ESAIM: Proceedings CEMRACS 1999
URL: http://www.emath.fr/Maths/Proc/Vol.10

MULTIRESOLUTION FINITE VOLUME SCHEME FOR A MIXED HYPERBOLIC-ELLIPTIC SYSTEM. *

Oliver Grosshans[1] and Marie Postel[2]

Abstract. A mixed hyperbolic-elliptic system arises in the context of liquid-vapor phase transition modelling. It has been studied numerically with an explicit finite volume method based on a relaxation scheme which must resolve very precisely the very steep phase transition region. This requirement leads to prohibitive cost when a uniform discretization is used. Multiresolution is a good candidate to improve the performances of such schemes and we examine the possibility of implementing the algorithms developed originally for strictly hyperbolic systems.

AMS Subject Classification. 65-10, 76T05, 76M12 .

.

1. Introduction

This work is part of a general project of modeling the thermovector fluid in the reactor of a nuclear plant. In the first phase of the research, a set of equations was established, taking into account the thermodynamical effects near the critical point, the capillarity phenomena and the fluid reaction on the wall of the reactor. Some preliminary numerical studies were performed whose conclusions were that these so called "second gradient" equations are very hard to solve [8].

The first difficulty is the modeling of the phase transition by a continuous variation of the density - whose shape is important for the engineer although it occurs on a very fine scale. The presence of second and third derivatives in the density requires a special numerical treatment, as well as the instability arising from the particular thermodynamical closure law. This theme of research was first proposed by CEA Grenoble at Cemracs'98.

Our interest in this problem was really aroused by the first difficulty which can be viewed as the challenge of describing very accurately a phenomenon arising in a very limited time dependent area of a relatively large domain. Setting aside the other inherent technical difficulties, this is the typical setting for applying a multiscale technique such as the one initially introduced by Harten [7] for hyperbolic equations and lately developed by Cohen [2],[3], Dahmen [5], Donat [1].

Keywords and phrases: Multiresolution, finite volumes

* *The authors thank Albert Cohen and Frederic Coquel (LAN) for their help on this project during the Cemracs99, and Olivier Lebaigue (CEA, Grenoble) for his sponsorship*

[1] SFB 359, IWR, INF 368, Universität Heidelberg, D-69120 Heidelberg, Germany
e-mail: O.Grosshans@iwr.uni-heidelberg.de

[2] Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie, Boite courrier 187, 75252 Paris Cedex 05, France
e-mail: postel@ann.jussieu.fr

The techniques rely on the basic observation that systems of conservation laws generally require very complicated and costly schemes, because they are nonlinear and may therefore develop singularities even starting from smooth initial data. On the other hand, these costly schemes, like for instance ENO type flux reconstructions, are really necessary only in the vicinity of the discontinuities and could be advantageously replaced by simpler high order centered schemes in the smooth regions. This requires an efficient way to detect the singularities which brings immediately in mind wavelet decomposition techniques. The original multiresolution algorithm proposed by Harten relies on wavelet theory but is really adapted to the particular case of solutions of finite volumes schemes, that is functions known by their mean values on a grid. In broad lines, it consists in building a hierarchy of nested grids. The solution given on a uniform grid can be described by its mean values on the immediately coarser grid, plus the differences between the fine representation and its approximation by interpolation from the coarse one. Repeating along the succession of levels provides the encoded representation of the solution. The size of the differences provides smoothness indicators and the underlying interpolation operator can be used — wherever these differences are negligible — to compute the evolution of the solution on the coarse grid only.

In this paper, we describe more precisely the multiresolution method adapted to the specific finite volume scheme designed to solve the second gradient equations. Since the hyperbolicity assumption used in the original design of the method is not valid here, numerical experiments have been performed to study the stability and robustness of the method. So far the original Harten scheme has been implemented and the solution is computed on the finest level everywhere at each time step. This prototype is meant to be developed into a more complicated algorithm were the solution will be computed on a completely adaptive grid. For sake of simplicity we have only consider the 1D, isothermal case. Simultaneously, in this second stage taking place during Cemracs'99, several projects aimed towards extending the first years results [4] to the case of the non-isothermal form of the equations, 1D, 2D, and implicit schemes, and are being discussed in [6].

2. The second gradient equations

The initial system of equations

$$\begin{aligned} \partial_t \rho + \partial_x(\rho u) &= 0 \\ \partial_t(\rho u) + \partial_x(\rho u^2 + P(\rho)) &= \lambda \rho \partial_{xxx} \rho \end{aligned} \tag{1}$$

where P is for instance the Van der Waals pressure law, has been studied in D. Jamet's thesis [8]. During Cemracs'98 F. Coquel and O. Großhans transformed this system by introducing an auxiliary unknown N into

$$\begin{aligned} \partial_t \rho + \partial_x(\rho u) &= 0 \\ \partial_t(\rho N) + \partial_x(\rho N u) &= \sqrt{\lambda}\partial_{xx} u \\ \partial_t(\rho u) + \partial_x(\rho u^2 + P(\rho) + \frac{3}{2}N^2\rho^4) &= -\sqrt{\lambda}\partial_{xx}(\rho^3 N) \end{aligned}$$

and studied in detail its numerical behaviour. For simplicity of presentation let us consider the following equations

$$\begin{aligned} \partial_t \rho + \partial_x(\rho u) &= 0 \\ \partial_t(\rho u) + \partial_x(\rho u^2 + p(\rho)) &= 0 \end{aligned} \tag{2}$$

with $p(\rho) = P(\rho) + \frac{\lambda}{2}(\partial_x \rho)^2 - \lambda\rho(\partial_{xx}\rho)$ which are equivalent to (1). Now all the difficulties are hidden in this extremely complex pressure $p(\rho)$. For more details please refer to [4, 6]

3. Numerical scheme — a relaxation method

To solve the system (2) we apply a relaxation method and relax on the pressure $p(\rho)$. This leads to

$$\begin{aligned} \partial_t \rho + \partial_x(\rho u) &= 0 \\ \partial_t(\rho u) + \partial_x(\rho u^2 + \Pi) &= 0 \\ \partial_t(\rho \Pi) + \partial_x(\rho u \Pi + a^2 u) &= \frac{\rho}{\varepsilon}\left(P(\rho, N) - \Pi\right) \end{aligned} \tag{3}$$

where $a = \rho^2 \frac{\partial p}{\partial \rho}$. For $\varepsilon \to 0$ we get the original system (2) back and $\varepsilon \to \infty$ gives us the relaxed system. The algorithm is now as follows: First we solve the $\varepsilon \to \infty$ system, that is we obtain values for $\rho, \rho u, \rho\Pi$ at $t_{n+1,-}$ and then we let $\varepsilon \to 0$ which essentially means computing Π^{n+1} as $\rho(t_{n+1})$ because $\rho, \rho u$ at t_{n+1} are equal to $\rho, \rho u$ at $t_{n+1,-}$

We use an explicit finite volume scheme to solve the equations (3) in the case $\varepsilon \to \infty$. Since Π can be viewed as a dependent variable in this algorithm, we denote by U the vector of conservative quantities $(\rho, \rho u)^T$ and by F the flux function

$$F(x,t) = \left(\rho u, \rho u^2 + \Pi\right)^T$$

The computational domain is subdivided into N intervals $I_k = [x_{k-1/2}, x_{k+1/2}]$ of length h. The numerical solution is an approximation of the mean values $\bar{U}_k^n \approx U_k^n = \mathcal{A}(I_k)U(.,t_n)$, where $\mathcal{A}(I)u$ denotes the *mean* operator on an interval I_k

$$\mathcal{A}(I)u = \frac{1}{|I|}\int_I u(x)dx.$$

The numerical solution is computed in its conservative form as

$$\bar{U}_k^{n+1} = \bar{U}_k^n - \frac{\Delta t}{h}\left(\hat{f}_{k+1/2}^n - \hat{f}_{k-1/2}^n\right)$$

where $\hat{f}_{k-/2}^n$ is the numerical flux at point $x_{k-1/2}$, *i.e.* an approximation of the flux across the intervals boundary

$$\hat{f}_i^n \approx \bar{f}_i^n = \frac{1}{\Delta t}\int_{n\Delta t}^{(n+1)\Delta t} F(i\Delta x, t)dt.$$

There are two difficulties in the algorithm just above. The flux evaluation is one of them: the evolution of the two independent quantities ρ and u using system (3) requires to solve at each time step the local Riemann problem for the homogeneous system obtained when letting ϵ go to ∞ in (3). Furthermore, computing the initial condition for the next time step amounts to relax the system (3) with $\epsilon \to 0$. In other words, knowing $\bar{u}^{n+1}$ and $\bar{\rho}^{n+1}$, compute Π^{n+1} we need the third derivative of ρ^{n+1}. Therefore the evaluation of Π at one point will require the values of $\bar{\rho}$ and $\bar{u}$ at four neighboring points. We refer to [6] for the details of derivation and summarize the computation of the numerical fluxes by

$$\hat{f}_{i+1/2}^n = \hat{F}\left(\bar{U}_{i-1}^n, \bar{U}_i^n, \bar{U}_{i+1}^n, \bar{U}_{i+2}^n\right) \tag{4}$$

It is therefore crucial to limit the number of those flux evaluations and the multiresolution can play an important role here. The domain is subdivided in several levels of resolution nested within each other. Each interval is the union of two intervals from the finer level. The method consists in solving the problem everywhere on the finest grid while taking advantage of smoothness wherever possible to reduce flux evaluations (4). At each time step, details provided by the multiresolution analysis of the solution are used to choose the flux computation

method. In the areas where the solution is smooth enough fluxes at the edges of the interval are interpolated from the fluxes at the edges of intervals on the coarser level. Correspondingly, in areas of strong variations, fluxes are computed precisely using (4) with the finite volume solution on the finest grid as input values.

In the following section we recall Harten's algorithms as developed in [7] and references within. We first describe the encoding / decoding algorithms for a function represented by its mean values or by point values. Then we show how they are used in the multiresolution coupled with finite volumes scheme.

4. Finite volume scheme and multiresolution. Harten 's algorithm

In this section we introduce notations and recall the original form of Harten's algorithm found in [7]. We refer the reader to this article for details and justifications The computational domain Ω is divided at each level ℓ of resolution in N_ℓ intervals $I_k^\ell = [x_{k-1/2}^\ell, x_{k+1/2}^\ell]$ of length h_ℓ. The levels are nested within each other starting from the coarse grid Ω^0 until the finest grid Ω^L. Intervals on one level ℓ all have same length h_ℓ and are divided by two to form the immediately finer level $\ell+1$: $I_k^\ell = I_{2k}^{\ell+1} \cup I_{2k+1}^{\ell+1}$. We therefore have $x_{2k-1/2}^{\ell+1} = x_{k-1/2}^\ell$ and $x_{2k+1/2}^{\ell+1} = (x_{k-1/2}^\ell + x_{k+1/2}^\ell)/2$. We note $\bar{U}_k^\ell = \mathcal{A}(I_k^\ell)U$ the mean values of the vector function U on each interval k of a level ℓ. Harten describes two methods to encode a function with a multiresolution analysis depending on whether the function is known through its mean values on the intervals (algorithm 1) or through its point values at the intervals bounds. In fact the finite volume scheme will use both representations: the mean value one for the solution $\bar{U}_k^{\ell,n}$ itself, and the point value one for the fluxes $\hat{f}_{k+1/2}^{\ell,n}$, since these last quantities may actually be considered as primitives of $F_x(U)$ evaluated at the discretization points $x_{k+1/2}^\ell$.

Algorithm 1. *Encoding (mean values)*

U is known through its approximated mean values $\bar{U}^L$ on finest grid Ω^L
for $\ell = L-1 \searrow 0$

1. Compute $\bar{U}^\ell$: $\bar{U}_k^\ell = (\bar{U}_{2k}^{\ell+1} + \bar{U}_{2k+1}^{\ell+1})/2$ for $k = 1, \dots, N_\ell$
2. Compute $\tilde{U}^{\ell+1}$ on $\Omega^{\ell+1}$: $\tilde{U}_{2k}^{\ell+1} = \mathcal{I}(I_{2k}^{\ell+1}; \bar{U}^\ell)$ for $k = 1, \dots, N_\ell$
3. Compute *details* between $\Omega^{\ell+1}$ and Ω^ℓ: $d_k^\ell = \bar{U}_{2k}^{\ell+1} - \tilde{U}_{2k}^{\ell+1}$ for $k = 1, \dots, N_\ell$

End of **for** ℓ

We note $\bar{U}_{MR} = (\bar{U}^0, d^0, \dots, d^{L-1})$ the *multiresolution* representation of function $\bar{U}$, namely its mean values on the coarse grid plus the details allowing to reconstruct it on the finest grid using the following algorithm:

Algorithm 2. *Decoding (mean values)*

U is known by its mean values $\bar{U}^0$ on the coarsest grid Ω^0 and all the details d^ℓ for $\ell = 0, \dots, L-1$
for $\ell = 0 \nearrow L-1$

1. *Interpolation*: compute $\tilde{U}^{\ell+1}$: $\tilde{U}_{2k}^{\ell+1} = \mathcal{I}(I_{2k}^{\ell+1}; \bar{U}^\ell)$ for $k = 1, \dots, N_\ell$
2. *Reconstruction*: compute $\bar{U}^{\ell+1}$ on $\Omega^{\ell+1}$: $\bar{U}_{2k}^{\ell+1} = d_k^\ell + \tilde{U}_{2k}^{\ell+1}$ and $\bar{U}_{2k+1}^{\ell+1} = 2\bar{U}_k^\ell - \bar{U}_{2k}^{\ell+1}$

End of **for** ℓ

In the two previous algorithms, the linear interpolation operator $\mathcal{I}$ is obtained by imposing polynomial exactness of degree s. In the numerical simulations, we will set $s = 1$. which means that to go from a level l to the level $l+1$ we use a three cell averages interpolation, and as explicit formulas we have

$$
\begin{aligned}
\tilde{U}_{2k}^{\ell+1} &= \frac{1}{8}(\bar{U}_{k-1}^{\ell} + 8\bar{U}_{k}^{\ell} - \bar{U}_{k+1}^{\ell}) \quad \text{inside the domain} \\
\tilde{U}_{0}^{\ell+1} &= \frac{1}{8}(11\bar{U}_{0}^{\ell} - 4\bar{U}_{1}^{\ell} + \bar{U}_{2}^{\ell}) \quad \text{on the left boundary} \\
\tilde{U}_{n_{\ell+1}}^{\ell+1} &= \frac{1}{8}(-\bar{U}_{n_\ell-2}^{\ell} + 4\bar{U}_{n_\ell-1}^{\ell} + 5\bar{U}_{n_\ell}^{\ell}) \quad \text{on the right boundary}
\end{aligned}
$$

We denote by M the encoding operator such that $U_M = (\bar{U}^0, d^0, d^1, ..d^{L-1}) = M\bar{U}^L$. If $\Lambda = \{(k,\ell)\}$ is a set of double indices such that $\cup_{\lambda\in\Lambda} I_\lambda \subset \cup_{\ell=0}^{L}\Omega^\ell$, we denote by T_Λ the thresholding operator on an encoded function such that $T_\Lambda(\bar{U}^0, d^0, d^1, ..d^{L-1}) = (\bar{U}^0, \tilde{d}^0, \tilde{d}^1, ..\tilde{d}^{L-1})$ with $\tilde{d}_\lambda = 0$ if $\lambda \notin \Lambda$ and we introduce a level dependent thresholding based on a tolerance ε which defines a set of indices $\Lambda_\varepsilon = \{\lambda = (k,\ell), \|d_k^\ell\| > 2^\ell\varepsilon\}$. It is easy to recover the basic result $\|T_\varepsilon\bar{U} - \bar{U}\| < C\varepsilon$ where $T_\varepsilon = M^{-1}T_{\Lambda_\varepsilon}M$.

Similar algorithms exist for a function f known this time through its values $f_{k-1/2}^\ell$ at points $x_{k-1/2}^\ell$. The finite volume mean values being primitives of the flux function at interval end points, the corresponding order of interpolation for the fluxes on a level $\ell+1$ is a third degree polynomial depending on the values of the fluxes at four points on the level ℓ as shown below

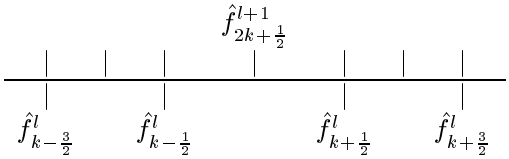

TABLE 1. flux interpolation

The explicit formulas are given as

$$
\begin{aligned}
\hat{f}_{2k+\frac{1}{2}}^{\ell+1} &= \frac{1}{16}(-\hat{f}_{k-\frac{3}{2}}^{\ell} + 9\hat{f}_{k-\frac{1}{2}}^{\ell} + 9\hat{f}_{k+\frac{1}{2}}^{\ell} - \hat{f}_{k+\frac{3}{2}}^{\ell}), \quad \text{inside the domain} \\
\hat{f}_{\frac{1}{2}}^{\ell+1} &= \frac{1}{16}(5\hat{f}_{-\frac{1}{2}}^{\ell} + 15\hat{f}_{\frac{1}{2}}^{\ell} - 5\hat{f}_{\frac{3}{2}}^{\ell} + \hat{f}_{\frac{5}{2}}^{\ell}), \quad \text{on the left boundary} \\
\hat{f}_{2n_\ell+\frac{1}{2}}^{\ell+1} &= \frac{1}{16}(\hat{f}_{n_\ell-\frac{5}{2}}^{\ell} - 5\hat{f}_{n_\ell-\frac{3}{2}}^{\ell} + 15\hat{f}_{n_\ell-\frac{1}{2}}^{\ell} + 5\hat{f}_{n_\ell+\frac{1}{2}}^{\ell}), \quad \text{on the right boundary.}
\end{aligned} \tag{5}
$$

Similarly to the thresholding described above for the mean value encoding, we can define a thresholding for a point value function which will amount to use the interpolated values (5) whenever they are close enough to the true values. As it is explained and justified in details in [2], the multiresolution scheme relies on the construction of a set $\tilde{\Lambda_\varepsilon}^{n+1}$ which contains both Λ_ε^n and $\Lambda_\varepsilon^{n+1}$. The solution at time t^{n+1} on $\tilde{\Lambda}_\varepsilon^{n+1}$ is therefore close enough to the thresholded solution on $\Lambda_\varepsilon^{n+1}$. The construction of this set $\tilde{\Lambda}_\varepsilon^{n+1}$ will be further detailed and justified. Since the thresholded solutions at both time t^n and t^{n+1} are within the prescribed tolerance ε it is straightforward to obtain that the point values fluxes can be thresholded on the same set of indices. In the case where the initial finite volume scheme is L^1 contractive, it is shown that the error between the multiresolution solution and the standard finite volume one is in $C(t^n)\varepsilon$. A natural choice for the parameter ε is a value which makes this last estimate of the same order as the intrinsic error estimate of the finite volume scheme (typically in $2^{-L/2}$).

The following algorithm summarizes the full multiresolution scheme

Algorithm 3. *Finite Volumes + Multiresolution*

Initialization $\bar{u}_k^{L,0} = \mathcal{A}(I_k^L)u_0 \quad \forall I_k^L \in \Omega^L$
Coding $\bar{u}^{L,0} \to \bar{u}_{MR}^0 = (\bar{u}^{0,0}, d^0, \dots, d^{L-1})$. (see algorithm 1)
Loop on time step n

- Determination of $\tilde{\Lambda}_\varepsilon^{n+1}$ and thresholding $\mathbf{u}_{MR}^n \to \mathbf{u}_{MR\varepsilon}^n$ (see algorithm 4)
- Decoding $\mathbf{u}_{MR}^n \to \mathbf{u}^{L,n}$. (see algorithm 2)
- Flux computation for $I_k^0 \in \Omega^0$ on coarse grid, using fine data
  $\hat{f}_{k-1/2}^0 = \hat{F}(\bar{u}_{2^L k-1}^L, \bar{u}_{2^L k}^L)$ (4)
- Compute solution at time $n+1$ on coarse grid Ω^0
  $u_k^{n+1,0} = \bar{u}_k^{n,0} - \frac{\Delta t}{h_0}\left(\hat{f}_{k+1/2}^{n,0} - \hat{f}_{k-1/2}^{n,0}\right)$
- **Loop on** levels $\ell = 0 \nearrow L-1$
  - **Loop on** intervals $I_k^\ell \in \Omega^\ell$
    - $\hat{f}_{2k-1/2}^{\ell+1} = \hat{f}_{k-1/2}^\ell$
    - $\tilde{f}_{2k+1/2}^{\ell+1} = \mathcal{J}(x_{2k+\frac{1}{2}}^{\ell+1}; \hat{f}^\ell)$, (5)
    - **if** $I_k^\ell \notin \tilde{\Lambda}_\varepsilon$ interpolation
      - $\hat{f}_{2k+1/2}^{\ell+1} = \tilde{f}_{2k+1/2}^{\ell+1}$
      - $d_k^{\ell,n+1} = d_k^{\ell,n} = 0$
    - **else** precise computation, using fine data
      - $\hat{f}_{2k+1/2}^{\ell+1} = \hat{F}(\bar{u}_{2^{L-\ell+1}k}^L, \bar{u}_{2^{L-\ell+1}k+1}^L)$ (4)
      - $d_k^{\ell,n+1} = d_k^{\ell,n} - \frac{\Delta t}{h_{\ell+1}}\left(\hat{f}_{2k+1/2}^{\ell+1} - \tilde{f}_{2k+1/2}^{\ell+1}\right)$
    - *end* **If**
  - *End of* **loop on** intervals.
- *End of* **loop on** levels.

End of **loop on** time step.

We now recall the intermediate algorithm of thresholding and determination of the smoothness indicators, that is the set of indices $\tilde{\Lambda}_\varepsilon^{n+1}$. This is actually a tricky part in the whole algorithm. It relies on the size of the details at the given time and in the initial Harten's setting also on the fact that the equations are hyperbolic. This means that if the CFL condition is fulfilled spatial gradients will not propagate faster than one grid point per time step. It also allows to estimate the increase of the details in one time step meaning that not more than one level of refinement should be necessary from one time step to the next. The hyperbolicity assumption is not verified in the case of the second gradient equations and the robustness of the following strategy for the thresholding has been verified numerically:

Algorithm 4. *thresholding and smoothness indicators* $\tilde{\Lambda}_\varepsilon^n$

Initialization $\tilde{\Lambda}_\varepsilon^{n+1} = \cup_{\ell=0}^L \Omega^\ell$
Loop on levels $\ell = L \searrow 1$

- Tolerance $\varepsilon_\ell = 2^\ell \varepsilon$
- **Loop on** $I_k^\ell \in \Omega^\ell$
  - **if** $|d_k^\ell| < \varepsilon_\ell$ then $d_k^\ell = 0$ et $I_k^\ell \notin \tilde{\Lambda}_\varepsilon^{n+1}$
  - **else**
    - $I_{k+m}^\ell \in \tilde{\Lambda}_\varepsilon^{n+1}$ for $|m| < 2$
    - **if** $|d_k^\ell| > 2^p \varepsilon_\ell$ and **if** $\ell < L$ then $I_{2k}^{\ell+1} \cup I_{2k+1}^{\ell+1} \in \tilde{\Lambda}_\varepsilon^{n+1}$
  - *end* **If**
- *End of* **loop on** intervals.

End of **loop on** levels.

This algorithm defines a thresholding operator T_ϵ which acts on the multiresolution representations of $\bar{U}_M$. The thresholding parameter ϵ is level dependent and for the moment we choose $\epsilon_l = 2^{l-L}\epsilon$ which means we are keeping less details if go to a coarser level. To have a kind of measure for the efficiency of the thresholding we define

$$\text{rate of compression} = \frac{n_L}{\#(\tilde{\Lambda}_\varepsilon) + 2^{-L} n_L} \tag{6}$$

This quantity depends on time since the set of "non smooth indices" $\tilde{\mathcal{D}}_\varepsilon$ is computed anew at each time step.

5. Numerical experiments

We study the behaviour of our algorithm on the time evolution of an equilibrium solution with reflecting boundary conditions.

For this we compute an equilibrium solution on the finest level (Fig. 1). The computations are done on

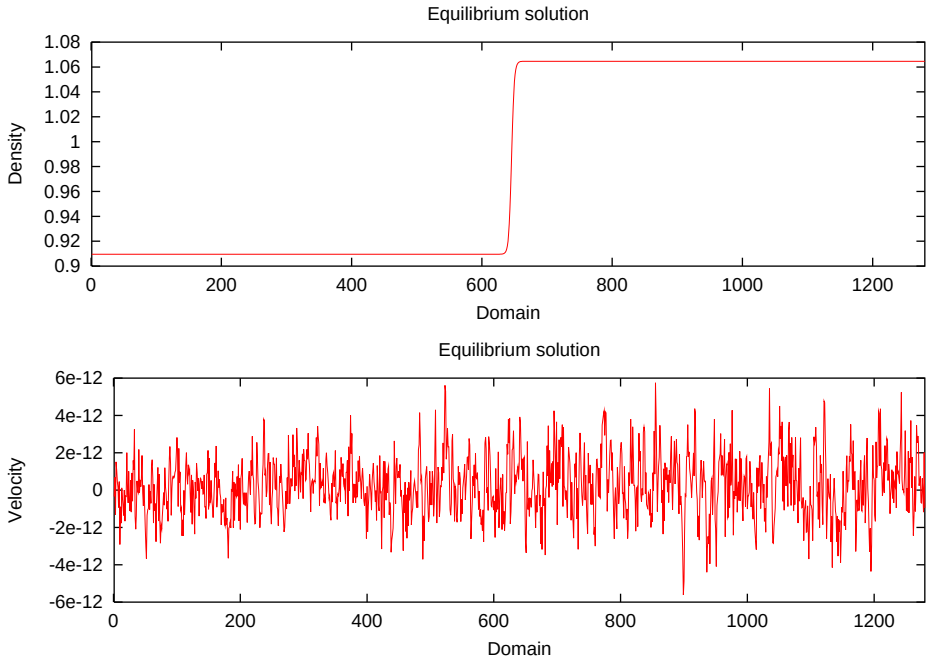

Figure 1. Equilibrium solution

a domain of $2.5 \cdot 10^{-5} m$ and 1280 cells on the finest level. We compute a time evolution of the equilibrium solution on the finest level without using any multiresolution. It is this reference solution to which we compare the solutions of the multiresolution algorithm. A time step of $7.0 \cdot 10^{-8}$ is used and comparisons are made after $1 \cdot 10^4$ and $1 \cdot 10^6$ time steps.

For zero threshold, that is all coefficients are kept, we cannot see any difference between the multiresolution and the reference solution in the density variable in (Fig. 2). The velocity profile already indicates, that this variable behaves differently than the density. After 10^4 time steps everything seems to be fine, but after 10^6 time steps it is obvious, that the multiresolution algorithm introduces some additional roundoff error (the multiresolution solution is the upper curve in the graph), even without doing any thresholding. For a threshold

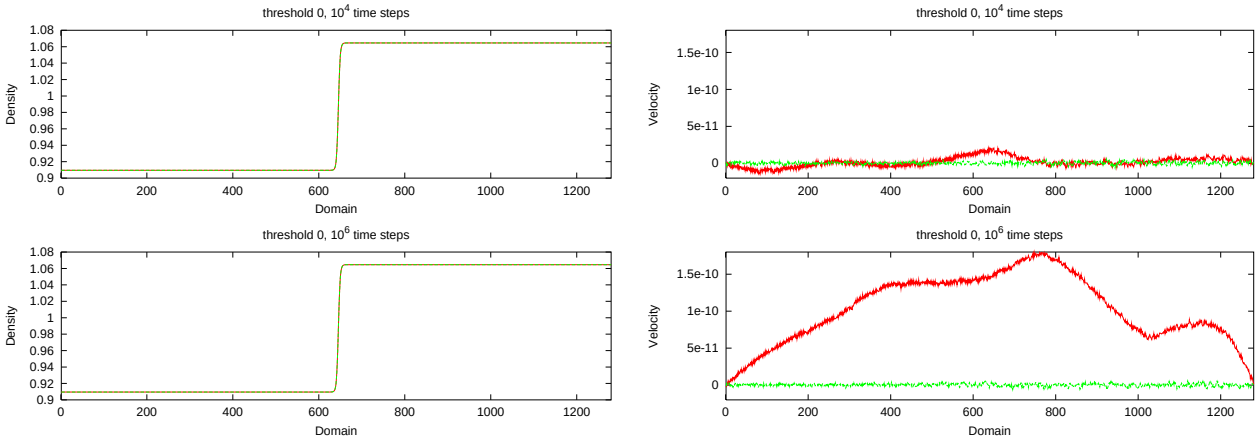

FIGURE 2. Density (on the left) and Velocity (on the right) variable: reference and multiresolution solution with no threshold

of 10^{-7} the density profile looks still fine (Fig. 3), both solutions, the reference and the multiresolution, are printed on top of each other, so it is not possible to distinguish between both. The situation is different for the velocity variable (Fig. 3). Already after 10^{-4} time steps the velocity variable shows a considerable deviation

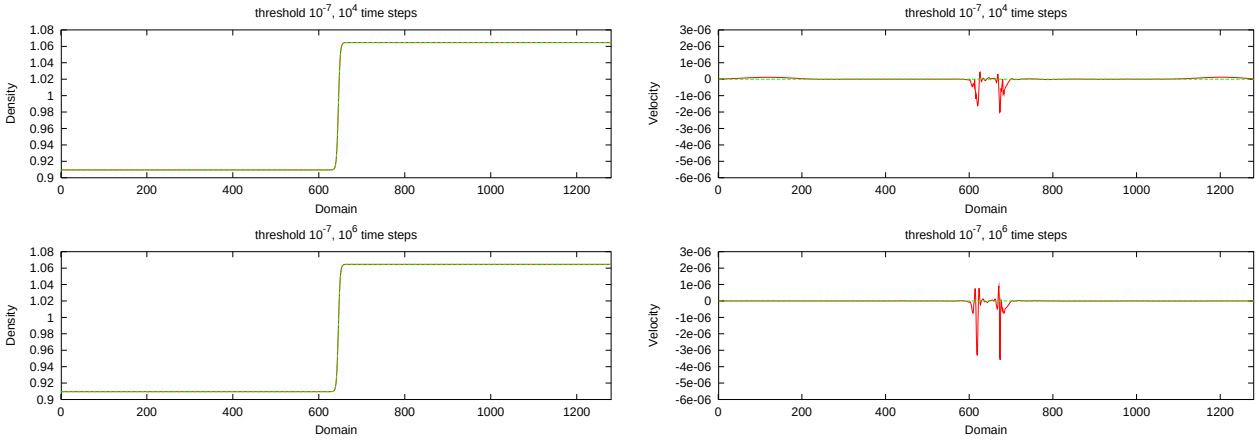

FIGURE 3. Density (on the left) and Velocity (on the right) variable: reference and multiresolution solution with threshold 10^{-7}

from the reference solution. But interestingly, this difference is not amplified, when we evolve in time. Even after 10^6 time steps, the difference is of the same order and also bounded in space to the region of the interface.

This different behaviour of the density and the velocity variable can also be observed for higher values of the threshold. The following two figures (Fig. 4) show the solution for a threshold of 10^{-5} and again, the difference in the velocity variable is large, but does not get amplified. Again, the same can be seen for an even larger threshold of 10^{-3} (Fig. 5).

To obtain a quantitative view of the error between the reference and the multiresolution solution, we have used different ways to calculate the error of the density and the velocity variable. For physical applications the

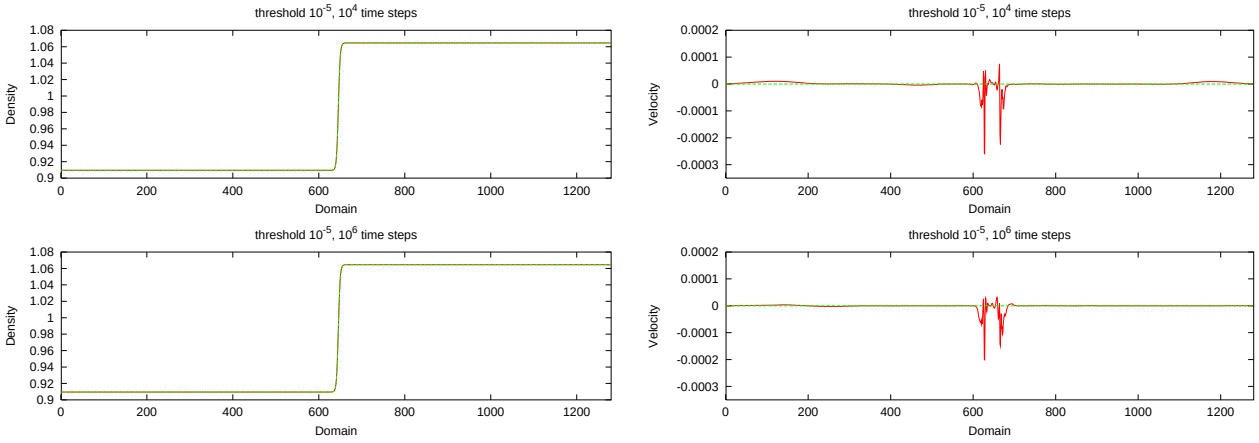

FIGURE 4. Density (on the left) and Velocity (on the right) variable: reference and multiresolution solution with threshold 10^{-5}

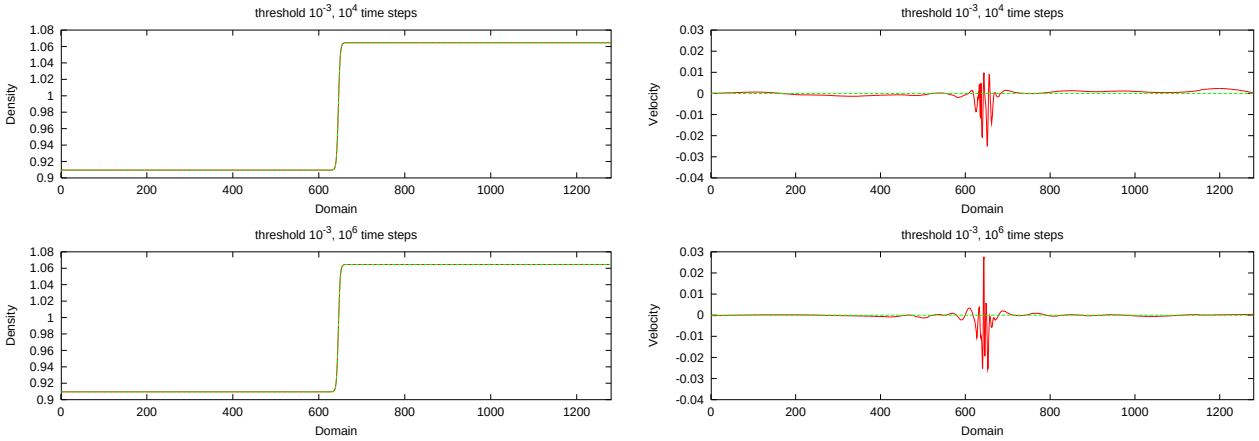

FIGURE 5. Density (on the left) and Velocity (on the right) variable: reference and multiresolution solution with threshold 10^{-3}

surface tension of the interface is a very important value which should not change if we use a multiresolution algorithm. The surface tension is given by $\int(\nabla\rho)^2$ and therefore, we use $\int(\nabla\rho_r - \nabla\rho_m)^2$ to measure the error in the gradient of the density between the reference solution ρ_r and the multiresolution solution ρ_m. The results are shown in (Fig. 6). The different points in the graph correspond to different number of time steps. Even with a high threshold, the error is negligible. This figure also shows, that even with a relatively small threshold we can obtain a high compression rate which also means a low number of exact flux evaluations!

Since the error of the velocity variable seems to be bounded, we look for the behaviour of the maximum of the difference between the reference and the multiresolution solution, that is $\max(v_r - v_m)$. In (Fig. 6) a similar behaviour of the error of the velocity in this measure compared to the error of the density variable can be observed.

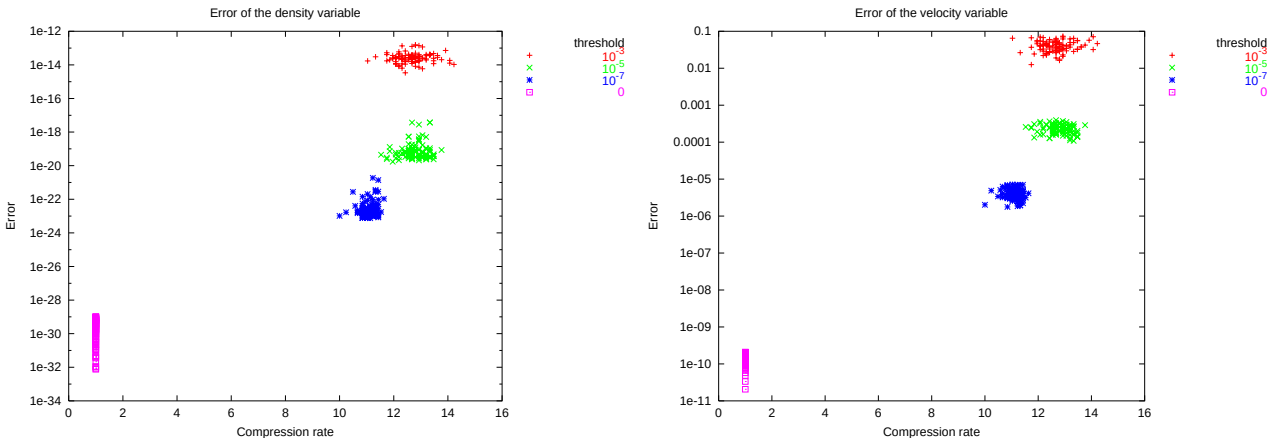

FIGURE 6. Error of the velocity solution depending on different thresholds

6. CONCLUSIONS

The multiresolution algorithm has been successfully implemented in the case of second gradient equations. Its robustness and performances in terms of flux computation savings have been numerically tested and the results are very encouraging. We are now studying an improvement of this scheme consisting in computing the solution directly on an hybrid grid — fine where the solution varies a lot and coarser in smooth areas. This fully adaptive scheme has been extensively studied both from the theoretical and numerical sides in the case of hyperbolic equations in [3] and its implementation in the case of second gradient equations is under progress.

REFERENCES

[1] G. Chiavassa and R. Donat, *Point value multi-scale algorithms for 2d compressible flows*, Tech. report, UCLA CAM Report 00-17, 1999, also submitted to Siam J. Sci. Comput., Oct. 99.
[2] A. Cohen, N. Dyn, S.M. Kaber, and M. Postel, *Multiresolution schemes on triangles for scalar conservation laws*, Journal of Computational Physics **1** (2000), no. 161, 264–286.
[3] A. Cohen, S.M. Kaber, S. Müller, and M. Postel, *Fully adaptive multiresolution finite volume schemes for conservation laws*, submitted, Prepublications du Laboratoire d' Analyse Numérique (2000), no. R0009.
[4] Frédéric Coquel and Oliver Großhans, Cemracs'98.
[5] W. Dahmen, B. Gottschlich-Müller, and S. Müller, *Multiresolution schemes for conservation laws*, Tech. report, Bericht Nr. 159, IGPM, RWTH Aachen, 1998.
[6] G. DEHAIS, B. MATHIEU, D. JAMET, O. LEBAIGUE, and F. COQUEL, *Numerical method for systems of conservation laws of mixed type*, Proceedings CEMRACS99 (F. Coquel et S. Cordier, ed.), ESAIM, 2001.
[7] A. Harten, *Adaptive multiresolution schemes for shock computations*, J. Comp. Phys. **115** (1994), 319–338.
[8] D. Jamet, *Théorie du second gradient: un outil pour la simulation numérique directe d'écoulments diphsiques liquid-vapeur*, Ph.D. thesis, Ecole Centrale Paris, 1998.