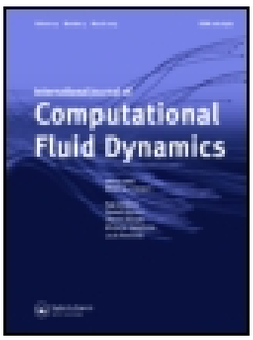

International Journal of Computational Fluid Dynamics

ISSN: 1061-8562 (Print) 1029-0257 (Online) Journal homepage: http://www.tandfonline.com/loi/gcfd20

Numerical investigation of BB-AMR scheme using entropy production as refinement criterion

Thomas Altazin, Mehmet Ersoy, Frédéric Golay, Damien Sous & Lyudmyla Yushchenko

To cite this article: Thomas Altazin, Mehmet Ersoy, Frédéric Golay, Damien Sous & Lyudmyla Yushchenko (2016): Numerical investigation of BB-AMR scheme using entropy production as refinement criterion, International Journal of Computational Fluid Dynamics

To link to this article: http://dx.doi.org/10.1080/10618562.2016.1194977

Published online: 07 Jun 2016.

INTERNATIONAL JOURNAL OF COMPUTATIONAL FLUID DYNAMICS, 2016
http://dx.doi.org/10.1080/10618562.2016.1194977

Taylor & Francis
Taylor & Francis Group

Numerical investigation of BB-AMR scheme using entropy production as refinement criterion

Thomas Altazin[a], Mehmet Ersoy[a], Frédéric Golay[a], Damien Sous[b] and Lyudmyla Yushchenko[a]

[a]Institut de Mathématique de Toulon (IMATH), Université de Toulon, La Garde, France; [b]Université de Toulon, Aix-Marseille Université, CNRS/INSU, IRD, Mediterranean Institute of Oceanography (MIO), Marseille, France

ABSTRACT

In this work, a parallel finite volume scheme on unstructured meshes is applied to fluid flow for multidimensional hyperbolic system of conservation laws. It is based on a block-based adaptive mesh refinement strategy which allows quick meshing and easy parallelisation. As a continuation and as an extension of a previous work, the useful numerical density of entropy production is used as mesh refinement criterion combined with a local time-stepping method to preserve the computational time. Then, we numerically investigate its efficiency through several test cases with a confrontation with exact solution or experimental data.

ARTICLE HISTORY
Received 28 September 2015
Accepted 23 May 2016

KEYWORDS
multidimensional hyperbolic system; finite volume; adaptive mesh refinement; numerical density of entropy production; local time stepping

1. Introduction

The numerical simulation of two-phase fluid flows in complex three-dimensional (3D) configurations remains a challenging task constrained by two conflicting goals: the precise description of each involved physical processes over the whole simulated domain down to the smallest spatio-temporal scales of the fluid motion and the computational cost. In reality, a compromise is generally made according to the requirements by the studied physical cases (for a more detailed presentation of the numerical/physical strategic issues of such compromise in the particular case of wave impact on rigid structures, the reader can refer, for instance, to Golay et al. [2015]). Our research work focuses on a multidimensional numerical scheme able to accurately solve nonlinear hyperbolic systems of conservation laws while preserving the computational time. This scheme has been first presented by Ersoy, Golay, and Yushchenko (2013) in the 1D case and further extended to three dimensions and confronted to experimental data by Golay et al. (2015).

The present research work is dedicated to the study of the numerical performance of the model through several test cases. The model framework and the related scientific issues having been detailed in recent above-mentioned publications. We briefly recall here the overall principles of the block-based adaptive mesh refinement (BB-AMR) scheme, while the main numerical features of the model are presented in the next section. The equations system of interest is

$$\begin{cases} \frac{\partial \boldsymbol{w}(t)}{\partial t} + \nabla \cdot \boldsymbol{f}(t, \boldsymbol{w}) = \boldsymbol{G}, & (t, x) \in \mathbb{R}^+ \times \mathbb{R}^d \\ \boldsymbol{w}(0, x) = \boldsymbol{w}_0(x), & x \in \mathbb{R}^d, \end{cases} \tag{1}$$

where $\boldsymbol{w}$, $\boldsymbol{f}$, and $\boldsymbol{G}$ stands, respectively, for conservative variables, flux, and source.

As a well-known result, the uniqueness of the (weak) solution is lost even if the initial data are smooth. It can be recovered by completing system (1) with an entropy inequality of the following form:

$$S = \frac{\partial s(\boldsymbol{w})}{\partial t} + \nabla \cdot \boldsymbol{\psi}(\boldsymbol{w}) \leq 0\,, \tag{2}$$

where (s, ψ) stands for a convex entropy–entropy flux pair. Even if we are not able to prove the uniqueness in the multidimensional case, this inequality allows to select the physical relevant solution and provides a 'smoothness' indicator since the entropy satisfies a conservation equation only in regions where the solution is smooth and an inequality when the solution develops discontinuities. Thus, the discrete quantity S can always be considered as a measure of the amount of violation of the entropy equation (as pointed out in Berger and Oliger, 1984; Houston et al., 1999; Karni, Kurganov, and Petrova, 2002; Karni and Kurganov, 2005; Ersoy, Golay, and Yushchenko, 2013). As already done in Ersoy, Golay, and Yushchenko (2013), S, which is called the

CONTACT Frédéric Golay frederic.golay@univ-tln.fr

numerical density of entropy production, can be used as a smoothness indicator providing 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).

A major issue of many modelling challenges is to accurately simulate processes over very large ranges in spatial scale inexorably leading to heavy computational time. Thus, in order to efficiently implement the local time-stepping method in a parallel framework, the approach taken here is to equip our adaptive mesh refinement (AMR) technique with a block-based (BB-AMR) strategy. Sufficient spatio-temporal resolution can be reached in a reasonable CPU time, which is especially useful for hyperbolic problems generally requiring fine CPU-consuming uniform mesh. Roughly speaking, the BB-AMR technique provides an efficient control of the shared memory leading to well-balanced computational time between cores by domain-like decomposition. Nevertheless, data locality is critical to obtain good performance since the memory access times are not uniform and may become expensive. Consequently, the re-meshing step has to be carefully managed. Taking advantage of such a block-based structure (as in domain decomposition), we can define two different time steps: the first one is based on the Courant–Friedrich–Levy (CFL) condition (i.e computed through the finest cells), while the second one is defined at the level of the block. The stability of the scheme is thus respected for any time and the re-meshing cost is minimised.

The first part of this paper (Section 2) is dedicated to the summarised presentation of the multidimensional finite volume solver on unstructured meshes with a BB-AMR technique. The second and the main part of the paper (Section 3) consists of an evaluation of the method performance through a series of test cases. First, as an addendum to Ersoy, Golay, and Yushchenko (2013), our 1D solver is tested on the well-known Woodward and Colella interacting blast-wave problem (Woodward, 1982; Woodward and Colella, 1984). Additional tests are performed on classical 2D Riemann problem for polytropic Euler equations which have been extensively studied (Zhang and Zheng, 1990; Schulz-Rinne, Collins, and Glaz, 1993; Lax and Liu, 1998; Kurganov and Tadmor, 2002; Liska and Wendroff, 2003). Solutions are composed of 19 possible geometric configurations connected by shocks, rarefactions, and contacts (see, for instance, Lax and Liu, 1998). The so-called configuration 17 is used here. It consists of a solution composed of a shock, a rarefaction wave, and two contact discontinuities. The final set of numerical tests concerns the 2D and 3D simulation of two-phase flows on a dam-break problem. For those cases, our model is based on a 3D compressible, low Mach, two-phase flows model with a linearised 'artificial pressure' law in which physical relevancy has already been demonstrated in the context of highly dynamical and aerated flows for breaking and impacting waves by Sambe et al. (2011) or Golay et al. (2015).

2. Finite volume approximation for hyperbolic conservation laws

This section summarises the main features of our method, including the semi-discrete finite volume numerical approximation of a general nonlinear hyperbolic equation (Equation (1)), the time integration, and mesh refinement procedure. Note that $\boldsymbol{w}$, $\boldsymbol{f}$, and $\boldsymbol{G}$ are the conservative variables, the flux, and the source term, respectively, and $d \in [\![1, 3]\!]$.

2.1. Multidimensional finite volume approximation

The computational domain $\Omega \subset \mathbb{R}^d$ is split into a set of control volumes, also referred to as cells, $\Omega = \cup_k C_k$ of mesh size $|C_k|$. For the sake of simplicity, the source term is here omitted.

On a given cell C_k, noting $\boldsymbol{w}_k(t)$,

$$\boldsymbol{w}_k(t) \simeq \frac{1}{|C_k|} \int_{C_k} \boldsymbol{w}\,(t, x)\; d\boldsymbol{x},$$

the approximation of the mean value of the unknown $\boldsymbol{w}(t, x)$ on C_k at time t, and integrating (1) over each cell, we obtain

$$\int_{C_k} \frac{\partial \boldsymbol{w}(t)}{\partial t} + \sum_a \int_{\partial C_{k/a}} \boldsymbol{f}(t, \boldsymbol{w}) \cdot \boldsymbol{n}_{k/a} = 0, \tag{3}$$

where $\boldsymbol{n}_{k/a}$ denotes the unit normal vector on the boundary $\partial C_{k/a}$ between cells k and a.

Next, $\boldsymbol{F}\left(\boldsymbol{w}_k(t), \boldsymbol{w}_a(t), \boldsymbol{n}_{k/a}\right)$, the flux approximation, being written as

$$\boldsymbol{F}\left(\boldsymbol{w}_k(t), \boldsymbol{w}_a(t), \boldsymbol{n}_{k/a}\right) \approx \frac{1}{|\partial C_{k/a}|} \int_{\partial C_{k/a}} \boldsymbol{f}(t, \boldsymbol{w}) \cdot \boldsymbol{n}_{k/a} ds\,,$$

the semi-discrete finite volume approximation of Equation (1) (see, for instance, Godlewski and Raviart, 1996; Toro, 2009; Eymard and Herbin, 2000) is obtained:

$$\frac{\partial \boldsymbol{w}_k(t)}{\partial t} + \frac{1}{|C_k|} \sum_a |\partial C_{k/a}| \boldsymbol{F}\left(\boldsymbol{w}_k(t), \boldsymbol{w}_a(t), \boldsymbol{n}_{k/a}\right) = 0, \tag{4}$$

where $\boldsymbol{F}\left(\boldsymbol{w}_k(t), \boldsymbol{w}_a(t), \boldsymbol{n}_{k/a}\right)$ is defined via the Godunov solver, i.e. it is computed with the exact solution of the 1D Riemann problem at the interface k/a with the states

$w_k(t)$ and $w_a(t)$ (for further details, see, for instance, Toro, 2009; Golay and Helluy, 2007).

Equation (1) is completed with the entropy inequality of Equation (2), where

$$(\nabla_w \boldsymbol{\psi}(s(\boldsymbol{w})))^t = (\nabla_w s(\boldsymbol{w}))^t \; D_{\boldsymbol{w}} f(\boldsymbol{w}) \, .$$

Following Ersoy, Golay, and Yushchenko (2013), the production of entropy (2) is approximated after each time step using the same semi-discrete finite volume scheme (4) and the same time integration scheme. The obtained discrete quantity, called the numerical density of entropy production, is then used as a mesh refinement criterion (see Section 2.3).

For further details on the construction of the numerical scheme, we refer to Ersoy, Golay, and Yushchenko (2013), since the definition of the numerical fluxes reduce to a 1D computational at each interface k/a. Up to now, the first- and second-order Godunov schemes are implemented.

2.2. Time integration

The time integration of Equations (4) and (2) can be achieved in a classical way either by a Runge–Kutta or Adams–Bashforth scheme. Note that, even if the Adams–Bashforth scheme is known to be less stable and less accurate, it can be easily handled in the framework of local time stepping to save computational time (see e.g. Altmann et al., 2009; Ersoy, Golay, and Yushchenko, 2013).

2.2.1. Runge–Kutta schemes

By integrating Equation (4) and (2) during the time step $]t_n, t_{n+1}[$ of length δt_n and by evaluating the numerical fluxes at time t_n, the well-known first-order Euler's scheme is obtained:

$$\begin{aligned} \boldsymbol{w}_k(t_{n+1}) &= \boldsymbol{w}_k(t_n) \\ &\quad - \frac{\delta t_n}{|C_k|} \sum_a |\partial C_{k/a}| \boldsymbol{F}\left(\boldsymbol{w}_k(t_n), \boldsymbol{w}_a(t_n), \boldsymbol{n}_{k/a}\right) . \end{aligned} \quad (5)$$

In order to increase the accuracy, a second-order Runge–Kutta method can be used as follows:

$$\begin{aligned} \boldsymbol{w}_k(t_{n+1}) &= \boldsymbol{w}_k(t_n) \\ &\quad - \frac{\delta t_n}{|C_k|} \sum_a |\partial C_{k/a}| \boldsymbol{F}\left(\boldsymbol{w}_k(t_{n+1/2}), \boldsymbol{w}_a(t_{n+1/2}), \boldsymbol{n}_{k/a}\right) , \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{w}_k(t_{n+1/2}) &= \boldsymbol{w}_k(t_n) \\ &\quad - \frac{\delta t_n}{2|C_k|} \sum_a |\partial C_{k/a}| \boldsymbol{F}\left(\boldsymbol{w}_k(t_n), \boldsymbol{w}_a(t_n), \boldsymbol{n}_{k/a}\right) . \end{aligned}$$

The numerical density of entropy production (2) is then obtained with a second-order Runge–Kutta scheme.

2.2.2. Adams–Bashforth schemes

The Adams–Bashforth method of order m consists in replacing the numerical flux of Equation (4) by a polynomial interpolation of the same order (Hairer, Nørsett, and Wanner, 1993). For example, the second-order Adams–Bashforth method reads

$$\begin{aligned} \boldsymbol{w}_k(t_{n+1}) &= \boldsymbol{w}_k(t_n) \\ &\quad - \frac{\delta t_n}{|C_k|} \sum_a |\partial C_{k/a}| \boldsymbol{F}\left(\boldsymbol{w}_k(t_n), \boldsymbol{w}_a(t_n), \boldsymbol{n}_{k/a}\right) \\ &\quad - \frac{\delta t_n^2}{2\delta t_{n-1}\,|C_k|} \Bigg(\sum_a |\partial C_{k/a}| \boldsymbol{F}\left(\boldsymbol{w}_k(t_n), \boldsymbol{w}_a(t_n), \boldsymbol{n}_{k/a}\right) \\ &\quad - \sum_a |\partial C_{k/a}| \boldsymbol{F}\left(\boldsymbol{w}_k(t_{n-1}), \boldsymbol{w}_a(t_{n-1}), \boldsymbol{n}_{k/a}\right) \Bigg) . \end{aligned}$$

The Adams–Bashforth methods of order 2 and 3 are stable for CFL condition less than one (Allahviranloo, Ahmady, and Ahmady, 2007). Practically, for stability purpose, we limit our applications to the second-order scheme.

We also perform the same discretisation above for entropy production (2).

2.3. Mesh refinement criterion and BB-AMR technique

By contrast to the 1D case, defining a robust mesh refinement parameter for 2D and 3D configurations is not enough to design a suitable numerical solver. The treatment of data is also a crucial point and in particular the way to share the memory in a parallel process. This point is handled in a hierarchical block-based way that we have called BB-AMR. First, we present the main strategy to adapt the mesh with respect to the numerical density of entropy production and then how to manage data in an efficient way.

2.3.1. Numerical entropy production as mesh refinement criterion

The efficiency of the numerical density of entropy production as a relevant mesh refinement parameter have been already demonstrated in a previous work (Ersoy, Golay, and Yushchenko, 2013; Golay et al., 2015). It has been numerically observed (and from theoretical considerations) that the production of the numerical density of entropy is almost zero for smooth solution and non-positive when the solution develops discontinuities. As a consequence, the mesh is automatically and proportionally (with respect to the production) refined inside an area

where the production is non-zero. More precisely, Ersoy, Golay, and Yushchenko (2013) have demonstrated that, for the 1D gas dynamics equation, the support of the relative error coincides with the support of the numerical density of entropy production. The extension towards the multidimensional case is detailed in the case of two-fluid flows in Golay et al. (2015).

According to the finite volume approximation defined in Section 2, a local numerical entropy production S_k^n is computed on each cell at time t_n and compared to the average entropy production $\bar{S} = \frac{1}{|\Omega|}\sum_k S_k^n$. Two coefficients $0 \leq \alpha_{\min} \leq \alpha_{\max} \leq 1$ are thus defined to determine the ratio of numerical production of entropy leading to mesh refinement or mesh coarsening.

For each cell C_k,

- if $S_k^n > \bar{S}\alpha_{\max}$, the mesh is refined and split; and
- if $S_k^n < \bar{S}\alpha_{\min}$, the mesh is coarsened whenever it is possible following the rule defined hereinafter.

The threshold parameters $\alpha_{\max}$ and $\alpha_{\min}$ are empirically determined, according to the simulated case requirements, to reach a relevant compromise between computational cost and accuracy. Thus $\alpha_{\min}$ and $\alpha_{\max}$ allow to set a percentage of mesh refinement and mesh coarsening with respect to the quantity $\bar{S}$. For instance, the smaller the $\alpha_{\min}$ and $\alpha_{\max}$ are, the more accurate the results are at the expense of CPU time.

2.3.2. Mesh refinement process

For the 1D case, the local mesh refinement procedure is constructed following dyadic tree applied at each time step. 'Macro cells' are used to be easily refined by generating hierarchical grids. Each cell can be split into two. Dyadic cells graph are thus produced, in basis 2 numbering, to allow a quick computing scan to determine the adjacent cells. For stability reasons, the mesh refinement level cannot exceed 2 between two adjacent cells. More details can be found in Ersoy, Golay, and Yushchenko (2013).

The 3D extension of the mesh refinement procedure is a challenging task. The present section aims to present the parallelisation implementation to demonstrate the interest of local time-stepping and BB-AMR schemes in terms of CPU time but not to study the performance of the parallelisation procedure for itself. Many interesting works on AMR techniques have been presented for 2D Cartesian grid for quadtree (Berger and Colella, 1989; Yiu et al., 1996; Min and Gibou, 2007; Zhang and Wu, 2011), octree for 3D simulations (Losasso, Gibou, and Fedkiw, 2004; Fuster et al., 2009), and anisotropic AMR (Coupez and Hachem, 2013; Hachem et al., 2013). The extension from 1D to 3D leads to natural octree meshing. But, the presence of a complex moving interface (composed of rarefaction, shocks and/or contacts) implies to re-mesh at each time step, which is obviously a costly process, even if some AMR tools like Octor or Gerris (Tu, O'Hallaron, and Ghattas, 2005; Fuster et al., 2009) are very powerful and well parallelised. Guided by the need to reach a relevant compromise between the contradictory aims of solution accuracy and computing speed, a Cartesian block-based mesh approach is introduced, somewhat like in Williamschen and Groth (2013) and Zheng and Groth (2012). The computational domain is divided into several blocks, each corresponding to the initial unstructured mesh composed of hexahedral cells. These blocks are, in their turn, split in a Cartesian way ($2^{L-1}n_x$, $2^{L-1}n_y$, $2^{L-1}n_z$), where (n_x, n_y, n_z) stands for the initial block discretisation, and L the level of mesh refinement. For each refined cells (or blocks), averaged values are projected on each sub-cell and fluxes are computed as simply as possible to avoid heavy computation. Then, in order to balance the CPU load, the cells of each block are re-distributed in a fixed number of domains (part of the total computational domain) according to the Cuthill–McKee numbering (see Figure 1). The number of domains being fixed, each domain is loaded in a given MPI process and all MPI processes are then distributed on a fixed number of cores (not necessarily the same). The re-numbering and re-meshing being expensive, the mesh is finally kept constant on a time interval, called AMR time step, given by the smallest block (rather by the smallest cell) and the maximum velocity. More details on the 3D BB-AMR are given in Golay et al. (2015).

3. Numerical results

This section is dedicated to the numerical validation of the BB-AMR scheme through several multidimensional test cases.

3.1. Euler equations of gas dynamics

Let us consider a compressible perfect gas confined in a domain Ω. The governing equations for the motion of the compressible gas in $[0, T] \times \Omega$ are the so-called Euler equations:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0, \\ \dfrac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot \left(\rho \boldsymbol{u} \otimes \boldsymbol{u} + p \boldsymbol{I}_d\right) = 0, \\ \dfrac{\partial \rho E}{\partial t} + \nabla \cdot \left(\left(\rho E + p\right) \boldsymbol{u}\right) = 0, \end{cases} \tag{6}$$

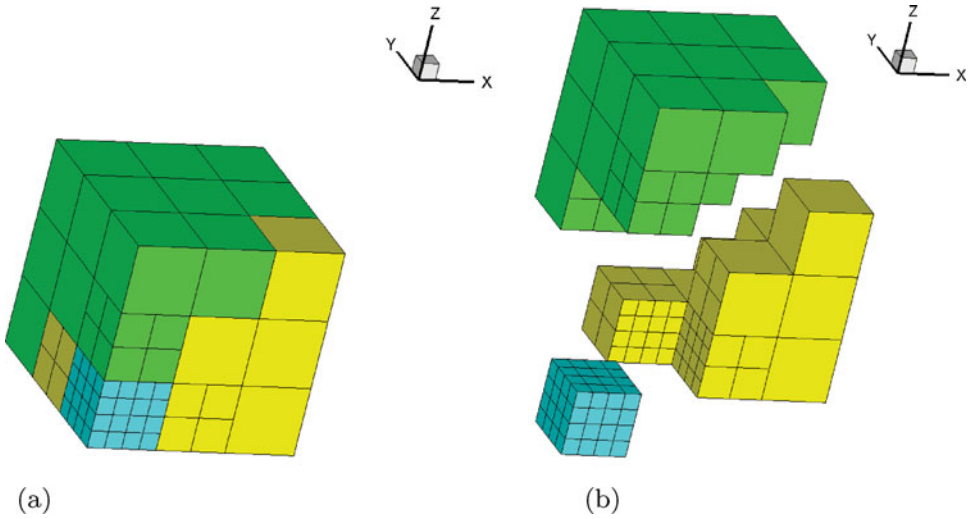

Figure 1. Example of three-dimensional block-based mesh with 3 domains and 27 blocks (Golay et al., 2015): (a) block-based mesh; (b) domain decomposition.

where ρ is the density, $p = (\gamma - 1)\rho\varepsilon$ is the pressure, u the velocity, and E the total energy defined by

$$E = \varepsilon + \frac{\|\boldsymbol{u}\|^2}{2}\,.$$

Here ε stands for the internal specific energy and γ (set to 1.4) is the ratio between specific heats.

System (6) is written as a system of conservation laws as in Equation (1) completed with the entropy inequality (2) where the convex entropy $s(\boldsymbol{w}) = s\,(\rho, \rho\boldsymbol{u}, \rho E)$ and the entropy flux $\psi\,(\boldsymbol{w})$ are classically given by the following relations:

$$s(\boldsymbol{w}) = -\rho \ln\left(\frac{p}{\rho^\gamma}\right), \quad \psi\,(\boldsymbol{w}) = u\, s(\boldsymbol{w})\,. \tag{7}$$

Let us note that even if system (6) is strictly hyperbolic on the set $\{\rho > 0\}$, the previous quantities defined in Equation (7) make sense for $\rho \geq 0$.

3.1.1. One-dimensional Riemann problem

The first test case is based on the Woodward–Colella blast-wave benchmark case. This 1D test problem, which was initially introduced in Woodward (1982), is one of the most challenging test cases to solve on a uniform grid even with a very large number of cells. In particular, it illustrates the strong relationship between the accuracy of the overall flow solution and the thinness of discontinuities on the grid involving multiple interactions of discontinuities (shocks and contact discontinuities) and rarefactions.

The initial condition consists of three constant states:

$$x \in [-1, 1], \quad \rho(0, x) = 1, \quad u(0, x) = 0,$$

$$p(0, x) = \begin{cases} 1000, & x \leqslant 0.1, \\ 0.01, & 0.1 < x \leqslant 0.9, \\ 100, & x > 0.9 \end{cases}$$

on a computational domain $[0, 1]$ with prescribed reflecting boundary conditions.

A full set of comparisons with different approximations is proposed here, using the following settings and notations:

- Both the first- and second-order schemes are compared. Thus, we will refer to AB1, AB2, and RK2 as the first- and second-order Adams–Basforth schemes and the second-order Runge–Kutta scheme, respectively. Both AB2 and RK2 use a MUSCL reconstruction (Golay and Helluy, 2007). Computations are performed on a dynamic grid with a uniform time step, except

  – if the case acronym ends with 'U', which refers to a uniform fixed grid or,
  – if the acronym ends with 'M', which refers to the local time-stepping algorithm (see Ersoy, Golay, and Yushchenko [2013] for further details on the local time stepping adaptation).

  The CPU time is noted as follows:
  – cpu_g for the uniform (or global) time stepping,
  – cpu_l for the local time stepping.

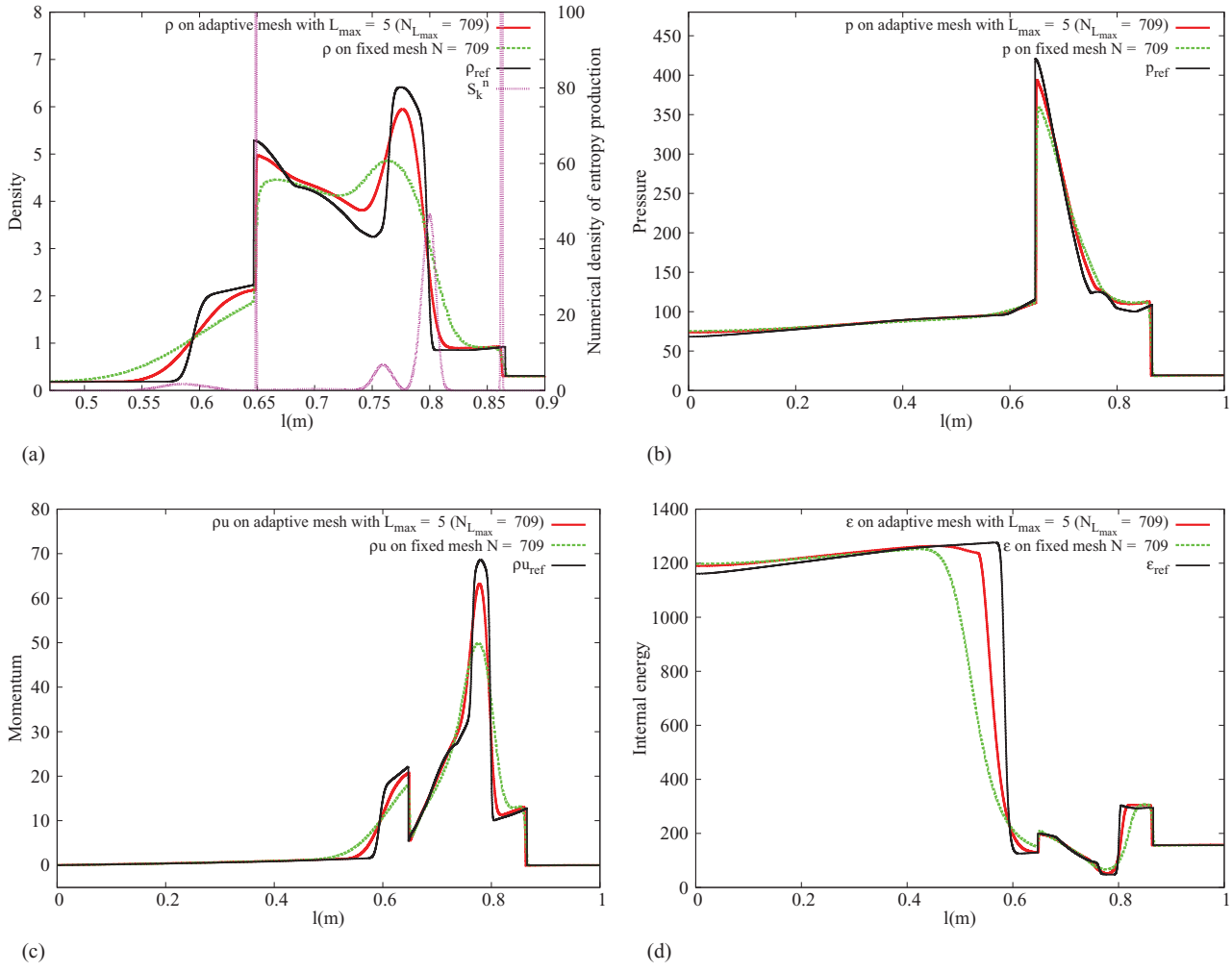

Figure 2. Comparison between AB1 (green lines), AB1U (red lines), and reference (black lines) cases for the density (a), the pressure (b), the momentum (c), and the internal energy (d) at time $t = 0.038$ s with $L_{\max} = 5$. Numerical density of entropy production (in absolute value) is also plotted in (a): (a) density and numerical density of entropy production; (b) pressure; (c) momentum; (d) internal energy.

- A reference solution (black line in Figure 2) is computed on a uniform grid with 20, 000 cells using the RK2U scheme.
- For all numerical tests in this section, the following parameters are used:

CFL	: 0.25
Simulation time (s)	: 0.038
Initial number of cells	: 200
Maximum level of mesh refinement	: $L_{\max}$
Mesh refinement parameter $\alpha_{\max}$	: 0.01
Mesh coarsening parameter $\alpha_{\min}$	: 0.001
Mesh refinement parameter $\bar{S}$	: $\frac{1}{\lvert\Omega\rvert}\sum_{k_b} S^n_{k_b}$

- Adaptive numerical solutions are compared to those computed on uniform fixed grid. For a relevant 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 $L_{\max}$ which is the maximum level of refinement.
- Each presented result displays a positive density. Thus, for the sake of visual commodity, the numerical density of entropy production will be depicted in absolute value on the density plots.
- To study the numerical convergence of AB1, AB1U, AB1M, RK2U, AB2U, AB2M, and RK2 cases, the discrete l^1_x-norm is used for density, momentum, pressure, and internal energy error (which is representative of the scheme efficiency, in particular for low-density flows).
- Tables of schemes performance are displayed in Tables 1 and 2. These tables are useful to compare the computational times and, in particular, to assess the expression 'saving the computational time keeping the same order of accuracy'. Since the errors of

Table 1. Convergence tests for the first-order AB1 scheme, l_x^1-norm at final time with respect to the averaged number of cells $N_{L_{\max}}$ of ρ, u, p, and ε, cpu_g for global time stepping and cpu_l local time stepping, N_{T_f} being the number of cells at the final time.

$L_{\max}$	$\|\rho-\rho_{ex}\|_{l_x^1}$	$\|p-p_{ex}\|_{l_x^1}$	$\|u-u_{ex}\|_{l_x^1}$	$\|\varepsilon-\varepsilon_{ex}\|_{l_x^1}$	N_{T_f}	cpu_g	cpu_l
1	0.645E+00	0.259E+02	0.104E+01	0.279E+03	200	1.30	1.30
2	0.528E+00	0.192E+02	0.923E+00	0.208E+03	289	3.38	2.48
3	0.409E+00	0.141E+02	0.801E+00	0.149E+03	435	9.16	6.21
4	0.294E+00	0.104E+02	0.679E+00	0.102E+03	697	25.78	17.61
5	0.197E+00	0.796E+01	0.588E+00	0.678E+02	1169	73.22	50.98

the AB1 and AB1M (respectively, AB2 and AB2M) schemes are similar, only the former is presented in Table 1 (respectively, Table 2).

Figure 2 depicts the solution profiles for density (Figure 2(a), together with numerical density of entropy production S_k^n), pressure (Figure 2(b)), momentum (Figure 2(c)), and internal energy (Figure 2(d)) for the AB1U, AB1, and reference cases computed with $L_{\max} = 5$ and 709 cells (in average for the AB1 case). Figure 2(a) first demonstrates the relevancy of the entropy production as refinement criterion: the stronger the density gradient, the larger the density of entropy production is. Each of the plotted profiles then shows the solution improvement provided by the use of adaptive mesh.

Figure 3 plots, for both the first-order (left) and second-order (right) schemes, a mesh convergence study based on the evolution of $\|\varepsilon - \varepsilon_{ex}\|_{l_x^1}$ discretisation error. As already noticed in Ersoy, Golay, and Yushchenko (2013), 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 α_{maax} at the expense of the CPU time. Finally, one notes the clear improvement of mesh convergence for both the first- and second-order schemes. In addition, we provide the numerical error for the density, pressure, velocity, and energy for both the first- and second-order schemes in Tables 1, 2 and 3.

It is here demonstrated that the adaptive grid strategy using the numerical density of entropy production can considerably increase the accuracy without imposing a large number of cells and related extra CPU time required in solving the problem on a uniform grid.

Table 3. Convergence tests for the first- and second-order schemes.

Rate	$\|\rho-\rho_{ex}\|_{l_x^1}$	$\|p-p_{ex}\|_{l_x^1}$	$\|u-u_{ex}\|_{l_x^1}$	$\|\varepsilon-\varepsilon_{ex}\|_{l_x^1}$
AB1U	0.35	0.60	0.41	0.47
AB1	0.67	0.67	0.33	0.80
AB1M	0.66	0.68	0.35	0.81
RK2U	0.44	0.70	0.55	0.53
AB2U	0.43	0.65	0.42	0.51
RK2	0.90	0.75	0.48	0.98
AB2	0.87	0.77	0.48	0.95
AB2M	0.77	0.72	0.39	0.91

3.1.2. *Two-dimensional Riemann problem*

Our tests are now extended to the 2D Riemann problem on the unit square with the following initial data:

$$(\rho, u, v, p)(0, x, y) = \begin{cases} (1, 0, -0.4, 1) & \text{if } x > 0.5 \text{ and } y > 0.5, \\ (2, 0, -0.3, 1) & \text{if } x < 0.5 \text{ and } y > 0.5, \\ (1.0625, 0, 0.2145, 0.4) & \text{if } x < 0.5 \text{ and } y < 0.5, \\ (0.5197, 0, -1.1259, 0.4) & \text{if } x > 0.5 \text{ and } y < 0.5. \end{cases}$$

In this configuration, the solution consists of two contacts (north and south), a shock (west), and a rarefaction (east) as displayed in Figure 4. A reference solution is computed on a uniform grid with 1, 000, 000 cells using the RK2U scheme. Figure 5 presents the reference solution at time

Table 2. Convergence tests for the second-order AB2 scheme, l_x^1-norm at final time with respect to the averaged number of cells $N_{L_{\max}}$ of ρ, u, p, and ε, cpu_g for global time stepping and cpu_l local time stepping, N_{T_f} being the number of cells at the final time.

$L_{\max}$	$\|\rho-\rho_{ex}\|_{l_x^1}$	$\|p-p_{ex}\|_{l_x^1}$	$\|u-u_{ex}\|_{l_x^1}$	$\|\varepsilon-\varepsilon_{ex}\|_{l_x^1}$	N_{T_f}	cpu_g	cpu_l
1	0.548E+00	0.193E+02	0.872E+00	0.217E+03	200	1.31	1.36
2	0.421E+00	0.132E+02	0.706E+00	0.156E+03	282	3.32	2.46
3	0.301E+00	0.929E+01	0.566E+00	0.106E+03	418	8.82	5.64
4	0.202E+00	0.672E+01	0.448E+00	0.700E+02	670	23.64	15.26
5	0.127E+00	0.528E+01	0.396E+00	0.440E+02	1064	65.01	41.86

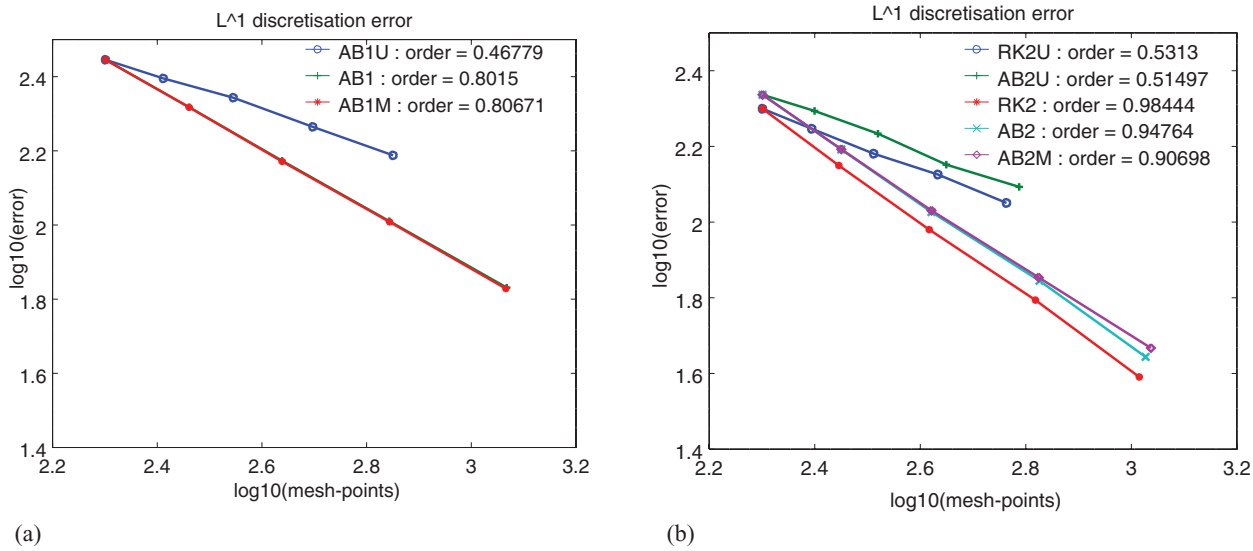

Figure 3. $\|\varepsilon - \varepsilon_{ex}\|_{l_x^1}$ with respect to the average number of cells at time $t = 0.038$: (a) first-order and (b) second-order schemes.

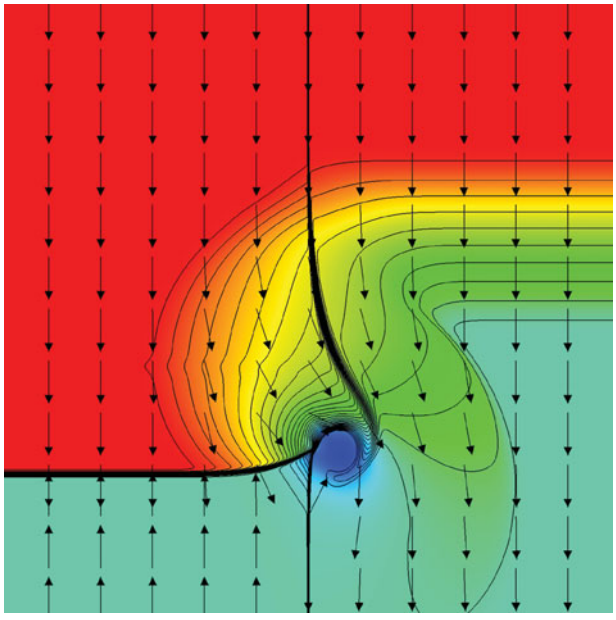

Figure 4. Reference solution for the 2D Riemann problem at time $t = 0.3$: pressure (colour), density (contours; 0.53–1.98 step 0.05), velocity field (arrows).

$t = 0.3$ with pressure (from 0.53 to 1.98 by 0.05 steps) and velocity field in Figure 4, density in Figure 5(a), and numerical density of entropy production in Figure 5(b). One notes that the shock and the contacts are associated with strong production of numerical density of entropy, which again demonstrates the relevancy of such numerical quantity to describe the most tricky regions of the computational domain.

For the following numerical experiments, we have used the following parameters:

CFL	: 0.5
Simulation time (s)	: 0.3
Initial number of blocks	: 30×30
$n_x = n_y$	: 1
Number of domain	: 1
Maximum level of mesh refinement	: $L_{\max}$
Mesh refinement parameter $\alpha_{\max}$	: 0.08
Mesh coarsening parameter $\alpha_{\min}$	: 0.05
Mesh refinement parameter $\bar{S}$	: $\frac{1}{\lvert\Omega\rvert}\sum_{k_b} S_{k_b}^n$

Convergence study is performed here with the first-order scheme only for AB1, AB1U, and AB1M with the discrete l_x^1-norm for density and pressure. Computational time is provided keeping in mind that model performance can be easily enhanced by the use of multi-domains as proposed in the next numerical experiment. The error between considered and reference solutions is performed by projecting the former on the fine grid of reference solution.

For the AB1 and AB1M schemes, four tests have been carried out by varying the level of mesh refinement from 1 to 4. As quoted before, the errors between the two schemes being similar, we only plot the pointwise pressure error between reference and AB1 case (see Figure 6). Figure 7 shows pressure contours and mesh refinement. As expected, increasing the level of mesh refinement based on the numerical density of entropy production leads to a better description of the most difficult part of the problem: shock, rarefaction, and contact regions, for instance. In comparison with other results (see, for

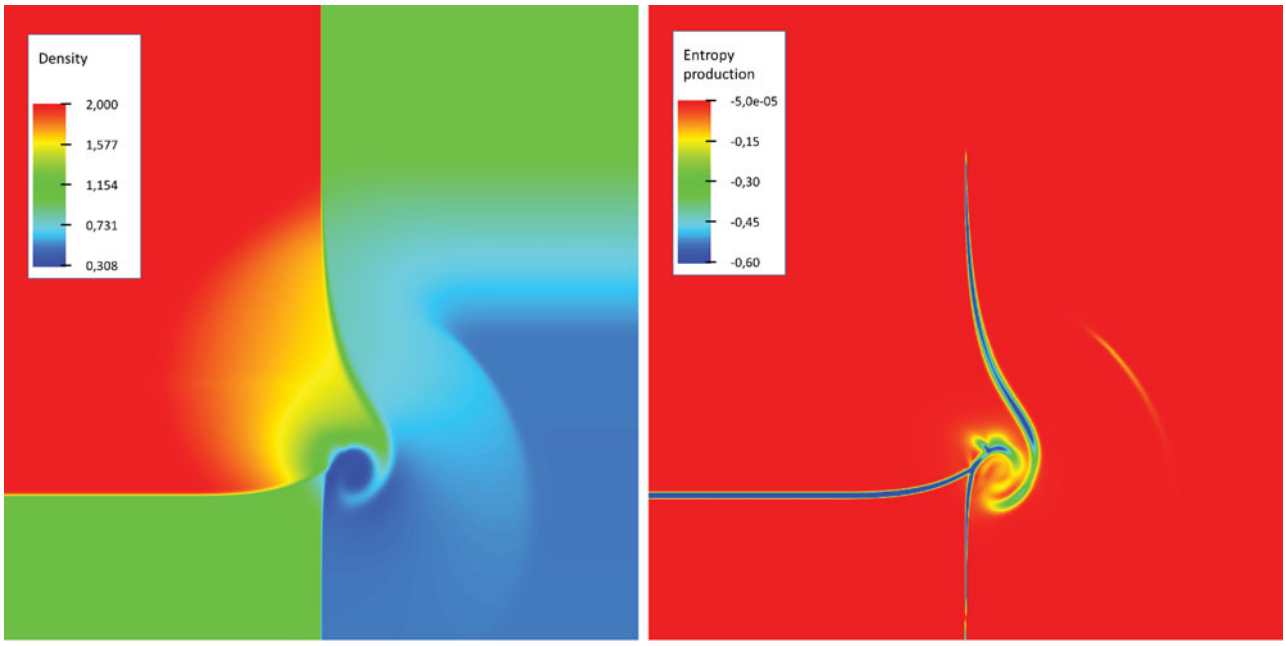

Figure 5. Reference solution for the 2D Riemann problem at time $t = 0.3$ and numerical density of entropy production: (a) density (colour); (b) numerical density of entropy production.

instance, Liska and Wendroff, 2003) the contact is precisely characterised, without any spurious vorticity and one observes the ripple formed in the NW quadrant.

Figure 8(a) and 8(b) compares the errors with the discrete l_x^1-norm for density and pressure for the AB1, AB1M, and AB1U schemes. As expected in the presence of contact discontinuity, the order of convergence is of order 1/2 for all schemes.

3.2. Compressible low Mach two-phase flows equations and interface sharpening

Let us consider now a compressible two-fluid flows problem where viscosity, surface tension, and heat conduction are neglected. The incompressibility condition is relaxed using a low Mach approach in order to lead to an hyperbolic system of conservation laws. Thus, based on Golay and Helluy (2007), the following 3D isothermal hyperbolic and compressible Euler equation system is applied to a mixture fluid of air and water:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \\ \dfrac{\partial (\rho \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = \rho \boldsymbol{g}, \end{cases} \tag{8}$$

where the unknowns are the density ρ, the three components of the velocity $\boldsymbol{u} = (u, v, w)$, and the pressure p. Here, $\boldsymbol{g}$ stands for gravitational acceleration.

Air and water fractions within the mixture are defined by the volume fraction function $\varphi \in [0, 1]$ ($\varphi = 0$ in the water, and $\varphi = 1$ in the air). With this definition of φ, the pressure of the two-phase flow problem is a function of the density ρ and the volume fraction φ, where φ solves the following non-conservative transport equation:

$$\frac{\partial \varphi}{\partial t} + \boldsymbol{u} \cdot \nabla \varphi = 0. \tag{9}$$

It is usually admitted that a flow is incompressible if the Mach number $M = \|\boldsymbol{u}\| / c$ is lower than 0.1 (c is the sound speed), keeping in mind that the real (physical) Mach number is generally much smaller (of the order of 1/400 $\sim$ 1/1600). In particular, this is constraining for explicit finite volume solver in which the time step Δt needs to satisfy a CFL condition. Note also that the numerical scheme efficiency is expected to decrease with the Mach number. Therefore, an artificial pressure law (isothermal equation of state) is used to close the system:

$$p = c_0^2(\rho - (\varphi \rho_A + (1 - \varphi)\rho_W)) + p_0. \tag{10}$$

In this expression, ρ_A and ρ_W stand for air and water densities, respectively, c_0 is the artificial sound speed (defined below), and p_0 a reference pressure. For further details about the EOS choice, the reader is referred to Golay and Helluy (2007). The value of c_0 is chosen as a compromise between the limits of compressible effects, the rate of numerical diffusion and a reasonable CFL constraint. In the present context, i.e. for flow velocity of the order of 1 m/s, an 'optimised' value $c_0 = 20$ m/s is used. It is emphasised that in the boundary mixture region $0 < \varphi <$

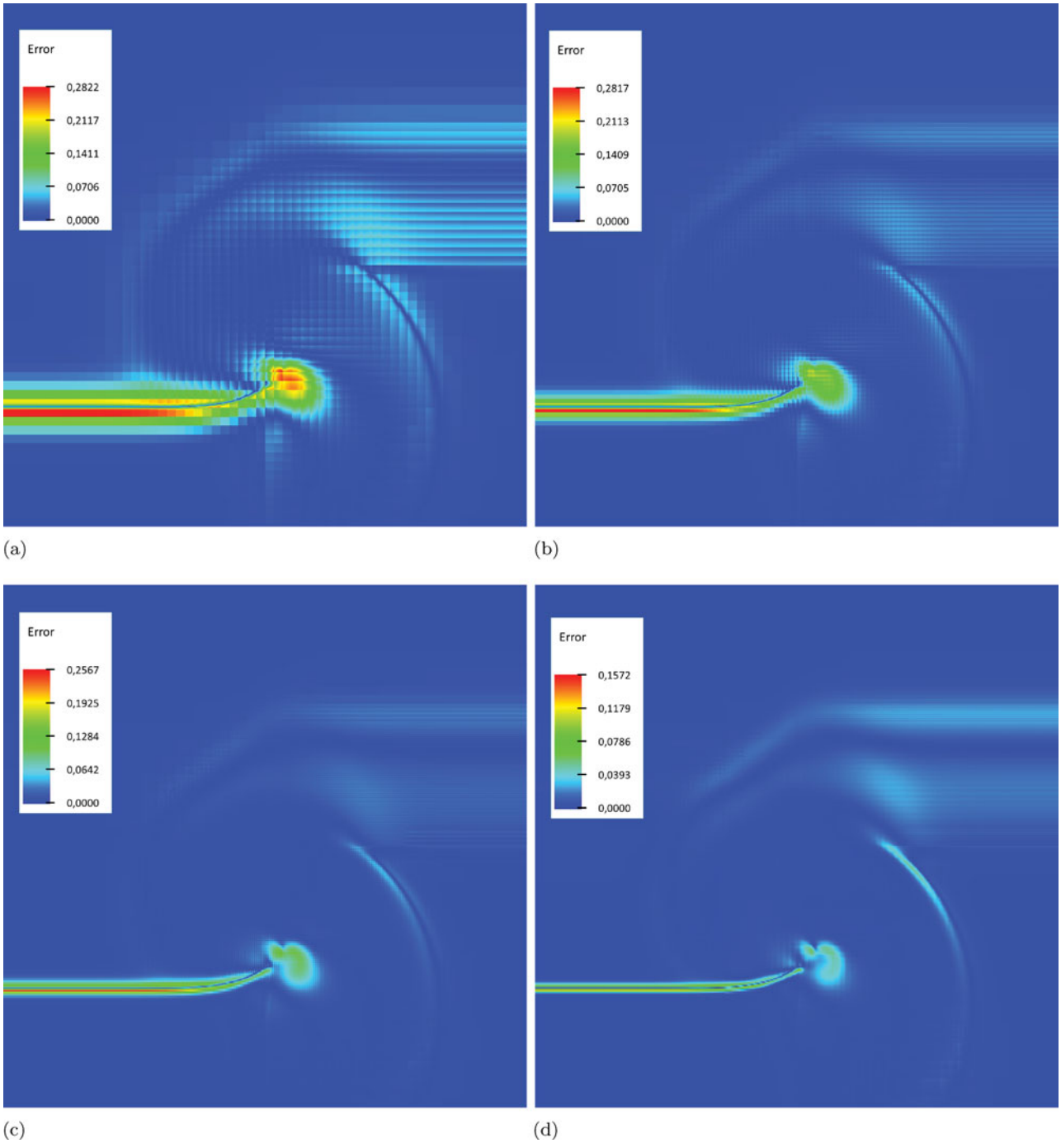

Figure 6. Pointwise pressure error between the approximate solution and the reference one at time $t = 0.3$: (a) $L_{max} = 1$. (b) $L_{max} = 2$. (c) $L_{max} = 3$. (d) $L_{max} = 4$.

1 related to numerical diffusion processes, the proposed pressure law has no physical meanings.

In the present case, for the two-fluid model, the expression of entropy and entropy flux in (2) are

$$s = \frac{1}{2}\rho \boldsymbol{u}^2 + c_0^2 \rho \ln \rho - c_0^2(\rho_W - \rho_A)\varphi,$$
$$\boldsymbol{\psi} = \left(\frac{1}{2}\rho \boldsymbol{u}^2 + c_0^2 \rho(\ln \rho + 1)\right)\boldsymbol{u}\,.$$

In the region, $0 < \varphi < 1$ where both phases coexist, the numerical diffusion is expected to deteriorate the description of the air–water interface. Following Kokh (2001) and Shyue (2014), an interface sharpening method is applied using a source term $S_c = \phi^2(1-\phi)^2(\phi - c)$ in the transport equation (9). The constant c is defined as a mass conservation parameter (Kačeniauskas, 2008) leading to $\int_\Omega S_c \equiv 0$. For each time step, a fractional step method is used to solve Equation (8). The non-homogeneous system is then solved with the interface sharpening source

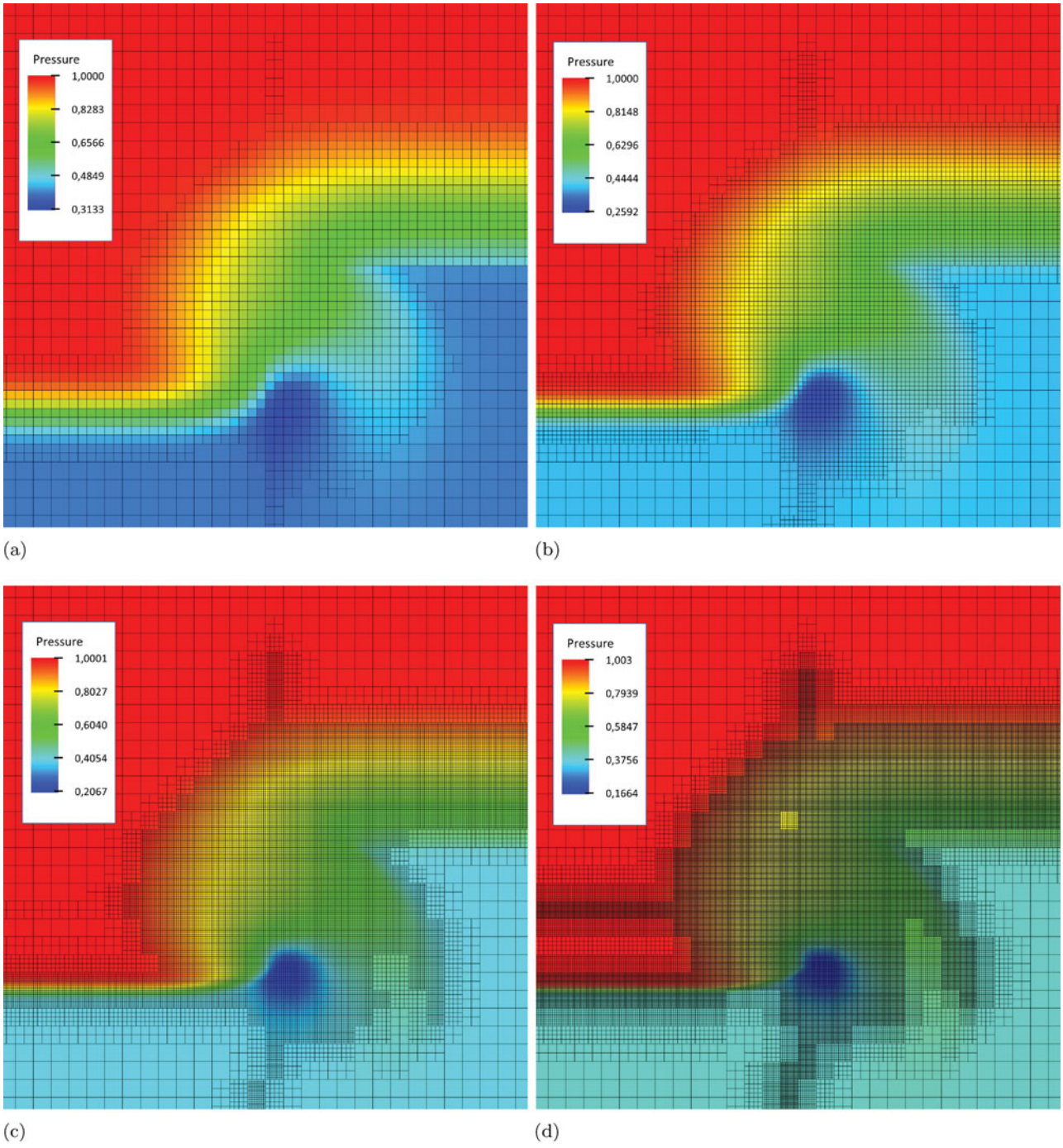

Figure 7. Pressure and the mesh at time $t = 0.3$: (a) $L_{max} = 1$. (b) $L_{max} = 2$. (c) $L_{max} = 3$. (d) $L_{max} = 4$.

terms (with a 1-iteration explicit scheme):

$$\begin{aligned} \frac{\partial \varphi}{\partial \tau} &= S_c \\ \frac{\partial \rho}{\partial \tau} &= S_c \left(\rho_A - \rho_W\right), \\ \frac{\partial \rho \boldsymbol{u}}{\partial \tau} &= S_c \boldsymbol{u} \left(\rho_A - \rho_W\right), \end{aligned} \tag{11}$$

where τ is a fictive time.

3.2.1. A 2D dam-break problem

The numerical model is confronted with the classical experiment of Martin and Moyce (1952). As shown in Figure 9, a column of water ($a \times 2a$) collapses in a box ($4a \times 3a$). The initial mesh is composed of 594 blocks ($27 \times 22 \times 1$) which are initially split into $n_x \times n_y \times n_z$ cells with $n_x = n_y = 2$ and $n_z = 1$. The mesh is refined around the air–water interface with a level 3 (i.e. 512 cells per block) with a total initial number of cells around 10,500 cells (since the level between blocks cannot exceed 2). Blocks

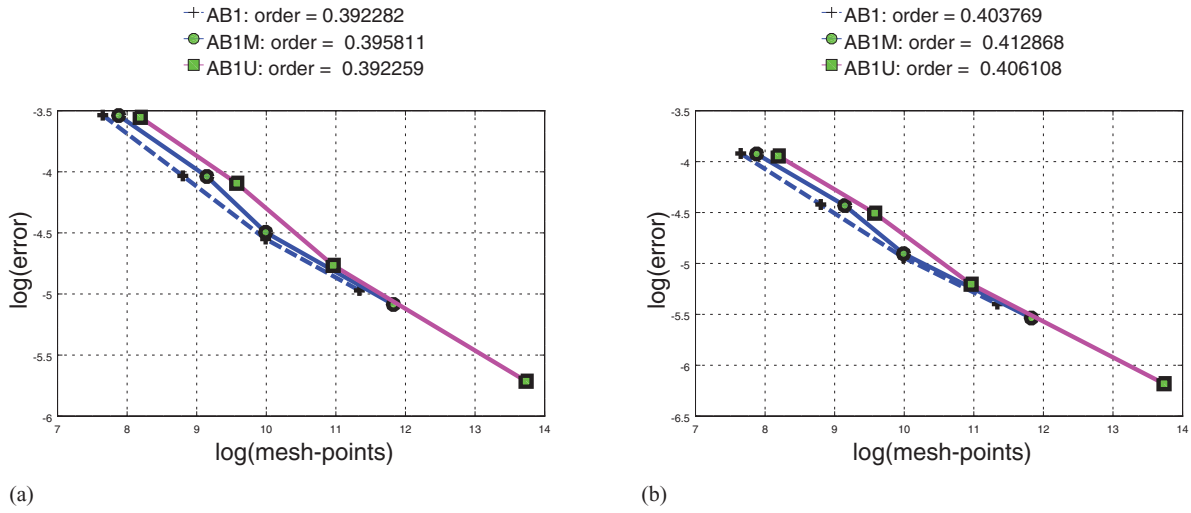

Figure 8. Error analysis for the density and the pressure: number of cells vs. l_x^1-error in log scale: (a) density; (b) pressure.

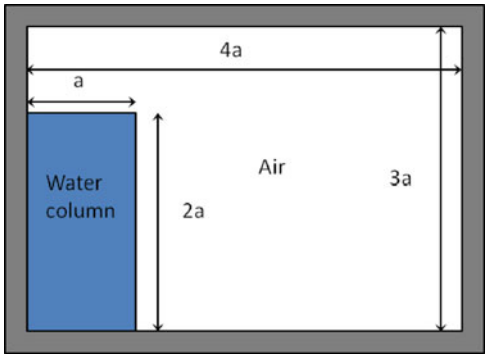

Figure 9. Collapse of water column (Koshizuka, Tamako, and Oka, 1995; Martin and Moyce, 1952).

are then distributed on 10 domains which evolve during each re-meshing time step (see Figure 10). Symmetry boundary conditions are imposed. The simulation time is $T = 0.4s$ and the AMR time step is fixed to $0.01s$, i.e. the re-meshing occurs 40 times during the simulation with the mesh refinement and coarsening parameters, respectively, set to $\alpha_{\max} = 0.2$ and $\alpha_{\min} = 0.02$. During the global simulation, the number of cells evolves from 7500 to 17, 500 cells as shown in Figure 11. Note that, even if the computation has been initially started with a very large number of cells, the mesh would have been quickly coarsened and automatically adapted to the production of the numerical density of entropy as displayed in Figures 12 and 13.

Figure 14 shows the rapid collapse of the water column and the impact on the right wall after $t = 0.3$ s. The interface sharpening method together with the dynamic mesh refinement shown in Figure 12, thanks to the entropy production (Figure 13), allows an accurate description of the interface. A very good quantitative agreement in obtained with the experimental data of Martin and Moyce (1952), as shown in Figure 15 in non-dimensional data, where the length of water l denotes the position of the air–water interface at the bottom of the box.

3.2.2. A 3D dam-break problem

A 3D dam-break problem is here used as a benchmark test to evaluate the method speed-up potential in three dimensions.

We consider a water column $\left(\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2}\right)$ collapsing in a unit cube. The initial mesh is composed of 3375 blocks $(15 \times 15 \times 15)$ which are initially split into $n_x \times n_y \times n_z$ cells with $n_x = n_y = n_z = 1$. As done in previous test case, the blocks around the air–water interface are of level 3.

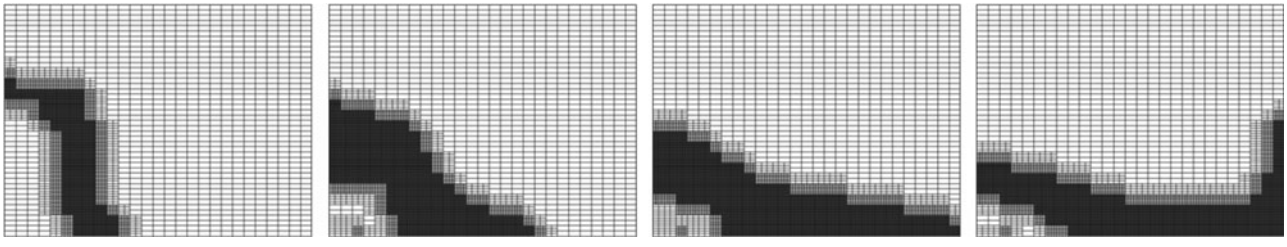

Figure 10. Block distribution in 10 domains at time 0.1, 0.2, 0.3, and 0.4 s.

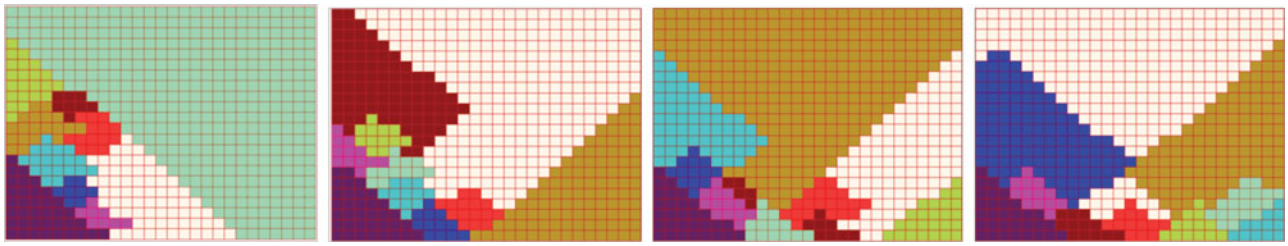

Figure 11. Number of cells during the computation.

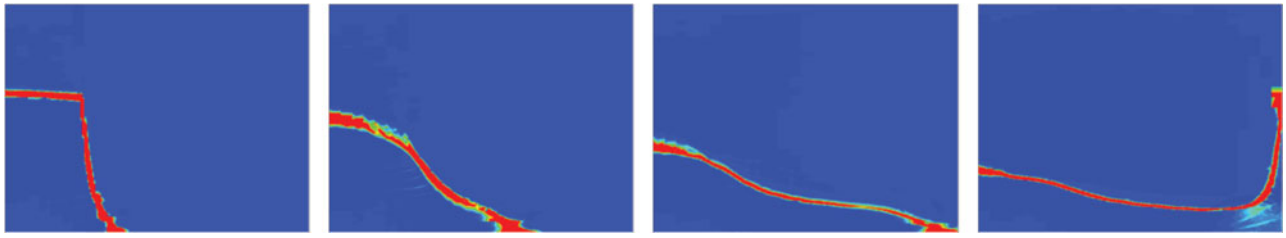

Figure 12. Mesh at time 0.1, 0.2, 0.3, and 0.4 s.

Figure 13. Numerical production of entropy at time 0.1, 0.2, 0.3, and 0.4 s (blue, zero; red, negative values).

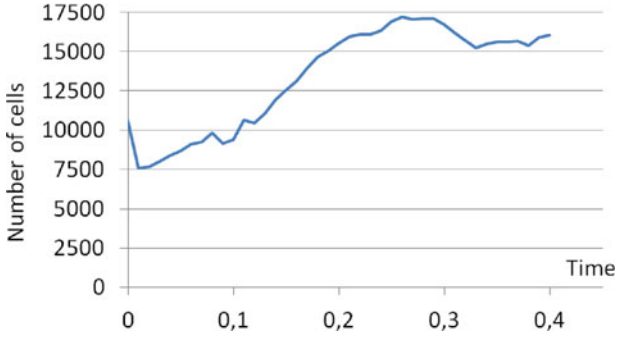

Figure 14. Density at time 0.1, 0.2, 0.3, and 0.4 s (blue, air; red, water).

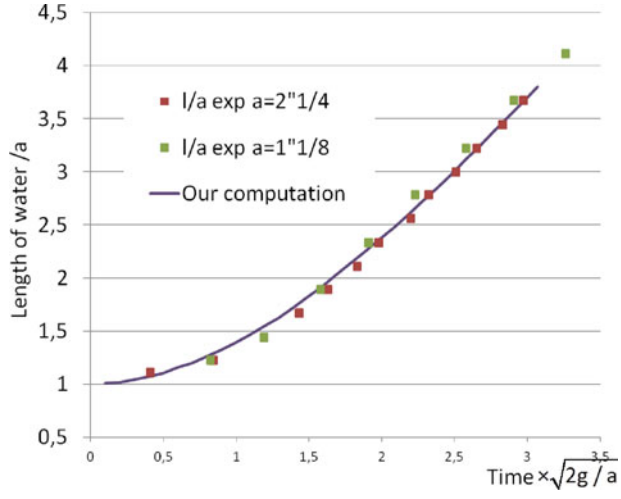

Figure 15. Length of water during the collapsing. Confrontation between the computation and the experiment of Martin and Moyce (1952).

The simulation time is $T = 2.5$ and the AMR time step is fixed to 0.025, i.e. the re-meshing occurs 100 times during the simulation with the mesh refinement and coarsening parameters, respectively, set to $\alpha_{\max} = 0.2$ and $\alpha_{\min} = 0.02$. During the global simulation, the number of cells evolves from 172, 215 to 587, 763. Snapshots of the simulation are shown in Figure 16. The 3D dam-break collapse is well simulated, as well as the multiple reflection against the walls.

For the sake of comparison, each simulation has been computed on the same small cluster (2 nodes, 40 cores). In order to show the efficiency of the parallel implementation, the RK2 and AB2 CPU times are compared during the first AMR time step. The normalised CPU time,

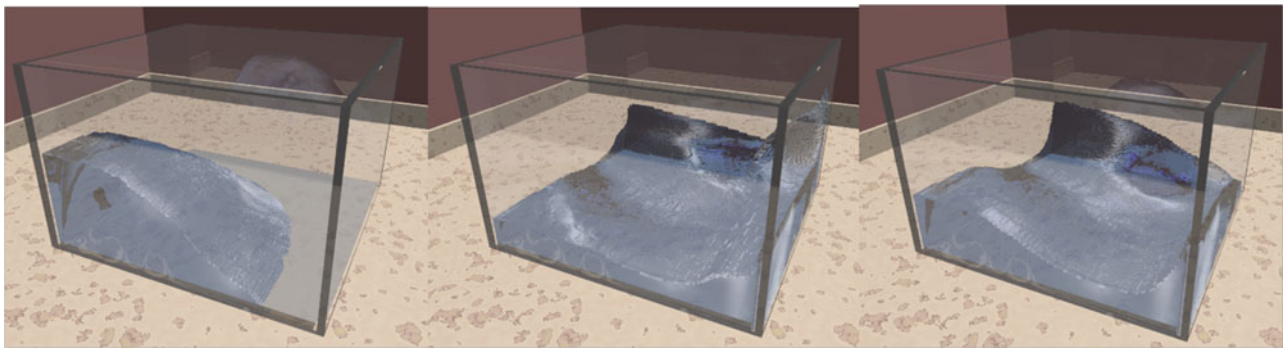

Figure 16. Collapse of 3D water column at time $t = 0.25, 0.50, 0.75$s

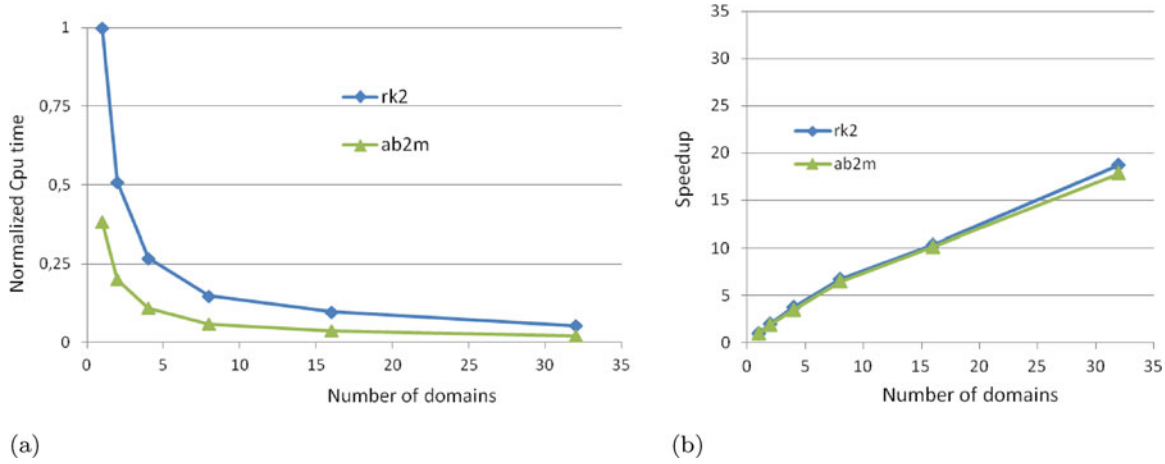

Figure 17. Normalised CPU time and speed-up versus number of domains with Runge–Kutta or Adams–Bashforth scheme: (a) normalised CPU time; (b) speed-up.

shown in Figure 17, stands for the inverse of the so-called speed-up. The computational domain is split into 1, 2, 4, 8, 16, and 32 domains.

As expected, the Adams–Bashforth scheme with local time stepping allows a great improvement of the CPU time (about a factor of 2.5 faster), because we spare the computation of an intermediate time step and some computations of fluxes on coarsed cells. The efficiency, i.e. $\frac{\text{speed-up}}{\text{number of processors}}$, of the computation is roughly 85% for 8 domains and 60% for 32 domains. From a numerical point of view, the presented method is robust although the efficiency of the parallel process reaches a steady state after 20 processors, indicating that the parallel procedure has to be optimised.

4. Conclusion

This paper reports on the extension of the 1D scheme presented by Ersoy, Golay, and Yushchenko (2013) to a general multidimensional framework. A finite volume solver is used to solve the nonlinear hyperbolic equation system on unstructured meshes. An AMR is introduced to improve both solution accuracy and CPU performance. It is based on a useful numerical criterion: the numerical density of entropy production. From a computational viewpoint, to make the local time-stepping method more efficient in a parallel context, a new BB-AMR is applied.

A series of 1D, 2D, and 3D test cases have been performed to test and validate our approach, using several model configurations (uniform or adaptive grids, Adams–Bashforth or Runge–Kutta schemes, etc.). 1D and 2D Riemann problems demonstrated that the adaptive grid strategy using the numerical density of entropy production helps to greatly improve the accuracy and reduce the computational effort. Dam-break benchmarks have also been carried out, both in 2D and 3D, to test our method when applied to compressible low Mach two-phase flows problems. Interface sharpening techniques are used here to improve the description of the complex wavy motion of the free surface during the collapse of water columns and impacts on rigid walls. Very good agreement is obtained with the existing experimental results. Comparison between Adams–Bashforth and Runge–Kutta shows the significant computation speed-up provided by the former scheme.

Further on-going developments concern, in particular, the optimisation of the proposed numerical scheme in order to increase the efficiency (i.e. the ratio of the speed-up over the number of process) as pointed out in the last 3D test case.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work is supported by the ModTerCom project within the APEX program of the region Provence-Alpe-Côte d'Azur.

References

Allahviranloo, T., N. Ahmady, and E. Ahmady. 2007. "Numerical Solution of Fuzzy Differential Equations by Predictor–Corrector Method." *Information Sciences* 177: 1633–1647.

Altmann, C., T. Belat, M. Gutnic, P. Helluy, H. Mathis, E. Sonnendrücker, W. Angulo, and J. Hérard. 2009. "A Local Time-Stepping Discontinuous Galerkin Algorithm for the MHD System." In *CEMRACS 2008 – Modelling and Numerical Simulation of Complex Fluids, 33–54. Vol. 28 of ESAIM Proceedings*. Les Ulis: EDP Sciences. Editors-in-Chief: Djalil Chafaï, Pauline Lafitte, Tony Lelièvre and Clément Mouhot http://dx.doi.org/10.1051/proc/2009038

Berger, M., and P. Colella. 1989. "Local Adaptive Mesh Refinement for Shock Hydrodynamics." *Journal of Computational Physics* 82: 64–84.

Berger, M., and J. Oliger. 1984. "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations." *Journal of Computational Physics* 53: 484–512.

Coupez, T., and E. Hachem. 2013. "Solution of High-Reynolds Incompressible Flow with Stabilized Finite Element and Adaptive Anisotropic Meshing." *Computer Methods in Applied Mechanics and Engineering* 267: 65–85.

Ersoy, M., F. Golay, and L. Yushchenko. 2013. "Adaptive Multiscale Scheme Based on Numerical Density of Entropy Production for Conservation Laws." *Central European Journal of Mathematics* 11: 1392–1415.

Eymard, R., T. Gallouet, and R. Herbin. 2000. "Finite Volume Methods." In *Handbook of Numerical Analysis*, 713–1020. Vol. VII of *Handbook of Numerical Analysis*, P.G. Ciarlet, J.L. Lions eds, vol 7, pp. 713–1020. Amsterdam: North-Holland.

Fuster, D., G. Agbaglah, C. Josserand, S. Popinet, and S. Zaleski. 2009. "Numerical Simulation of Droplets, Bubbles and Waves: State of the Art." *Fluid Dynamics Research* 41: 065001.

Godlewski, E., and P. Raviart. 1996. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Vol. 118 of *Applied Mathematical Sciences*. New York: Springer-Verlag.

Golay, F., M. Ersoy, L. Yushchenko, and D. Sous. 2015. "Block-Based Adaptive Mesh Refinement Scheme Using Numerical Density of Entropy Production for Three-Dimensional Two-Fluid Flows." *International Journal of Computational Fluid Dynamics* 29: 67–81.

Golay, F., and P. Helluy. 2007. "Numerical Schemes for Low Mach Wave Breaking." *International Journal of Computational Fluid Dynamics* 21(2): 69–86.

Hachem, E., S. Feghali, R. Codina, and T. Coupez. 2013. "Immersed Stress Method for Fluid Structure Interaction Using Anisotropic Mesh Adaptation." *International Journal for Numerical Methods in Engineering* 94: 805–825.

Hairer, E., S.P. Nørsett, and G. Wanner. 1993. *Solving Ordinary Differential Equations*. I. Vol. 8 of *Springer Series in Computational Mathematics*. 2nd ed. Berlin: Springer-Verlag.

Houston, P., J. Mackenzie, E. Süli, and G. Warnecke. 1999. "A Posteriori Error Analysis for Numerical Approximations of Friedrichs Systems." *Numerical Mathematics* 82: 433–470.

Kačeniauskas, A. 2008. "Development of Efficient Interface Sharpening Procedure for Viscous Incompressible Flows" *Informatica* 19: 487–504.

Karni, S., and A. Kurganov. 2005. "Local Error Analysis for Approximate Solutions of Hyperbolic Conservation Laws." *Advances in Computational Mathematics* 22: 79–99.

Karni, S., A. Kurganov, and G. Petrova. 2002. "A Smoothness Indicator for Adaptive Algorithms for Hyperbolic Systems." *Journal of Computational Physics* 178: 323–341.

Kokh, S. 2001. "Aspects numériques et théoriques de la modélisation des écoulements diphasiques compressibles par des méthodes de capture d'interface." PhD thesis, Université de Paris VI.

Koshizuka, S., H. Tamako, and Y. Oka. 1995. "A Particle Method for Incompressible Viscous Flow with Fluid Fragmentations." *Computational Fluid Dynamics Journal* 4: 29–46.

Kurganov, A., and E. Tadmor. 2002. "Solution of Two-Dimensional Riemann Problems for Gas Dynamics Without Riemann Problem Solvers." *Numerical Methods for Partial Differential Equations* 18: 584–608.

Lax, P.D., and X.-D. Liu. 1998. "Solution of Two-Dimensional Riemann Problems of Gas Dynamics by Positive Schemes." *SIAM Journal on Scientific Computing* 19: 319–340.

Liska, R., and B. Wendroff. 2003. "Comparison of Several Difference Schemes on 1D and 2D Test Problems for the Euler Equations." *SIAM Journal on Scientific Computing* 25: 995–1017.

Losasso, F., F. Gibou, and R. Fedkiw. 2004. "Simulating Water and Smoke with an Octree Data Structure." *ACM Transactions on Graphics* 23: 457–462.

Martin, J., and W. Moyce. 1952. "Part iv. an Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane." *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences* 244: 312–324.

Min, C., and F. Gibou. 2007. "A Second Order Accurate Level Set Method on Non-graded Adaptive Cartesian Grids." *Journal of Computational Physics* 225: 300–321.

Sambe, A.N., D. Sous, F. Golay, P. Fraunié, and R. Marcer. 2011. "Numerical Wave Breaking with Macro-roughness." *European Journal of Mechanics - B/Fluids* 30: 577–588.

Schulz-Rinne, C.W., J.P. Collins, and H.M. Glaz. 1993. "Numerical Solution of the Riemann Problem for Two-dimensional Gas Dynamics." *SIAM Journal on Scientific Computing* 14: 1394–1414.

Shyue, K.-M. 2014. "An Eulerian Interface-sharpening Algorithm for Compressible Gas Dynamics." In *Modeling, Simulation and Optimization of Complex*

Processes-HPSC 2012, 221–231. Springer. Editors: Bock, H.G., Hoang, X.P., Rannacher, R., Schlöder, J. (Eds.) http://www.springer.com/us/book/9783319090627

Toro, E.F. 2009. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. 3rd ed. Berlin: Springer-Verlag.

Tu, T., D.R. O'Hallaron, and O. Ghattas. 2005. "Scalable Parallel Octree Meshing for Terascale Applications. In *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, 4–4. doi: 10.1109/SC.2005.61 SC|05 12–18 november 2005 Seattle, http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=10435#TitleHist

Williamschen, M.J., and C. Groth. 2013. "Parallel Anisotropic Block-based Adaptive Mesh Refinement Algorithm for Three-dimensional Flows." In *21st AIAA Computational Fluid Dynamics Conference*, 1–22, June 24-27, San Diego, CA.

Woodward, P.R. 1982. "Trade-Offs in Designing Explicit Hydrodynamical Schemes for Vector Computers." In *Parallel Computations*, 153–171. Vol. 1 of *Computation Techniques*. Orlando, FL: Academic Press.

Woodward, P., and P. Colella. 1984. "The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks." *Journal of Computational Physics* 54: 115–173.

Yiu, K., D. Greaves, S. Cruz, A. Saalehi, and A. Borthwick. 1996. "Quadtree Grid Generation: Information Handling, Boundary Fitting and CFD Applications." *Computers & Fluids* 25: 759–769.

Zhang, M., and W. Wu. 2011. "A Two Dimensional Hydrodynamic and Sediment Transport Model for Dam Break Based on Finite Volume Method with Quadtree Grid." *Applied Ocean Research* 33: 297–308.

Zhang, T., and Y.X. Zheng. 1990. "Conjecture on the Structure of Solutions of the Riemann Problem for Two-Dimensional Gas Dynamics Systems." *SIAM Journal on Mathematical Analysis* 21: 593–630.

Zheng, Z., and C. Groth. 2012. "Block-Based Adaptive Mesh Refinement Finite-Volume Scheme for Hybrid Multiblock Meshes." In *7st Conference on Computational Fluid Dynamics (ICCFD7), July 9–13, Hawaii*, 1–19. http://www.iccfd.org/iccfd7/assets/pdf/papers/ICCFD7-2504_paper.pdf