho\|_{l_t^1 l_x^1}$  with respect to the averaged number of cells for the schemes of order 1. (b)  $\|\rho_{ex} - \rho\|_{l_t^1 l_x^1}$  with respect to the averaged number of cells for the schemes of order 2.

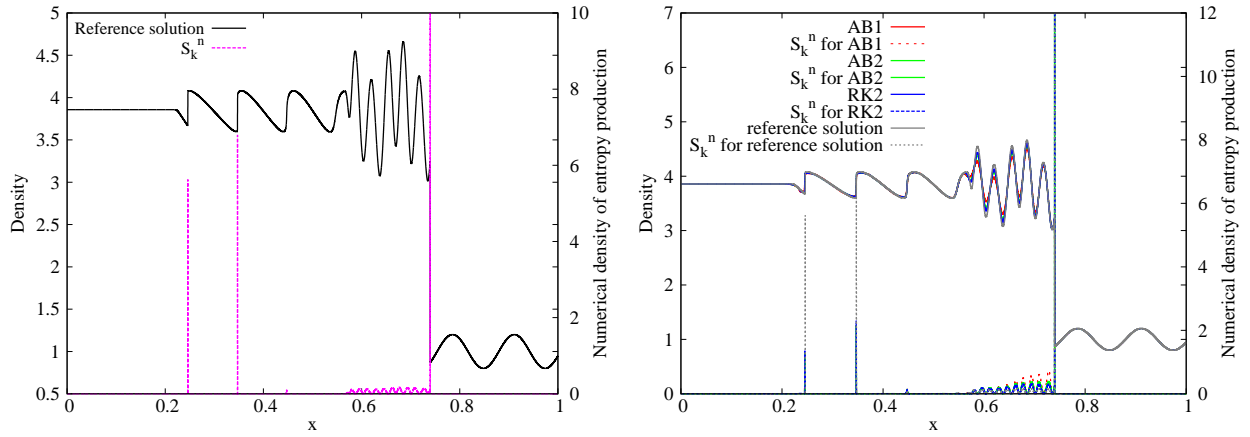


(c)  $\|\rho_{ex} - \rho\|_{l_x^1}$  with respect to the averaged number of cells for the schemes of order 1 at time  $t = 0.13$  s. (d)  $\|\rho_{ex} - \rho\|_{l_x^1}$  with respect to the averaged number of cells for the schemes of order 2 at time  $t = 0.13$  s.



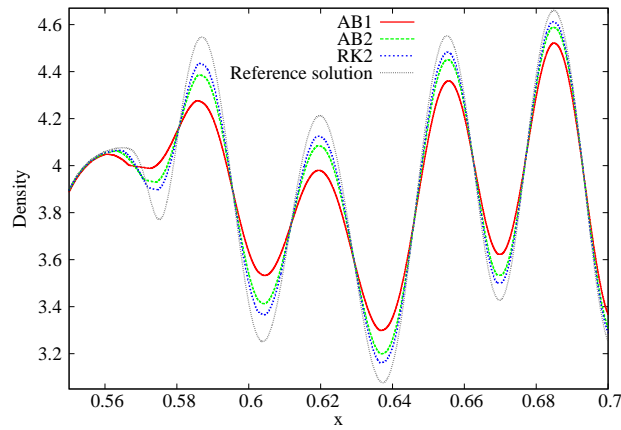
(e)  $\text{cpu-time} - \|\rho_{ex} - \rho\|_{l_t^1 l_x^1}$  with respect to the cpu-time for the schemes of order 1. (f)  $\text{cpu-time} - \|\rho_{ex} - \rho\|_{l_t^1 l_x^1}$  with respect to the cpu-time for the schemes of order 2.

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

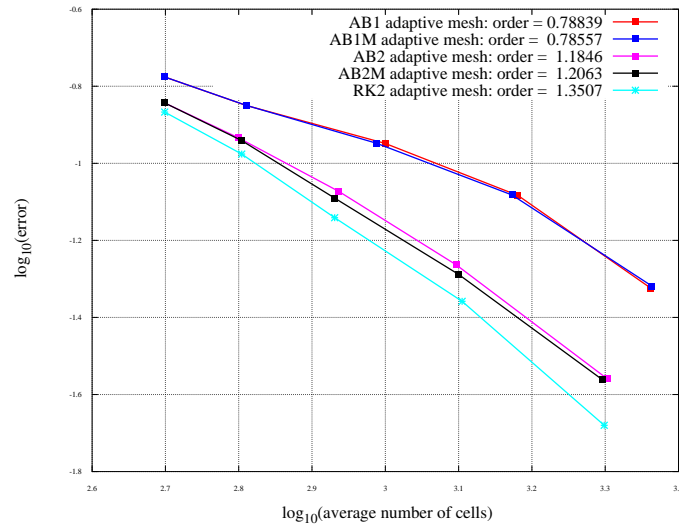


(a) Reference solution.

(b) Density and numerical density of entropy production.



(c) Zoom on oscillating region.



(d) Numerical order:  $\|\rho_{ex} - \rho\|_{L^1_x}$  with respect to the averaged number of cells for the schemes of order 1 and 2 at time  $t = 0.18$  s.

Figure 12: Shu Osher test case.