OpenCL simulations of two-fluid compressible flows with a random choice method

Philippe HELLUY[†]
[†]*IRMA, Université de Strasbourg and Inria TONUS*
helluy@math.unistra.fr

Jonathan JUNG[⋆]
[⋆]*IRMA, Université de Strasbourg*
jonathan.jung@math.unistra.fr

Abstract

In this paper, we propose a new very simple numerical method for solving liquid-gas compressible flows. Such flows are difficult to simulate because classic conservative finite volume schemes generate pressure oscillations at the liquid-gas interface. We extend to several dimensions the random choice scheme that we have constructed in [2]. The extension is performed through Strang dimensional splitting. The resulting scheme exhibits interesting conservation and stability properties. For achieving high performance, the scheme is tested on recent multi-core processors and GPUs, using the OpenCL environment.

Key words : OpenCL, GPU, two-fluid compressible flow, Lagrange-projection, Glimm, Strang splitting

1 Introduction

Compressible two-fluid flows are difficult to numerically simulate. Indeed, as first discovered in [1] and [24], classic conservative finite volume schemes do not preserve the velocity and pressure equilibrium at the two-fluid interface. This leads to oscillations, lack of precision and even, in some liquid-gas simulations, to the crash of the computation.

Several cures have been proposed to obtain better schemes. Among many works, we can cite [1, 24]. The resulting schemes are generally not conservative. Based on previous works of Chalons and Goatin [11] and Chalons and Coquel [9, 10], we have

proposed in [2] a Lagrange and remap scheme for solving compressible liquid-gas flows. The remap step of the scheme is based on the Glimm random choice method at the interface. In [2], we have tested the random scheme on one-dimensional test cases. The random scheme presents interesting properties: (statistical) conservation, better precision, it preserves the pressure and velocity equilibrium at the interface, it allows to perform computations that are not feasible with other classic schemes.

In this paper, we extend the method to two-dimensional equations, thanks to Strang dimensional splitting. In [14], Strang dimensional splitting is applied for solving the gas dynamics equations with the original Glimm scheme. It appears that, because of non-linear effects, the resulting scheme of [14] does not converge towards the right solution. In our method, we observe a much better precision, because the Glimm approach is applied only at the two-fluid interface, which corresponds to a linearly degenerated field.

In order to gain efficiency, we also replace the exact Riemann solver of [2] by an approximate Riemann solver constructed from an extended relaxation system. We adapt ideas presented in [8] in order to construct a simple and robust Riemann solver that handles vacuum.

The simplicity of the whole approach allows also an easy implementation of the method on recent multi-core processors and Graphic Processing Units (GPU). For this, we use the OpenCL programming environment.

We then perform several numerical experiments for evaluating the advantages and drawbacks of the random scheme.

2 Mathematical model

In this paper, we are interested in the numerical resolution of the following system of partial differential equations, modeling a liquid-gas compressible flow

$$\partial_t W + \partial_x F(W) + \partial_y G(W) = 0, \tag{1}$$

where

$$W = (\rho, \rho u, \rho v, \rho E, \rho\varphi)^T,$$

and

$$F(W) = (\rho u, \rho u^2 + p, \rho uv, (\rho E + p)u, \rho u\varphi)^T,$$
$$G(W) = (\rho v, \rho uv, \rho v^2 + p, (\rho E + p)v, \rho v\varphi)^T.$$

The unknowns are the density ρ, the two components of the velocity u, v, the internal energy e and the mass fraction of gas φ. The unknowns depend on the space variables x, y and on the time variable t. The total energy E is the sum of the internal energy and the kinetic energy

$$E = e + \frac{u^2 + v^2}{2}.$$

The pressure p of the two-fluid medium is a function of the other thermodynamical parameters

$$p = p(\rho, e, \varphi).$$

In this paper, we consider the stiffened gas pressure law

$$p(\rho, e, \varphi) = (\gamma(\varphi) - 1)\rho e - \gamma(\varphi)p_\infty(\varphi), \tag{2}$$

where γ and p_∞ are given functions of the mass fraction φ, and

$$\gamma(\varphi) > 1.$$

At the initial time, the mass fraction $\varphi(x, y, 0) = 1$ if the point (x, y) is in the gas region and $\varphi(x, y, 0) = 0$ if the point (x, y) is in the liquid region. The mass fraction is also solution of the transport equation

$$\partial_t\varphi + u\partial_x\varphi + v\partial_y\varphi = 0,$$

which implies that for any time $t > 0$, $\varphi(x, y, t)$ can take only the two values 0 or 1.

However, classic numerical schemes generally produce an artificial diffusion of the mass fraction, and in the numerical approximation we may observe $1 > \varphi > 0$. We have thus to interpolate the pressure law parameters in the mixture. In [31], for instance, the interpolation is defined by

$$\frac{1}{\gamma(\varphi) - 1} = \varphi\frac{1}{\gamma_2 - 1} + (1 - \varphi)\frac{1}{\gamma_1 - 1}, \tag{3}$$

$$\frac{\gamma(\varphi)p_\infty(\varphi)}{\gamma(\varphi) - 1} = \varphi\frac{\gamma_2 p_{\infty,2}}{\gamma_2 - 1} + (1 - \varphi)\frac{\gamma_1 p_{\infty,1}}{\gamma_1 - 1}, \tag{4}$$

where $(\gamma_1, p_{\infty,1})$ and $(\gamma_2, p_{\infty,2})$ corresponds respectively to the pressure law parameters of the pure liquid phase $\varphi = 0$ and the pure gas phase $\varphi = 1$.

If ν is a unit vector in $\mathbb{R}^2$, we define the flux $\mathcal{F}$ of system (1)

$$\mathcal{F}(W, \nu) := \begin{pmatrix} F(W) \\ G(W) \end{pmatrix} \cdot \nu,$$

and the phase space

$$\Omega = \Big\{ W = (\rho, \rho u, \rho v, \rho E, \rho\varphi) \in \mathbb{R}^5,\ \rho > 0,\ \varphi \in [0; 1],\\ p(\rho, E - \frac{u^2 + v^2}{2}, \varphi) + p_\infty(\varphi) > 0 \Big\}.$$

If $W \in \Omega$, we can define the speed of sound

$$c(\rho, e, \varphi) : = \sqrt{\gamma(\varphi)\frac{p(\rho, e, \varphi) + p_\infty(\varphi)}{\rho}}. \tag{5}$$

For all state W in the phase space Ω and for all unit vector ν the matrix

$$\mathcal{F}'(W, \nu) = \begin{pmatrix} F'(W) \\ G'(W) \end{pmatrix} \cdot \nu$$

is diagonalizable with the following eigenvalues

$$\lambda_1 = \begin{pmatrix} u \\ v \end{pmatrix} \cdot \nu - c,\ \lambda_2 = \lambda_3 = \lambda_4 = \begin{pmatrix} u \\ v \end{pmatrix} \cdot \nu,\ \lambda_5 = \begin{pmatrix} u \\ v \end{pmatrix} \cdot \nu + c \tag{6}$$

and the system (1) is thus hyperbolic (see [35, 20]).

Remark 1 The hyperbolicity set Ω is generally not convex when $p_{\infty,1} \neq p_{\infty,2}$. However, for a fixed value of the mass fraction $\varphi_0 \in [0;1]$, the following set is always convex:

$$\Omega_{\{\varphi_0\}} := \Big\{ W = (\rho, \rho u, \rho v, \rho E, \rho\varphi) \in \mathbb{R}^5,\ \rho > 0,\ \varphi = \varphi_0,$$
$$p\left(\rho, E - \frac{u^2+v^2}{2}, \varphi\right) + p_\infty(\varphi) > 0 \Big\}.$$

Remark 2 For any state in the domain Ω, the corresponding internal energy $e > 0$. However negative pressures are possible. This can be physically justified (see [18], [4]).

The solution of (1)-(2) is generally not unique. In order to recover uniqueness, we also have to apply an entropy growth criterion, which we do not detail here (see [35, 20]).

3 Random choice numerical method

3.1 Directional splitting

We consider a increasing sequence of times t_n, $n \in \mathbb{N}^*$ and an approximation $W^n(x,y)$ of $W(x,y,t_n)$. For constructing W^{n+1} we use the Strang dimensional splitting strategy [33]. It amounts to solving, for a time step Δt, the following Cauchy problem

$$\partial_t W + \partial_x F(W) = 0, \tag{7}$$

$$W(x,y,0) = W^n(x,y). \tag{8}$$

We obtain in this way a solution $W(x,y,\Delta t)$ at time Δt. Then we solve

$$\partial_t V + \partial_y G(V) = 0, \tag{9}$$

with the initial condition

$$V(x,y,0) = W(x,y,\Delta t),$$

and we set

$$W^{n+1}(x,y) = V(x,y,\Delta t).$$

This approximation is consistent with the initial problem (1) at order one in Δt. Simple modifications that allow to reach second order accuracy are also available [33]. In addition, in our application, thanks to the rotational invariance of the Euler equations, the equations (7) and (9) are equivalent if we simply exchange the space variables x and y and the velocity components u and v. It is thus enough to construct a scheme for solving the one-dimensional Cauchy problem

$$\partial_t W + \partial_x F(W) = 0, \tag{10}$$

$$W(x,0) = W_0(x). \tag{11}$$

We use the scheme developed in [2], which we recall now.

3.2 Random choice approach

In this section we consider the numerical scheme for solving (7) with (x,t) in $[a,b]\times\mathbb{R}^+$. We consider a sequence of times t_n, $n\in\mathbb{N}$ such that the time step $\Delta t_n := t_{n+1}-t_n>0$. We consider also a space step $h=(b-a)/N$, where N is a positive integer. We define the cell centers by $x_i=a+(i-1/2)h$, $i=0\cdots N+1$. The cells $i=0$ and $i=N+1$ are used for applying boundary conditions. The cell C_i is the interval $]x_{i-1/2},x_{i+1/2}[$ where $x_{i\pm1/2}=x_i\pm h/2$. We look for an approximation of $W(x_i,t_n)$

$$W_i^n \simeq W(x_i,t_n).$$

Our method enters the family of the so-called Lagrange-projection schemes [20]. Each time step of the scheme is made of two stages. In the first stage, the equations are solved on a moving grid. The first stage is called the Arbitrary Lagrangian Eulerian (or ALE) stage. It allows to obtain a new approximation, noted $W_i^{n+1,-}$ of W at time t_{n+1} on the moved grid. The second stage, called the projection stage, allows to return to the initial grid, *i.e.* to compute W_i^{n+1} from $W_i^{n+1,-}$.

3.2.1 Arbitrary Lagrangian Eulerian (ALE) stage

In the first stage, we allow the cell boundary $x_{i+1/2}$ to move at the velocity $\xi_{i+1/2}^n$. This velocity will be defined below. At the end of the first stage, the cell boundary is

$$x_{i+1/2}^{n+1,-} = x_{i+1/2} + \Delta t_n \xi_{i+1/2}^n.$$

Integrating the conservation law (7) on the space-time trapezoid

$$\left\{(x,t),\quad x_{i-1/2}+(t-t_n)\xi_{i-1/2}^n < x < x+(t-t_n)\xi_{i+1/2}^n,\quad t_n<t<t_{n+1}\right\}, \quad (12)$$

we obtain the following finite volume approximation

$$h_i^{n+1,-}W_i^{n+1,-} - hW_i^n + \Delta t_n\left(F_L(W_i^n,W_{i+1}^n,\xi_{i+1/2}^n) - F_R(W_{i-1}^n,W_i^n,\xi_{i-1/2}^n)\right)=0. \quad (13)$$

The new size of cell i is given by

$$h_i^{n+1,-} = x_{i+1/2}^{n+1,-} - x_{i\text{-}1/2}^{n+1,-} = h + \Delta t_n(\xi_{i+1/2}^n - \xi_{i-1/2}^n). \quad (14)$$

The Arbitrary Lagrangian Eulerian (ALE) numerical fluxes are of the form

$$F_L(W_L,W_R,\xi) := F(W_L) - \xi W_L - \int_{-\infty}^{\xi}(R(W_L,W_R,\theta)-W_L)\,d\theta, \quad (15)$$

$$F_R(W_L,W_R,\xi) := F(W_R) - \xi W_R + \int_{\xi}^{+\infty}(R(W_L,W_R,\theta)-W_R)\,d\theta, \quad (16)$$

where $R(W_L,W_R,\theta)$ is obtained by the resolution of a Riemann problem. In practice, R can be an exact or approximate Riemann solver [21] and it satisfies the conservation identity

$$F_L(W_L,W_R,\xi) = F_R(W_L,W_R,\xi). \quad (17)$$

We first describe the method for the exact Riemann solver. A more efficient approximate Riemann solver, based on a relaxation approach, is also described in Section 4.

3.2.2 Choice of the interface velocity

Several choices are possible for the interface velocity $\xi^n_{i+1/2}$. The standard Eulerian choice is

$$\xi^n_{i+1/2} = 0, \tag{18}$$

and the standard Lagrangian scheme consists in choosing

$$\xi^n_{i+1/2} = u^n_{i+1/2}, \tag{19}$$

where $u^n_{i+1/2}$ is the contact discontinuity velocity in the resolution of the Riemann problem $R(W^n_i, W^n_{i+1}, x/t)$.

Another mixed choice is to take $\xi^n_{i+1/2} = u^n_{i+1/2}$ only at the liquid-gas interface and $\xi^n_{i+1/2} = 0$ elsewhere. We explain in Section 7.1.3 why this choice is interesting. The cell boundary $i + 1/2$ corresponds to the liquid-gas interface if the following condition is satisfied

$$(\varphi^n_i - 1/2)(\varphi^n_{i+1} - 1/2) < 0,$$

because $\varphi = 0$ in the liquid and $\varphi = 1$ in the gas. In this case, the interface velocity is given by

$$\xi^n_{i+1/2} = \begin{cases} u^n_{i+1/2} & \text{if} \quad (\varphi^n_i - 1/2)(\varphi^n_{i+1} - 1/2) < 0, \\ 0 & \text{else.} \end{cases} \tag{20}$$

In the sequel we will denote by "Euler scheme" the scheme corresponding to choice (18), the "Lagrange scheme" the scheme corresponding to choice (19) and by the "ALE scheme" the scheme corresponding to choice (20).

3.2.3 Projection stage

The second stage of the time step is needed for returning to the initial mesh. We have to average on the cells C_i of the initial mesh the intermediate solution $W^{n+1,-}_i$, which is defined on the moved cells $C^{n+1,-}_i =]x^{n+1,-}_{i-1/2}, x^{n+1,-}_{i+1/2}[$. We consider an averaging process that depends on the location of the cell with respect to the material interface. The averaging is thus different if the cell touches the liquid-gas interface or not. More precisely, if the cell is not at the interface, i.e. if

$$(\varphi^n_i - 1/2)(\varphi^n_{i+1} - 1/2) > 0 \text{ and } (\varphi^n_{i-1}\text{-}1/2)(\varphi^n_i\text{-}1/2) > 0,$$

then we perform a standard averaging

$$\begin{aligned} W^{n+1}_i &= W^{n+1,-}_i - \frac{\Delta t_n}{h}(\max(\xi^n_{i-\frac{1}{2}}, 0)(W^{n+1,-}_i - W^{n+1,-}_{i-1}) \\ &+ \min(\xi^n_{i+\frac{1}{2}}, 0)(W^{n+1,-}_{i+1} - W^{n+1,-}_i)). \end{aligned} \tag{21}$$

Remark 3 If the interface velocities $\xi^n_{i\pm1/2} = 0$, in this case, we simply obtain

$$W^{n+1}_i = W^{n+1,-}_i,$$

and the scheme reduces to the classic Godunov scheme.

On the other hand, if the cell touches the interface

$$(\varphi_i^n - 1/2)(\varphi_{i+1}^n - 1/2) < 0 \text{ or } (\varphi_{i-1}^n\text{-}1/2)(\varphi_i^n\text{-}1/2)<0,$$

we then consider a random sequence $\omega_n \in [0,1[$ and we perform a random averaging

$$W_i^{n+1} = \begin{cases} W_{i-1}^{n+1,-}, \text{ if } \omega_n < \frac{\xi_{i-1/2}^n \Delta t_n}{h}, \\ W_i^{n+1,-}, \text{ if } \frac{\xi_{i-1/2}^n \Delta t_n}{h} \leq \omega_n \leq 1 + \frac{\xi_{i+1/2}^n \Delta t_n}{h}, \\ W_{i+1}^{n+1,-}, \text{ if } \omega_n > 1 + \frac{\xi_{i+1/2}^n \Delta t_n}{h}. \end{cases} \tag{22}$$

A good choice for the random sequence ω_n is the (k_1, k_2) van der Corput sequence, computed by the following C algorithm

```
float corput(int n,int k1,int k2){
  float corput=0;
  float s=1;
  while(n>0){
    s/=k1;
    corput+=(k2*n%k1)%k1*s;
    n/=k1;
  }
  return corput;
}
```

In this algorithm, k_1 and k_2 are two relatively prime numbers and $k_1 > k_2 > 0$. For more details, we refer to [35]. In practice, we consider the $(5, 3)$ van der Corput sequence.

Remark 4 It is not possible to perform the random averaging in all the cells, because the resulting scheme is generally not BV stable (see [2] and also a numerical example in Section 7.1.3).

4 Relaxation solver

4.1 Introduction

For computing the ALE numerical fluxes (15) and (16), we have to provide a Riemann solver $R(W_L, W_R, \xi)$. If R is the exact Riemann solver, the finite volume scheme (13) is the Godunov scheme (see[35]) and we have

$$\begin{aligned} F_L(W_L, W_R, \xi) &= F_R(W_L, W_R, \xi) \\ &= F\left(R(W_L, W_R, \xi)\right) - \xi R(W_L, W_R, \xi). \end{aligned}$$

We recall that $R(W_L, W_R, x/t) = W(x,t)$ is the exact entropy solution of the Riemann problem‘

$$\partial_t W + \partial_x F(W) = 0, \tag{23}$$

$$W(x,0) = \begin{cases} W_L & \text{if } x < 0, \\ W_R & \text{if } x > 0. \end{cases} \tag{24}$$

The exact solution can be computed (see [4]) but it requires solving a nonlinear equation using an iterative procedure. Numerically, this computation is perfectly feasible. However, one of our objectives is to implement our method on GPU. Because it involves many different branch tests the exact Riemann solver is not the most efficient on such computer architecture.

It is thus indicated to replace the exact Riemann solver by an approximated one. However we must construct it carefully. Indeed, other classic solvers, such as the Roe or VFRoe Riemann solvers [19, 29], even with an entropy correction, lead to crashes in the numerical simulations. The crashes occur because at the liquid-gas interface the internal energy becomes negative. It is necessary to construct a solver that preserves the energy positivity.

For constructing a positive flux, we follow the relaxation approach, which is explained now in many papers [23, 15, 7, 5]. We follow here the presentation of [13, 7, 8]. The principle of the relaxation solver is to use a larger system of PDE's (called the relaxation system), which extends (7). The relaxation system is chosen in such a way that the exact solution of the Riemann problem can be easily computed. In addition, the structure of the relaxed Riemann solver allows proving in an easier way the positivity of the approximate Riemann solver.

Definition 4.1 A relaxation system for (7) is a system of conservation laws in higher dimension $q > 5$

$$\partial_t \widetilde{W} + \partial_x \widetilde{F}(\widetilde{W}) = 0, \tag{25}$$

where $\widetilde{W}(x,t) \in \mathbb{R}^q$ and $\widetilde{F}(\widetilde{W}) \in \mathbb{R}^q$. The link between (7) and (25) is made by the assumption that we have a linear operator

$$L : \mathbb{R}^q \to \mathbb{R}^5$$

and a non-linear operator

$$M : \mathbb{R}^5 \to \mathbb{R}^q$$

such that for any $W \in \mathbb{R}^5$

$$\begin{aligned} L\left(M(W)\right) &= W, \\ L\left(\widetilde{F}\left(M(W)\right)\right) &= F(W). \end{aligned}$$

The main idea is that $W = L\left(\widetilde{W}\right)$ should be an approximate solution to (7) when $\widetilde{W}$ solves (25). The approximation is valid for a short time, corresponding to the time step of the scheme.

Remark 5 As it was introduced in [7, 8], we have to mention that we do not consider here right-hand sides in (25).

4.2 Construction of the relaxation system

A way to introduce the relaxation system is to start with a smooth solution of the system (7) and to derive an equation on the sound speed $c(\rho, e, \varphi)$ and the pressure

$p(\rho, e, \varphi)$ in order to delete the non-linearity of the system. Developing the energy equation of system (7) gives

$$\partial_t(\rho p) + \partial_x(\rho u p) + \rho^2 c^2 \partial_x u = 0.$$

In order to remove the non-linearity of this equation we replace $p(\rho, e, \varphi)$ by a new variable π and $(\rho c)^2$ by a new variable a^2, we get the following equation

$$\partial_t(\rho\pi) + \partial_x(\rho u \pi) + a^2 \partial_x u = 0.$$

We also impose that a is transported at the velocity of the fluid

$$\partial_t a + u \partial_x a = 0.$$

Finally, the relaxation system is given by

$$\begin{aligned}
\partial_t \rho + \partial_x(\rho u) &= 0, && (26)\\
\partial_t(\rho u) + \partial_x(\rho u^2 + \pi) &= 0,\\
\partial_t(\rho v) + \partial_x(\rho u v) &= 0,\\
\partial_t(\rho E) + \partial_x((\rho E + \pi) u) &= 0,\\
\partial_t(\rho\varphi) + \partial_x(\rho u \varphi) &= 0,\\
\partial_t(\frac{\rho\pi}{a^2}) + \partial_x(\frac{\rho\pi u}{a^2} + u) &= 0,\\
\partial_t(\rho a) + \partial_x(\rho u a) &= 0, && (27)
\end{aligned}$$

with

$$E = e + \frac{u^2 + v^2}{2}.$$

We can write the system (26)-(27) in the conservative form

$$\partial_t \widetilde{W} + \partial_x \widetilde{F}(\widetilde{W}) = 0, \tag{28}$$

where

$$\widetilde{W} = (\rho, \rho u, \rho v, \rho E, \rho\varphi, \frac{\rho\pi}{a^2}, \rho a)^T$$

and

$$\widetilde{F}(\widetilde{W}) = (\rho u, \rho u^2 + \pi, \rho u v, (\rho E + \pi) u, \rho u \varphi, \frac{\rho\pi u}{a^2} + u, \rho u a)^T.$$

The system (28) is a relaxation system for (7) with

$$\begin{aligned}
L:\ \mathbb{R}^7 &\to \mathbb{R}^5\\
\widetilde{W} &\mapsto \left(\widetilde{W}_1, \widetilde{W}_2, \widetilde{W}_3, \widetilde{W}_4, \widetilde{W}_5\right)
\end{aligned}$$

and

$$\begin{aligned}
M:\ \mathbb{R}^5 &\to \mathbb{R}^7\\
(\rho, \rho u, \rho v, \rho E, \rho\varphi) &\mapsto \left(\rho, \rho u, \rho v, \rho E, \rho\varphi, \frac{p}{\rho c^2}, \rho^2 c\right).
\end{aligned}$$

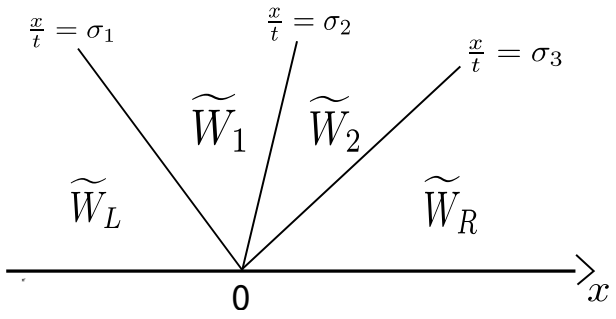


Figure 1: Structure of the solution of the relaxation system (28).

Let us note that in the relaxation system we have 7 unknowns $(\rho, u, v, E, \varphi, \pi, a)$ for our initial system of 5 unknowns (ρ, u, v, E, φ). The additional unknown π represents a relaxed pressure and a is a convected pressure law parameter. One has to take care that in (28) ρ, E, π and a are understood as independent variables.

The system (28) is hyperbolic over the following phase space

$$\widetilde{\Omega} : \quad = \quad \left\{ \widetilde{W} = (\rho, \rho u, \rho v, \rho E, \rho\varphi, \frac{\rho\pi}{a^2}, \rho a) \in \mathbb{R}^7, \ a > 0, \rho > 0 \right\}.$$

with the following increasingly ordered eigenvalues :

$$\lambda_1 = u - \frac{a}{\rho}, \ \lambda_2 = ... = \lambda_6 = u, \ \lambda_7 = u + \frac{a}{\rho}. \tag{29}$$

From the works of Liu [27], Chen, Levermore and Liu [13], it is known that for avoiding instabilities, the subcharacteristic condition has to be satisfied: the eigenvalues (29) of the relaxation system (28) and those of the original system (6) must be properly interlaced. In the present relaxation setting (28), these stability conditions are satisfied if the free coefficient $a > 0$, is larger than the exact Lagrangian sound speed

$$a > \rho c(\rho, e, \varphi). \tag{30}$$

4.3 Resolution of the relaxation system

Before giving the expression of the interface flux for the relaxation system (28) and then the flux for the initial system (7), we describe the exact resolution of the relaxed Riemann problem

$$\begin{aligned} \partial_t \widetilde{W} + \partial_x \widetilde{F}(\widetilde{W}) &= 0, \\ \widetilde{W}(x,0) &= \begin{cases} \widetilde{W}_L & \text{if} \quad x < 0, \\ \widetilde{W}_R & \text{if} \quad x > 0. \end{cases} \end{aligned}$$

Thanks to the additional relaxed variables, the eigenvalues of the relaxation system (28) are linearly degenerated. Thus we can compute easily the exact solution to the relaxed Riemann problem. It has three wave speeds $\sigma_1 = u - \frac{a}{\rho}$, $\sigma_2 = u$, $\sigma_3 = u + \frac{a}{\rho}$, with two intermediate states that we shall index by $\cdot_1$ and $\cdot_2$ (see Figure

1). We notice that because a is simply convected at the velocity u, we have $a_1 = a_L$ and $a_2 = a_R$. Then, according to the expressions of the Riemann invariants for the first and third wave and the fact that u and π are two independent Riemann invariants for the central wave, the intermediate states are obtained by

$$\begin{aligned} u_1 = u_2, \qquad & \pi_1 = \pi_2, \\ v_L = v_1, \qquad & v_2 = v_R, \\ \varphi_L = \varphi_1, \qquad & \varphi_2 = \varphi_R, \\ (\pi + ua)_L = (\pi + ua)_1, \qquad & (\pi - ua)_2 = (\pi - ua)_R, \\ \left(\frac{1}{\rho} + \frac{\pi}{a^2}\right)_L = \left(\frac{1}{\rho} + \frac{\pi}{a^2}\right)_1, \qquad & \left(\frac{1}{\rho} + \frac{\pi}{a^2}\right)_2 = \left(\frac{1}{\rho} + \frac{\pi}{a^2}\right)_R, \end{aligned} \tag{31}$$

$$\left(e - \frac{\pi^2}{2a^2}\right)_L = \left(e - \frac{\pi^2}{2a^2}\right)_1, \qquad \left(e - \frac{\pi^2}{2a^2}\right)_2 = \left(e - \frac{\pi^2}{2a^2}\right)_R. \tag{32}$$

The wave speeds are given by

$$\sigma_1 = u_L - \frac{a_L}{\rho_L},\ \sigma_2 = u_1 = u_2,\ \sigma_3 = u_R + \frac{a_R}{\rho_R}.$$

Then the exact solution of the relaxation system (28) is given by

$$\widetilde{R}(\widetilde{W}_L, \widetilde{W}_R, \xi) = \begin{cases} \widetilde{W}_L, & \text{if } \xi < \sigma_1 \\ \widetilde{W}_1, & \text{if } \sigma_1 \le \xi < \sigma_2 \\ \widetilde{W}_2, & \text{if } \sigma_2 \le \xi < \sigma_3 \\ \widetilde{W}_R, & \text{if } \sigma_3 \le \xi \end{cases} \tag{33}$$

where the states

$$\widetilde{W}_1 = (\rho_1, \rho_1 u_1, \rho_1 v_1, \rho_1 E_1, \rho_1 \varphi_1, \frac{\rho_1 \pi_1}{a_1^2}, \rho_1 a_1)^T$$

and

$$\widetilde{W}_2 = (\rho_2, \rho_2 u_2, \rho_2 v_2, \rho_2 E_2, \rho_2 \varphi_2, \frac{\rho_2 \pi_2}{a_2^2}, \rho_2 a_2)^T$$

are defined by

$$\frac{1}{\rho_1} = \frac{1}{\rho_L} + \frac{a_R(u_R - u_L) + \pi_L - \pi_R}{a_L(a_L + a_R)}, \tag{34}$$

$$\frac{1}{\rho_2} = \frac{1}{\rho_R} + \frac{a_L(u_R - u_L) + \pi_R - \pi_L}{a_R(a_L + a_R)}, \tag{35}$$

$$u_1 = u_2 = \frac{\pi_L - \pi_R + a_L u_L + a_R u_R}{a_R + a_L}, \tag{36}$$

$$\begin{aligned} v_1 &= v_L,\ v_2 = v_R, \\ \pi_1 &= \pi_2 = \frac{a_L \pi_R + a_R \pi_L + a_L a_R (u_L - u_R)}{a_L + a_R}, \end{aligned} \tag{37}$$

$$\begin{aligned} e_1 &= e_L - \frac{\pi_L^2 - \pi_1^2}{2a_L^2}, \\ e_2 &= e_R - \frac{\pi_R^2 - \pi_2^2}{2a_R^2}, \\ \varphi_1 &= \varphi_L,\ \varphi_2 = \varphi_R. \end{aligned} \tag{38}$$

Remark 6 The positivity of ρ_1 and ρ_2 is not guaranteed from (34)-(35). This is a requirement that constraints a_L and a_R to be large enough. Another requirement is that $\sigma_1 < \sigma_2 < \sigma_3$, but indeed this property follows from the previous one, since one has $\sigma_2 - \sigma_1 = \frac{a_L}{\rho_1}$ and $\sigma_3 - \sigma_2 = \frac{a_R}{\rho_2}$.

4.4 Expression of the numerical fluxes

As we have explained in section 3.2, in the first stage of the scheme, we use an ALE exact or approximated Riemann solver. It is thus necessary that the numerical flux depends on an additional velocity variable ξ. In practice, the additional velocity ξ is set to zero or to the contact discontinuity velocity. For the computations, it is convenient to introduce the left and right numerical fluxes for the relaxation system with the formula

$$\widetilde{F}_L(\widetilde{W}_L, \widetilde{W}_R, \xi) := \widetilde{F}(\widetilde{W}_L) - \xi \widetilde{W}_L - \int_{-\infty}^{\xi} \left(\widetilde{R}(\widetilde{W}_L, \widetilde{W}_R, \theta) - \widetilde{W}_L \right) d\theta, \quad (39)$$

$$\widetilde{F}_R(\widetilde{W}_L, \widetilde{W}_R, \xi) := \widetilde{F}(\widetilde{W}_R) - \xi \widetilde{W}_R + \int_{\xi}^{+\infty} \left(\widetilde{R}(\widetilde{W}_L, \widetilde{W}_R, \theta) - \widetilde{W}_R \right) d\theta. \quad (40)$$

From the numerical flux $\widetilde{F}_{L,R}(\widetilde{W}_L, \widetilde{W}_R, \xi)$ of the relaxation system (25) we then obtain the numerical flux for our original system (7) by

$$F_{L,R}(W_L, W_R, \xi) = L\left(\widetilde{F}_{L,R}\left(M(W_L), M(W_R), \xi\right) \right).$$

We also impose the conservation property (17) of the relaxation numerical flux [21]

$$\widetilde{F}_L(\widetilde{W}_L, \widetilde{W}_R, \xi) = \widetilde{F}_R(\widetilde{W}_L, \widetilde{W}_R, \xi).$$

This gives us

$$\widetilde{F}(\widetilde{W}_R) - \widetilde{F}(\widetilde{W}_L) = \sigma_1(\widetilde{W}_1 - \widetilde{W}_L) + \sigma_2(\widetilde{W}_2 - \widetilde{W}_1) + \sigma_3(\widetilde{W}_R - \widetilde{W}_2). \quad (41)$$

Then we can omit the indexes $\cdot_{L,R}$ and write the numerical flux under the form

$$\widetilde{F}(\widetilde{W}_L, \widetilde{W}_R, \xi) = \begin{cases} \widetilde{F}(\widetilde{W}_L) - \xi \widetilde{W}_L & \text{if } \xi < \sigma_1, \\ \widetilde{F}_1 - \xi \widetilde{W}_1 & \text{if } \sigma_1 \le \xi < \sigma_2, \\ \widetilde{F}_2 - \xi \widetilde{W}_2 & \text{if } \sigma_2 \le \xi < \sigma_3, \\ \widetilde{F}(\widetilde{W}_R) - \xi \widetilde{W}_R & \text{if } \sigma_3 \le \xi. \end{cases} \quad (42)$$

where $\widetilde{F}_1$ and $\widetilde{F}_2$ are defined by

$$\widetilde{F}_1 := \widetilde{F}(\widetilde{W}_L) + \sigma_1(\widetilde{W}_1 - \widetilde{W}_L), \quad (43)$$

$$\widetilde{F}_2 := \widetilde{F}(\widetilde{W}_R) + \sigma_3(\widetilde{W}_2 - \widetilde{W}_R), \quad (44)$$

$$= \widetilde{F}_1 + \sigma_2(\widetilde{W}_2 - \widetilde{W}_1). \quad (45)$$

In order to determine explicitly the flux $\widetilde{F}(\widetilde{W}_L, \widetilde{W}_R, \xi)$, we have to specify $\widetilde{F}_1$ and $\widetilde{F}_2$. As $\sigma_1 = u_L - \frac{a_L}{\rho_L} = u_1 - \frac{a_L}{\rho_1}$, the relation (43) gives

$$\begin{aligned}
\widetilde{F}_1 - u_1\widetilde{W}_1 &= \widetilde{F}(\widetilde{W}_L) - u_L\widetilde{W}_L + a_L\left(\frac{1}{\rho_L}\widetilde{W}_L - \frac{1}{\rho_1}\widetilde{W}_1\right), \\
&= (0, \pi_L, 0, \pi_L u_L, 0, u_L, 0)^T + a_L(0, u_L - u_1, 0, E_L - E_1, 0, \frac{\pi_L - \pi_1}{a_L^2}, 0)^T, \\
&= (0, \pi_L + a_L(u_L - u_1), 0, \pi_L u_L + a_L(E_L - E_1), 0, u_L + \frac{\pi_L - \pi_1}{a_L}, 0)^T,
\end{aligned}$$

with relations (31)-(32), we obtain

$$\begin{aligned}
\widetilde{F}_1 - u_1\widetilde{W}_1 &= (0, \pi_1, 0, \pi_1 u_1, 0, u_1, 0)^T, \\
\Rightarrow \widetilde{F}_1 &= \widetilde{F}(\widetilde{W}_1).
\end{aligned}$$

From (45) and as $u_1 = u_2$ and $\pi_1 = \pi_2$, we can write $\widetilde{F}_2$ as

$$\begin{aligned}
\widetilde{F}_2 &= u_2\widetilde{W}_2 + (0, \pi_2, 0, \pi_2 u_2, 0, u_2, 0)^T, \\
\Rightarrow \widetilde{F}_2 &= \widetilde{F}(\widetilde{W}_2).
\end{aligned}$$

Remark 7 It is remarkable that the intermediate numerical fluxes, which depends in the general cases on the left and right states, can be expressed as the flux of the corresponding intermediate state of the relaxed system (i.e. $\widetilde{F}(\widetilde{W}_L, \widetilde{W}_R, \xi) = \widetilde{F}(\widetilde{W}_*) - \xi\widetilde{W}_*$ for some $\widetilde{W}_* \in \mathbb{R}^7$).

In order to go back to the original system (7), we use in the formula (13) the interface flux given by

$$\begin{aligned}
F(W_L, W_R, \xi) &= L\left(\widetilde{F}\left(M(W_L), M(W_R), \xi\right)\right), \\
&= L\left(\widetilde{F}\left(\widetilde{R}\left(M(W_L), M(W_R), \xi\right)\right) - \xi\widetilde{R}\left(M(W_L), M(W_R), \xi\right)\right),
\end{aligned}$$

where $\widetilde{R}\left(M(W_L), M(W_R), \xi\right)$ is given by (33) and $\widetilde{F}$ is defined in (42).

Remark 8 Generally $\pi_1 \neq p(\rho_1, e_1, \varphi_1)$ and $\pi_2 \neq p(\rho_2, e_2, \varphi_2)$ and it is not possible to write $F(W_L, W_R, \xi) = F(W_*) - \xi W_*$ for some $W_* \in \mathbb{R}^5$.

5 Properties of the scheme

In this section, we list some properties of the scheme. As we consider dimensional splitting (see Section 3.1), we only give the properties in the one-dimensional case.

PROPOSITION 5.1 If we assume that $W_L \in \Omega_{\{\varphi_L\}}$, $W_R \in \Omega_{\{\varphi_R\}}$ and that a_L and a_R satisfy the relations

$$\text{if } p_R - p_L \geq 0, \quad \begin{cases} \frac{a_L}{\rho_L} &= c_L + \alpha \max\left(\frac{p_R - p_L}{\rho_R c_R} + u_L - u_R, 0\right), \\ \frac{a_R}{\rho_R} &= c_R + \alpha \max\left(\frac{p_L - p_R}{a_L} + u_L - u_R, 0\right), \end{cases} \tag{46}$$

$$\text{if } p_R - p_L \leq 0, \quad \begin{cases} \frac{a_R}{\rho_R} &= c_R + \alpha \max\left(\frac{p_L - p_R}{\rho_L c_L} + u_L - u_R, 0\right), \\ \frac{a_L}{\rho_L} &= c_L + \alpha \max\left(\frac{p_R - p_L}{a_R} + u_L - u_R, 0\right), \end{cases} \tag{47}$$

where $c_{L,R}$ is given by

$$c_{L,R} = c\left(\rho_{L,R}, e_{L,R}, \varphi_{L,R}\right)$$

(see the definition of c in (5)) and

$$\alpha = \max\left(\frac{\gamma(\varphi_L)+1}{2}, \frac{\gamma(\varphi_R)+1}{2}\right),$$

then we obtain the following stability properties

$$\begin{array}{ccccc} \sigma_1 = u_L - \dfrac{a_L}{\rho_L} < & \sigma_2 = u_1 = u_2 & < \sigma_3 = u_R + \dfrac{a_R}{\rho_R}, \\ L\left(\widetilde{W}_1\right) & \in & \Omega_{\{\varphi_L\}}, \\ L\left(\widetilde{W}_2\right) & \in & \Omega_{\{\varphi_R\}}. \end{array}$$

Proof We adapt the method described in [8]. Using expressions (34) and (35), it is easy to prove, under assumptions (46)-(47), that

$$\begin{cases} \rho_1 > 0, \\ \rho_2 > 0. \end{cases}$$

Proving that $e_1 - \frac{p_\infty(\varphi_L)}{\rho_1} > 0$ and $e_2 - \frac{p_\infty(\varphi_R)}{\rho_2} > 0$ requires more subtle arguments based on entropy inequalities satisfied by general relaxation solvers. See [8]. □

Remark 9 If a_L and a_R are defined with (46)-(47), the subcharacteristic condition (30) is obviously satisfied. With this choice for a_L and a_R, the solver also satisfies a discrete entropy inequality (see [7, 8]). In addition, the two relations (46)-(47) allow

$$\rho_L \to 0 \text{ or } \rho_R \to 0.$$

Thus, the solver can handle vacuum.

Proposition 5.2 The following stability property holds for the Lagrangian scheme (19) or the ALE scheme (20):

If for some n, i

- $W_i^n \in \Omega_{\{\varphi_i^n\}}$,
- $(a^n_{i\pm 1/2})_{L,R}$ satisfy conditions (46)-(47),
- Δt_n satisfies the 1/2 CFL condition

$$\begin{cases} \min\left(\xi^n_{i+1/2}, u_i^n - \frac{(a^n_{i+1/2})_L}{\rho_i^n}\right)\Delta t_n > \frac{-h}{2} & , \\ \max\left(\xi^n_{i-1/2}, u_i^n + \frac{(a^n_{i-1/2})_R}{\rho_i^n}\right)\Delta t_n < \frac{h}{2} & , \end{cases} \tag{48}$$

then

$$W_i^{n+1,-} \in \Omega_{\{\varphi_i^n\}}.$$

Proof We have

$$h_i^{n+1,-}W_i^{n+1,-} = hW_i^n - \Delta t_n \left(F_L(W_i^n, W_{i+1}^n, \xi_{i+1/2}^n) - F_R(W_{i-1}^n, W_i^n, \xi_{i-1/2}^n) \right),$$

where $F_{L,R}(W_L, W_R, \xi) = L\left(\widetilde{F}_{L,R}\left(M(W_L), M(W_R), \xi\right)\right)$. As L is a linear operator, we can write

$$W_i^{n+1,-} = L\left(\widetilde{W}_i^{n+1,-}\right)$$

where $\widetilde{W}_i^{n+1,-}$ is defined by

$$h_i^{n+1,-}\widetilde{W}_i^{n+1,-} = hM(W_i^n) - \Delta t_n \left(\widetilde{F}_L\left(M(W_i^n), M(W_{i+1}^n), \xi_{i+1/2}^n\right) - \right.$$

$$\left. \widetilde{F}_R\left(M(W_{i-1}^n), M(W_i^n), \xi_{i-1/2}^n\right)\right),$$

Using expression (39) for

$$\widetilde{F}_L\left(M(W_i^n), M(W_{i+1}^n), \xi_{i+1/2}^n\right)$$

and (40) for

$$\widetilde{F}_R\left(M(W_{i-1}^n), M(W_i^n), \xi_{i-1/2}^n\right)$$

we obtain

$$\begin{aligned} \widetilde{W}_i^{n+1,-} = & \ \frac{1}{h_i^{n+1,-}} \int_{x_{i+1/2}-h/2}^{x_{i+1/2}+\xi_{i+1/2}^n \Delta t_n} \widetilde{R}\left(M(W_i^n), M(W_{i+1}^n), \frac{x-x_{i+1/2}}{\Delta t_n}\right) dx \\ + & \ \frac{1}{h_i^{n+1,-}} \int_{x_{i-1/2}+\xi_{i-1/2}^n \Delta t_n}^{x_{i-1/2}+h/2} \widetilde{R}\left(M(W_{i-1}^n), M(W_i^n), \frac{x-x_{i-1/2}}{\Delta t_n}\right) dx. \end{aligned} \quad (49)$$

where $x_{i+1/2} - h/2 = x_{i-1/2} + h/2$. Under the CFL condition (48), Proposition 5.1 gives

$$\forall x \in [x_{i+1/2} - h/2; x_{i+1/2} + \xi_{i+1/2}^n \Delta t_n],$$

$$L\left(\widetilde{R}\left(M(W_i^n), M(W_{i+1}^n), \frac{x - x_{i+1/2}}{\Delta t_n}\right)\right) \in \Omega_{\{\varphi_i^n\}},$$

$$\forall x \in [x_{i-1/2} + \xi_{i-1/2}^n \Delta t_n; x_{i-1/2} + h/2],$$

$$L\left(\widetilde{R}\left(M(W_{i-1}^n), M(W_i^n), \frac{x - x_{i-1/2}}{\Delta t_n}\right)\right) \in \Omega_{\{\varphi_i^n\}}.$$

The Lagrangian or ALE choice of $\xi_{i\pm1/2}^n$ ensures that the two vectors are in $\Omega_{\{\varphi_i^n\}}$.

Finally $\widetilde{W}_i^{n+1,-}$ in (49) is a convex combination of vectors in $\Omega_{\{\varphi_i^n\}}$. Because $\Omega_{\{\varphi_i^n\}}$ is convex, we obtain

$$L\left(\widetilde{W}_i^{n+1,-}\right) \in \Omega_{\{\varphi_i^n\}}.$$

□

Remark 10 Proposition 5.1 would not hold if we had chosen the Eulerian flux. Indeed, in this case, $\widetilde{W}_i^{n+1,-}$ is a convex combination of vectors in Ω, which is generally not convex.

PROPOSITION 5.3 If at the initial time

$$\forall i,\ W_i^0 \in \Omega_{\{0\}} \cup \Omega_{\{1\}},$$

and if at any time t_n,

- $\forall i,\ (a^n_{i\pm 1/2})_{L,R}$ satisfies condition (46)-(47),
- Δt_n satisfies the 1/2 CFL condition

$$\Delta t_n < \min_i \left(\frac{h}{2\max\left(\mid u_i^n - \frac{(a^n_{i+1/2})_L}{\rho_i^n} \mid, \mid u_i^n + \frac{(a^n_{i-1/2})_R}{\rho_i^n} \mid \right)} \right), \tag{50}$$

- we consider the Lagrangian scheme (19) or the ALE scheme (20),
- we apply the standard averaging (21) in the pure phase and the random averaging (22) at the interface,
- then we obtain

$$\forall n \geq 1,\ \forall i,\ W_i^n \in \Omega_{\{0\}} \cup \Omega_{\{1\}}.$$

Remark 11 In conclusion, our scheme preserves the non-negativity of ρ, the positivity of e, and preserves a sharp interface.

Proof The property is satisfied at the initial time.

Assume that at any time t_n, we have

$$\forall i,\ W_i^n \in \Omega_{\{0\}} \cup \Omega_{\{1\}}.$$

Because we consider the ALE or the Lagrangian scheme and because the Eulerian CFL condition (50) is satisfied, the ALE CFL condition (48) is also satisfied. Then, Proposition 5.2 gives

$$\forall i,\ W_i^{n+1,-} \in \Omega_{\{\varphi_i^n\}},$$

with $\varphi_i^n \in \{0;1\}$. We just have to prove the stability after projection.

If

$$(\varphi_i^n - 1/2)(\varphi_{i+1}^n - 1/2) > 0 \text{ and } (\varphi_{i-1}^n\text{-}1/2)(\varphi_i^n\text{-}1/2) > 0,$$

(then $\varphi_{i-1}^n = \varphi_i^n = \varphi_{i+1}^n$), we perform the standard averaging,

$$\begin{aligned} W_i^{n+1} &= W_i^{n+1,-} - \frac{\Delta t_n}{h}(\max(\xi^n_{i-\frac{1}{2}}, 0)(W_i^{n+1,-} - W_{i-1}^{n+1,-}) \\ &+ \min(\xi^n_{i+\frac{1}{2}}, 0)(W_{i+1}^{n+1,-} - W_i^{n+1,-})). \end{aligned}$$

It is a convex combination of $W_{i-1}^{n+1,-}$, $W_i^{n+1,-}$ and $W_{i+1}^{n+1,-}$, which are all in the convex domain $\Omega_{\{\varphi_i^n\}}$. We obtain

$$W_i^{n+1} \in \Omega_{\{\varphi_i^n\}}.$$

Else, on cells that touches the interface, we perform the Glimm projection (22). Then

$$W_i^{n+1} \in \begin{cases} \Omega_{\{\varphi_{i-1}^n\}}, \text{ if } \omega_n < \frac{\xi_{i-1/2}^n \Delta t_n}{h}, \\ \Omega_{\{\varphi_i^n\}}, \text{ if } \frac{\xi_{i-1/2}^n \Delta t_n}{h} \leq \omega_n \leq 1 + \frac{\xi_{i+1/2}^n \Delta t_n}{h}, \\ \Omega_{\{\varphi_{i+1}^n\}}, \text{ if } \omega_n > 1 + \frac{\xi_{i+1/2}^n \Delta t_n}{h}, \end{cases}$$

and the result also holds. □ It is also possible to prove the following property:

PROPOSITION 5.4 The particular solutions $u = cst$, $v = cst$ and $p = cst$ are preserved by the scheme.

In practice, we do not exactly apply the CFL condition (50). We rather compute a local time step

$$\Delta t_{i,n} = \frac{h}{2 \max \left(\mid u_i^n - \frac{(a_{i+1/2}^n)_L}{\rho_i^n} \mid, \mid u_i^n + \frac{(a_{i-1/2}^n)_R}{\rho_i^n} \mid \right)}, \tag{51}$$

and deduce an approximation of the stability time step

$$\Delta t_n = \delta \min_i \Delta t_{i,n}, \tag{52}$$

where δ is a safety factor, which satisfies $0 < \delta < 1$.

Numerically, we have observed a excellent robustness of the Lagrange and projection scheme with the relaxation ALE flux. We were not able to construct a test case leading to a crash of the simulations. Of course, it is not easy to prove rigorously the convergence of the scheme, because the projection step, which mixes deterministic and random averaging, is rather untypical.

6 GPU and OpenCL implementation

6.1 OpenCL

For performance reasons, we decided to implement the 2D scheme on recent multicore processor architectures, such as a Graphic Processing Unit (GPU). Many different hardware exist, but schematically, a GPU can be considered as a device plugged into a computer, called a host. The device is made of (see Figure 2)

- global memory (typically 1 Gb[1]);
- Compute Units (CU, typically 27).

Each compute unit is made of:

- Processing Elements (PE, typically 8);
- local (or cache) memory (typically 16 kb).

[1] the typical values are given for a NVIDIA GeForce GTX 280 GPU

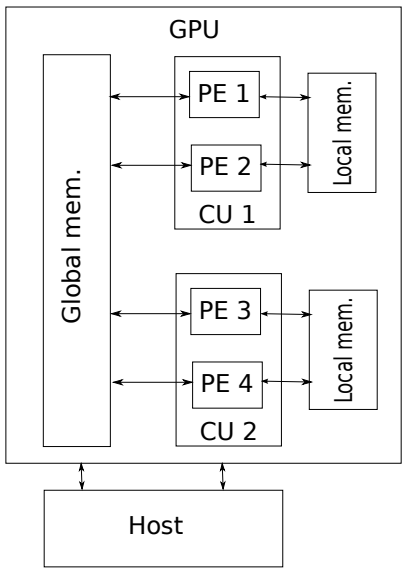


Figure 2: A (virtual) GPU with 2 Compute Units and 4 Processing Elements

The same program (a kernel) can be executed on all the processing elements at the same time, with the following rules:

- all the processing elements have access to the global memory.
- The processing elements have only access to the local memory of their compute unit.
- If several processing elements write at the same location at the same time, only one write is successful.
- The access to the global memory is slow while the access to the local memory is fast.

In order to operate a GPU, several tools are available. The CUDA environment, for instance, allows driving the NVIDIA GPUs. OpenCL is a recent set of tools, which allow to program many kinds of multicore processors, CPU or GPU. OpenCL means "Open Computing Language". It includes:

- A library of C functions, called from the host, in order to drive the GPU (or the multicore CPU);
- A C-like language for writing the kernels that will be executed on the processing elements.

OpenCL is available since september 2009 [26]. The specification is managed by the Khronos Group, which is also responsible of the OpenGL API design and evolutions. Virtually, it allows as many compute units (work-groups) and processing elements (work-items) as needed. The threads are sent to the GPU thanks to a mechanism of command queues on the real compute units and processing elements. The main advantage of the OpenCL API is its portability. The same program can run on a multicore CPU or a GPU. Many resources are available on the web for learning OpenCL. For a tutorial and simple examples, see for instance [17].

6.2 Implementation

We naturally organize the data in a (x, y) grid. In principle, the implementation is not very difficult because, thanks to the Strang splitting, the full algorithm is easy to parallelize. For recent GPU devices the number of compute units is of the order of several hundreds. This implies that the computations are very fast and that the time spent in the data memory transfers becomes the limiting factor. It is thus very important to well organize the data into memory. For instance, for computing the x-direction step (7), the data are well aligned into the memory and two successive processing elements access two neighboring memories, which permits to achieve optimal memory bandwidth (coalescing access). For the y-direction step (9), if nothing is done, two successive processing elements access different rows in memory, which leads to very slow memory access (typically ten times slower than the coalescing access!). Thus between the two steps, we have to perform an optimized transposition algorithm. The algorithm is very simple: it consists in splitting the grid into small tiles of size 32×32. Each tile is then loaded, row after row, into the cache memory of one compute unit. Each row can be read in a coalescing way. The tile is then transposed in the cache memory, which is very fast. Finally, the 32×32 tile is copied back to global memory, row after row in a coalescing way. The algorithm is described in details in [30].

That said, the algorithm for one time step is rather simple:

- we associate a processor to each cell of the grid.
- we compute the stability local time step for each cell i (see (51). If the local time step is smaller than the global one, then we replace the global time step. This implies, in some cases, concurrent memory access to the global memory if two work-items modify together the time step. From the OpenCL norm, we know that exactly one access will be successful but we cannot know which (see Section 6.1). In order to avoid instabilities we decrease the global time step by an adequate safety factor.
  This approach is very simple but we cannot guarantee that all executions of our program on different devices will give exactly the same results. It would be possible to implement a parallel algorithm for computing the smallest time step in the grid. Such algorithm is called a *reduction algorithm* and the general theory is explained in [6]. It is a little bit more complicated and in practice our simple approach is very satisfactory.

- we compute the fluxes balance in the x-direction for each cell of each row of the grid: a row or a part of a row is associated to one compute unit and one cell to one processor. As of October 2012, the OpenCL implementations generally imposes a limit (typically 1024) for the number of work-items inside a work-group. This forces us to split the rows for some large computations. The values in the cells are then loaded into the local cache memory of the compute unit. It is then possible to perform the Lagrange-projection algorithm with all the data into the cache memory in order to achieve the highest performance.

- we transpose the grid (exchange x and y) with an optimized memory transfer algorithm.

- we compute the fluxes balance in the y-direction for each row of the transposed grid. Memory accesses are optimal.

- we transpose again the grid.

6.3 Speedup

In Table 1, we compare our OpenCL code when it is run on one core or four cores of a multicore CPU, or on a recent GPU.

For evaluating the speedup of a GPU computation, we compare the same program, executed on a CPU and a GPU. It is an advantage of OpenCL over CUDA that exactly the same OpenCL software can be executed on one or several cores of a CPU or on a GPU. The transfer time from CPU to GPU for the initialization and the transfer time from GPU to CPU for post-processing are not taken into account, but are performed only at the beginning or at the end of the computation.

We observe interesting speedups for the GPU simulations compared to the one-core simulation. We also observe that OpenCL can be used for multicore CPU architectures with some efficiency. The test case corresponds to the computation of 300 time steps of the algorithm on a 1024×512 grid. One numerical flux evaluation corresponds approximately to 500 floating point operations (flop). Four flux evaluations are needed per cell and per time step. The amount of computations for this test is thus of the order of 300 Gflop.

In the algorithm, two points are important for performance:

- the first one is the optimized transposition for improving coalescent memory access. Without this optimization, the computations are approximately 10 times slower on the fastest GPUs.

- the second one is the relaxation approach. With this approach, the numerical fluxes have a simpler formulation than the exact Godunov flux, with less branch tests, and is thus more efficient on GPUs.

The performance that we have reached on GPU for this test case are similar to those achieved by other authors recently for GPU finite volume simulations (see, for instance, [36], [12]). On the best hardware, we achieve approximately 150 Gflop/s in single precision. Let us repeat that the speedups presented in Table 1 compare

hardware	time (s)	speedup
AMD A8 3850 (one core)	527	1
AMD A8 3850 (4 cores)	205	2.6
NVIDIA GeForce 320M	56	9.4
AMD Radeon HD 5850	3	175
AMD Radeon HD 7970	2	263

Table 1: Simulation times on different hardware

the same OpenCL code running on different hardware. It is clear that with some optimizations, the CPU sequential version of the code could be made more efficient. Anyway, even if we suppose that it is possible to accelerate the CPU sequential version of the code by a factor of 10 (with compilation optimizations, algorithmic optimizations and a better CPU) the OpenCL GPU software would be still 30 times faster than the CPU code.

7 Numerical results

7.1 One-dimensional results

7.1.1 Introduction

Firstly, we present some numerical results on the one-dimensional equations (7) where we do not take into account the y-velocity, it means that $v = 0$ at any time. We wish to justify our choice of projection. We compare different choices of projections for the Lagrangian approach (19). We explain why we do not use the averaging projection (21) or the Gimm projection (22) everywhere.

We could then consider the following combinations :

- the Lagrangian scheme (19) with the averaging projection (21). We call this scheme the "Averaging projection" scheme,
- the Lagrangian scheme (19) with the Glimm projection (22). We call this scheme the "Glimm projection" scheme,
- the Lagrangian scheme (19) with the projection described in Section 3.2.3. We call this scheme the "Mixed-projection" scheme,
- the ALE scheme (20) with the Glimm projection (22). We call this scheme the "ALE" scheme.

Remark 12 The averaging projection scheme is the only one that is exactly conservative, the three other schemes are only statistically conservative.

The averaging scheme produces a numerical diffusion of the mass fraction φ, we need then to use the mixture law parameters defined by (3)-(4). For the three other schemes, we do not need mixture law parameters: indeed if at initial time we have $\varphi \in \{0, 1\}$, we will have $\varphi = 0$ or $\varphi = 1$ at any time.

Quantities	Left	Right
$\rho\ (kg.m^{-3})$	10	1
$u\ (m.s^{-1})$	50	50
p (Pa)	$1,1 \times 10^5$	1×10^5
φ	1	0
γ	$1,4$	$1,1$
π	0	0

Table 2: Initial left and right states of the Riemann problem for illustrating the bad precision of the averaging projection scheme.

7.1.2 Failure of the averaging projection scheme

We consider a Riemann problem where the interface between the two fluids is located at time $t_0 = 0\,\mathrm{s}$ at position $x = 1\,\mathrm{m}$ with the left and right states of Table 2.

All quantities are plot on Figure 3 at the final time $t_1 = 0.002\,s$ on the domain $[-1; 1]$ with a number of cells equal to 500.

While the curves obtained with the Glimm projection, the mixed projection and the ALE scheme are similar to the exact solution, the averaging projection scheme gives very poor precision: we observe velocity and pressure oscillations (see figure 3). We also perform a convergence study on the velocity. We compare the error in L^1 norm

$$error = \sum_{i=1}^{N} h \mid u_i - u^{Exact}(x_i) \mid,$$

for the different schemes. In Figure 4, we observe that all schemes seems to be convergent with the same convergence rate approximately equal to 0.5.

Even if the averaging projection (21) gives a conservative scheme, its poor precision makes it useless. If we introduce different p_∞ on both sides, we can construct a Riemann problem where this scheme crashes after only one iteration: the oscillations on the pressure p lead to negative $p + p_\infty$ in the mixture zone $0 < \varphi < 1$ and the speed of sound c, given by (5), is no more defined.

7.1.3 A shock-interface interaction

We consider now a shock-interface interaction. At initial time $t_0 = 0\,s$, we have three states separated with two discontinuities. The first discontinuity is a shock starting from position $x_{shock} = -4\ m$ at velocity $\sigma = 4\ m/s$ and the second one is the interface between the two fluids, starting from position $x_{Interface} = 1m$ at velocity $u = -1\ m/s$. The initial positions of the interface and the shock are chosen in such way that they will interact at position $x_1 = 0$ m at time $t_1 = 1\,s$. We consider a grid of 500 cells on the computational domain $[-5; 2]$ m and a final time of $t_2 = 1.5\ s$. The numerical data for the right, the middle and the left states are given in Table 3. When the shock wave and the interface have interacted at time $t_1 = 1s$, the exact solution can be computed. It is simply given by the solution of a two-fluid Riemann problem between the left and the right states at final time $t = t_2 - t_1$.

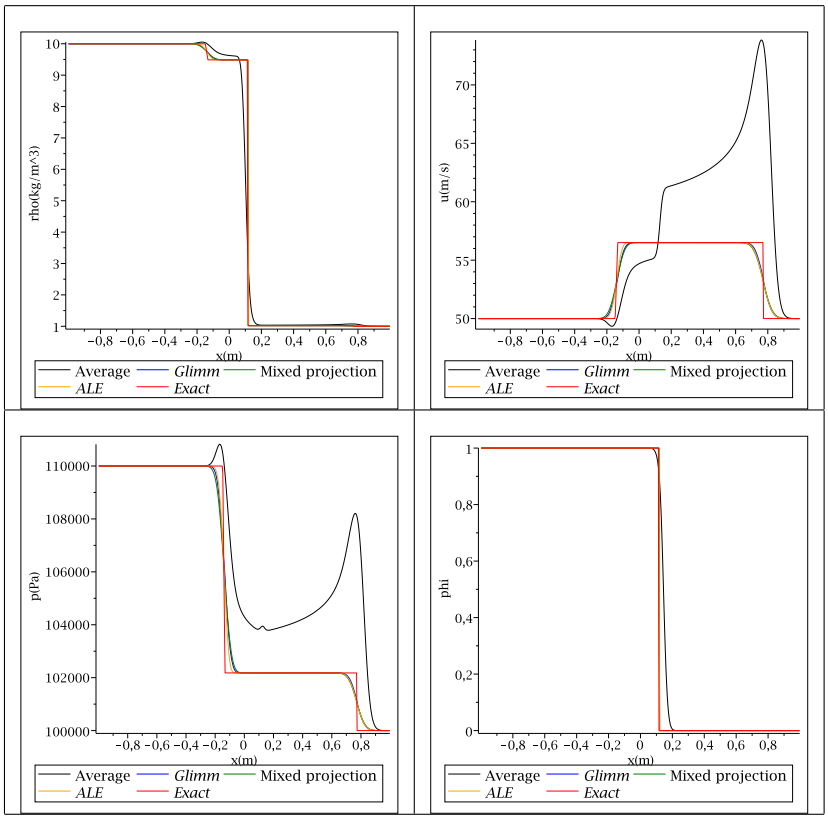


Figure 3: Density, velocity, pressure and mass fraction at the final time for the Riemann problem given by Table 2.

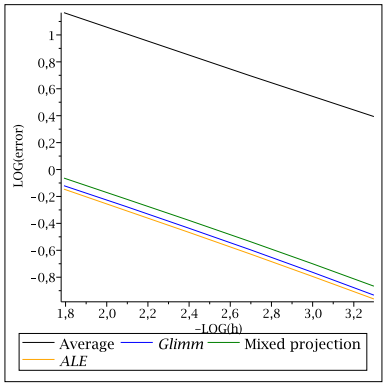


Figure 4: Convergence study for the Riemann problem given in Table 2.

Quantities	Left	Middle	Right
$\rho\ (kg.m^{-3})$	$3,488$	2	1
$u\ (m.s^{-1})$	$1,13$	-1	-1
p (Pa)	$23,33$	2	2
φ	1	1	0
γ	2	2	$1,4$
π	7	7	0

Table 3: Initial left, middle and right states for the shock-interface interaction.

In Figure 5, we compare all quantities for the different schemes. The curves for the velocity u, the pressure p and the mass fraction of gas φ are close to the exact solution. However, the density given by the Glimm projection scheme presents fast numerical oscillations. The total variation of the numerical density is not bounded. We will check that this phenomenon leads to a non-convergence of the Glimm projection scheme.

It is interesting to observe that the interface position is very well resolved (in only one mesh point) by the mixed projection scheme and the ALE scheme and that this good resolution of the contact wave also implies an improvement of the precision in the left rarefaction wave.

For this test case, we perform a convergence study on the density. We compare the error in L^1 norm with the exact solution

$$error = \frac{1}{x_{max} - x_{min}} \sum_{i=1}^{N} h \mid \rho_i - \rho^{Exact}(x_i) \mid,$$

for the different schemes. In Figure 6, we observe that the Glimm projection scheme does not converge. The other schemes seems to be convergent, the convergent rate is approximately 0.6 for the averaging projection scheme while it is approximately 0.8 for the ALE and the mixed projection scheme. We also observe that the ALE scheme is a little bit more precise than the mixed projection scheme.

7.1.4 Conclusion

The one-dimensional tests justify our choice of projection. The ALE scheme and the mixed-projection scheme give good results while the two other schemes are generally not precise and sometimes not convergent. Moreover, the interface position is very well resolved (in only one mesh point) with the ALE and mixed-projection. In the two convergence studies, the ALE scheme seems to be a little bit less diffusive than the mixed projection scheme. For the two-dimensional simulations, we decide thus to use only the ALE approach.

7.2 Pure convection

Now, we consider two-dimensional numerical tests for which we use the Strang dimensional splitting approach. The first test consists in the convection of a spherical

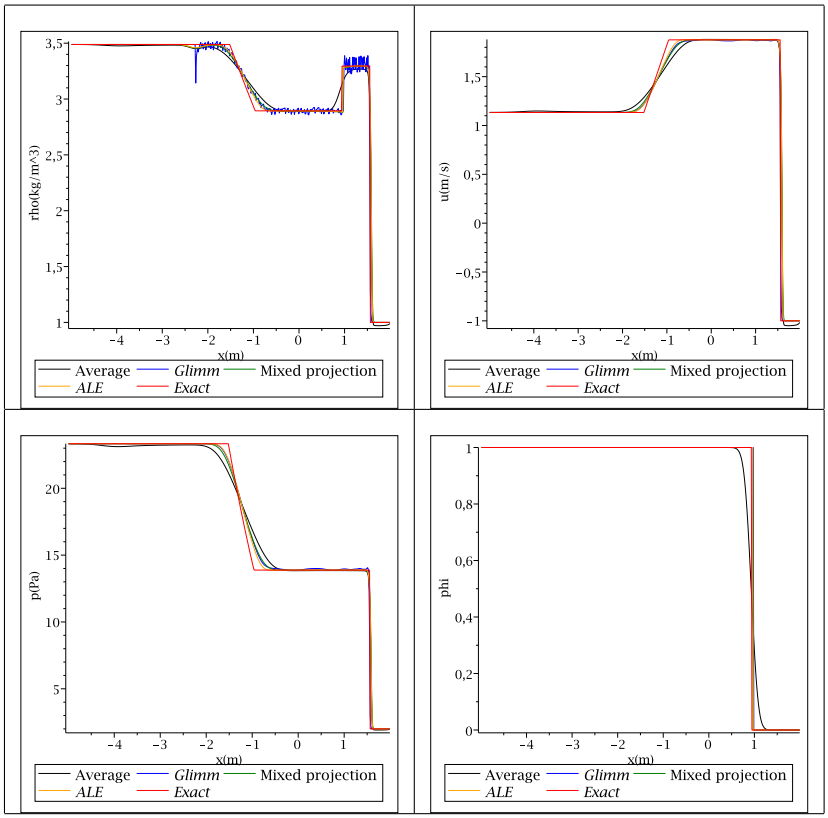


Figure 5: Density, velocity, pressure and mass fraction at the final time for the shock-interface interaction.

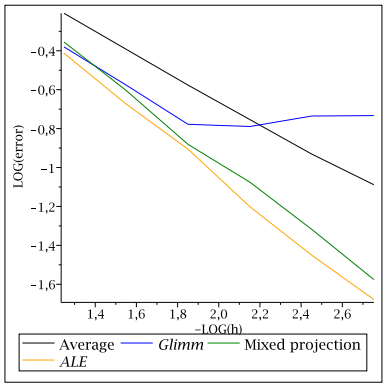


Figure 6: Convergence study for the shock-interface interaction.

Quantities	Inside the bubble	Outside the bubble
$\rho\ (kg.m^{-3})$	1.225	1000
$u\ (m.s^{-1})$	100	100
$v\ (m.s^{-1})$	-75	-75
p (Pa)	1.01325×10^5	1.01325×10^5
φ	1	0
γ	1.4	3
π	0	7.499×10^8

Table 4: Initial data for the test of a convected bubble.

bubble of gas in a liquid phase. At time $t = 0$ the bubble is in the left top corner (see Figure 7).

The initial parameters are presented in Table 4, the bubble moves from the top left corner to the bottom right corner at a constant velocity. The results are obtained with the ALE scheme and a uniform mesh of 512×512 points on the domain $[0, 1] \times [0, 1]$. The radius of the bubble is $0.1\ m$ and the final time of computation is $t_{final} = 0.0067\ s$. For the boundary conditions, we simply impose the initial data. Our algorithm does not diffuse the fraction of mass of gas φ, and thus $\varphi = 1$ in the bubble and $\varphi = 0$ outside the bubble at every time. We can plot φ to localize the bubble interface. We see in Figure 7 that the bubble moves correctly, the bubble at the final time is superimposed with the exact solution, which is plot in yellow dotted line. The last picture of the Figure 7 shows that the interface bubble is not smooth. This lack of smoothness is of course due to the pseudo-random nature of the Glimm projection.

7.3 Test of Zalesak

Now, we consider another classic two-dimensional convection numerical test. The test is proposed by Zalesak in [37] and consists in computing the rotation of a complex solid shape. We solve the equation

$$\partial_t \rho + \partial_x(\rho u) + \partial_y(\rho v) = 0, \tag{53}$$

where $u = -\Omega(y - y_0)$ and $v = \Omega(x - x_0)$. Here Ω is the constant angular velocity in rad/s and (x_0, y_0) is the axis of rotation. In order to solve this equation we use the dimensional splitting. We just have to detail the numerical scheme used to solve

$$\partial_t \rho + \partial_x(\rho u) = 0.$$

We apply the ALE numerical scheme

$$h_i^{n+1,-}\rho_i^{n+1,-} = h_i\rho_i^n - \Delta t(((\rho u)_{i+1/2}^n - \zeta_{i+1/2}^n \rho_{i+1/2}^n) - ((\rho u)_{i-1/2}^n - \zeta_{i-1/2}^n \rho_{i-1/2}^n)),$$

where $u_{i+1/2}^n = (-\Omega(y-y_0))_{i+1/2} = -\Omega(y-y_0)$ does not depend on i (it just depends on y) and

$$\rho_{i+1/2}^n = \begin{cases} \rho_i^n, & \text{if } u_{i+1/2}^n < 0, \\ \rho_{i+1}^n, & \text{if } u_{i+1/2}^n \geq 0. \end{cases}$$

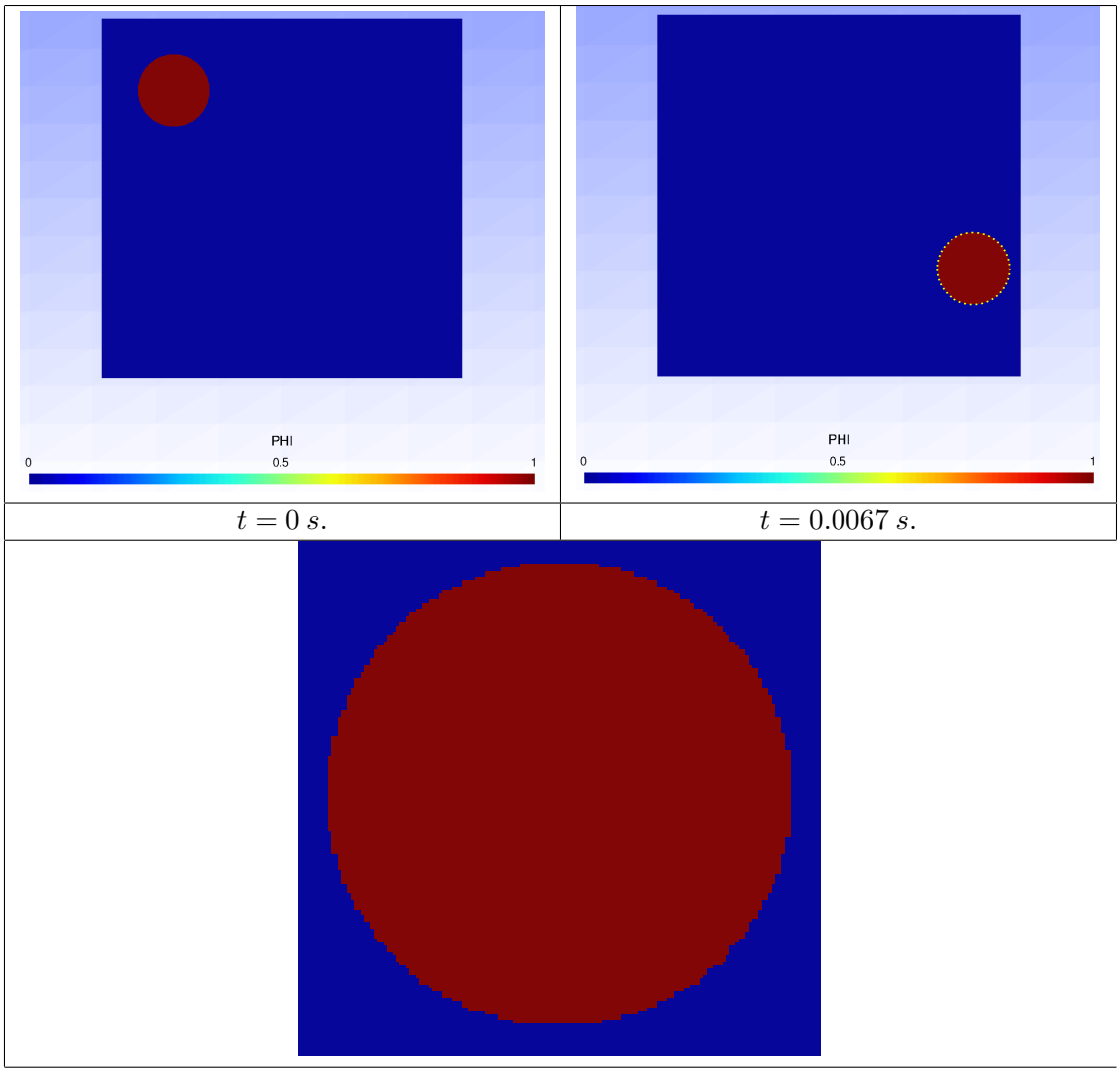


Figure 7: Initial position of the bubble (top left), final position of the bubble (top right) and zoom on the bubble interface (bottom).

The domain is $[0,1]\times[0,1]$ and the solid is a cylinder of radius $0.15\,m$. A slot is removed from the cylinder. The slot thickness is $0.05\,m$. We assume that the density inside the extruded cylinder is $\rho = 3$ and outside we have $\rho = 1$. The rotational speed is chosen in such a way that after $628\,s$ the cylinder has done one complete revolution around the central point, i.e. $\Omega = \frac{2\pi}{628}\,rad.s^{-1}$. The results are obtained on a uniform mesh of 1000×1000 points. The time step is chosen in such a way that the quantity $h_i^{n+1,-}$ remains positive, we choose

$$\Delta t \leq \frac{1}{2}\frac{\Delta x}{\max\limits_i \sqrt{u_i^2+v_i^2}}.$$

Remark 13 Because we just consider the equation (53), the system is hyperbolic with only one wave speed $\sigma = u$ then the CFL condition is less restrictive than for system (1).

In Figure 8, we compare the shape of the slotted cylinder after different times. Even after 10 revolutions, the global aspect of the solid is preserved. The length of the "bridge" seems to be the same on the top and the bottom. The interface shape is not so noisy, considering the simplicity of the Glimm approach, even if after 2 revolutions we observe some part of the solid in the bridge.

7.4 Shock-droplet interaction

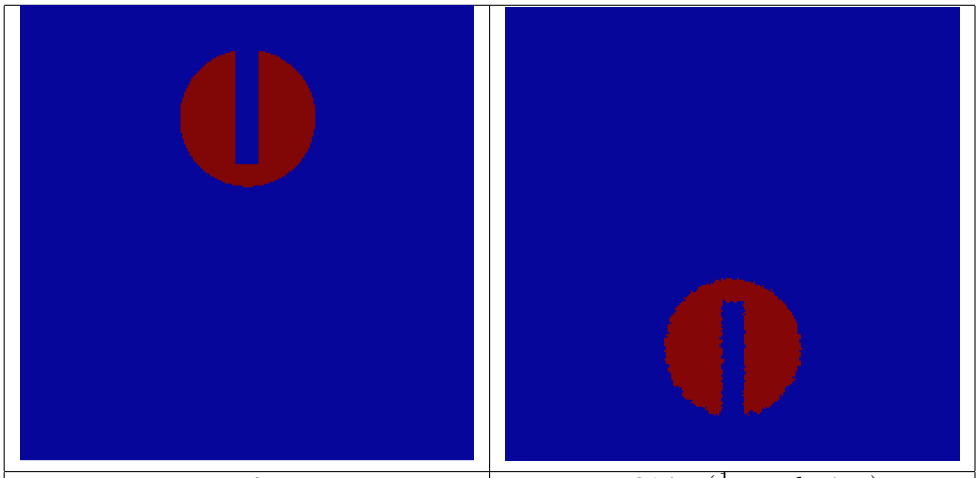
We now present a two-dimensional test that consists in simulating the impact of a Mach 1.22 shock traveling though air onto a (cylindrical) bubble of $R22$ gas. The shock speed is $\sigma = 415\,m.s^{-1}$. This test aims at simulating the experiment of [22] and has been considered by several authors [28, 32, 25]. The initial conditions are depicted in Figure 9: a bubble of $R22$ is surrounded by air inside a $L_x\times L_y$ rectangular domain. At $t = 0$, the bubble is at rest and its center is located at (X_1, Y_1). We denote by r the initial radius of the bubble. The planar shock is initially located at $x = L_s$ and moves from right to left towards the bubble. The parameters for this test are

$$L_x = 445\,\text{mm},\ L_y = 89\,\text{mm},\ L_s = 275\,\text{mm},\ X_1 = 225\,\text{mm},\ Y_1 = 44.5\,\text{mm},\ r = 25\,\text{mm}.$$

Both $R22$ and air are modeled by two perfect gases whose coefficients γ and initial states are given in Table 5.

The domain is discretized with a 5000×1000 regular mesh. Top and bottom boundary conditions are set to solid walls while we use constant state boundary conditions for the left and right boundaries.

The shock reaches the $R22$ bulk after approximately $60\,\mu s$. In the following we shall consider this time as the time origin $t = 0$. Figure 10 and Figure 11 display the evolution of the cylinder shape obtained with the ALE scheme and the experience of Haas and Sturtevant [22]. Our results can also be compared with the computations of Kokh and Lagoutière [25] because we consider the same test case with the same initial data. The profiles are obtained thanks to the fraction of mass of gas φ: our scheme preserved $\varphi = 1$ in the gas and $\varphi = 0$ in the liquid. The overall location



$t = 0\ s.$	$t = 314\ s.(\frac{1}{2}$ revolution)
$t = 628\ s.$(1 revolution)	$t = 1256\ s.$(2 revolutions)
$t = 3140\ s.$(5 revolutions)	$t = 6280\ s.$(10 revolutions)

Figure 8: Rotation of a solid shape.

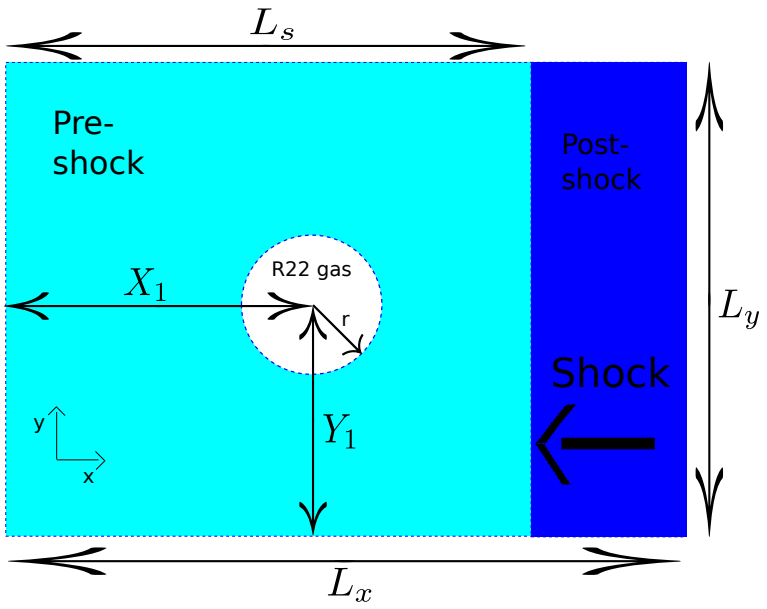


Figure 9: Air-$R22$ shock/cylinder interaction test. Description of the initial conditions.

Quantities	Air (post-shock)	Air (pre-shock)	$R22$
ρ $(kg.m^{-3})$	1.686	1.225	3.863
u $(m.s^{-1})$	-113.5	0	0
v $(m.s^{-1})$	0	0	0
p (Pa)	1.59×10^5	1.01325×10^5	1.01325×10^5
φ	0	0	1
γ	1.4	1.4	1.249
π	0	0	0

Table 5: Air-$R22$ shock/cylinder interaction test. Initial data.

Quantities	Liquid on left side	Liquid on right side	Gas
$\rho\ (kg.m^{-3})$	1030.9	1000.0	1.0
$u\ (m.s^{-1})$	300.0	0	0
$v\ (m.s^{-1})$	0	0	0
p (Pa)	3×10^9	10^5	10^5
φ	0	0	1
γ	4.4	4.4	1.4
π	6.8×10^8	6.8×10^8	0

Table 6: Liquid-gas bubble interaction test. Initial data.

of the bulk is quite similar to the experimental results (even if we consider only a two-dimensional model). The shape of the two vortices is not exactly symmetric with respect to $X = L_y/2$, because of the random nature of the Glimm projection.

7.5 Shock-bubble

We perform another shock-interface interaction test proposed in [31, 25]. It involves a gas bubble surrounded by a liquid. It is a stiffer problem, both numerically and physically. The geometry of the initial condition is depicted in Figure 9 with the following parameters:

$$L_x = 2\,\text{m},\ L_y = 1\,\text{m}, L_s = 0.04\,\text{m},\ X_1 = 0.5\,\text{m},\ Y_1 = 0.5\,\text{m},\ r = 0.4\,\text{m}.$$

The gas within the bubble is governed by a perfect gas law while the liquid is modeled with the stiffened gas law. A piston hits the left side at the velocity of 300 m/s yielding a shock pressure of about 3×10^9. The EOS parameters and initial states are given in Table 6. The Riemann problem between the liquid on left and right side will induce a 3-shock wave, with a speed $\sigma = 2181.6\,m.s^{-1}$, that will interact with the gas bubble.

The computation domain is discretized with a 3000×1000 grid and we use solid wall boundary conditions for the top and bottom boundaries, while we impose constant states at the left and right boundaries.

Figure 13 and 14 display density and pressure at several instants. As the maximum pressure increases with time, we cannot use the same scale for all the pictures. The jump of density at the bubble interface is huge, we can not see the shock position, but we can easily see the interface by observing the density field. We can clearly see the important variations of the interface topology and more specially the creation of two symmetrical vortices on each side of the axis (O, x) where O is the center the domain.

Remark 14 At some times, we obtain negative pressures in the liquid. This is not a problem because the internal energy $e = \frac{p+\gamma\pi}{\rho}$ remains positive, indeed as $p \simeq -6 \times 10^6$Pa, $\gamma_{liquid} = 4.4$, $\pi_{liquid} = 6.8 \times 10^8$ and $\rho > 0$ we obtain $e > 0$.

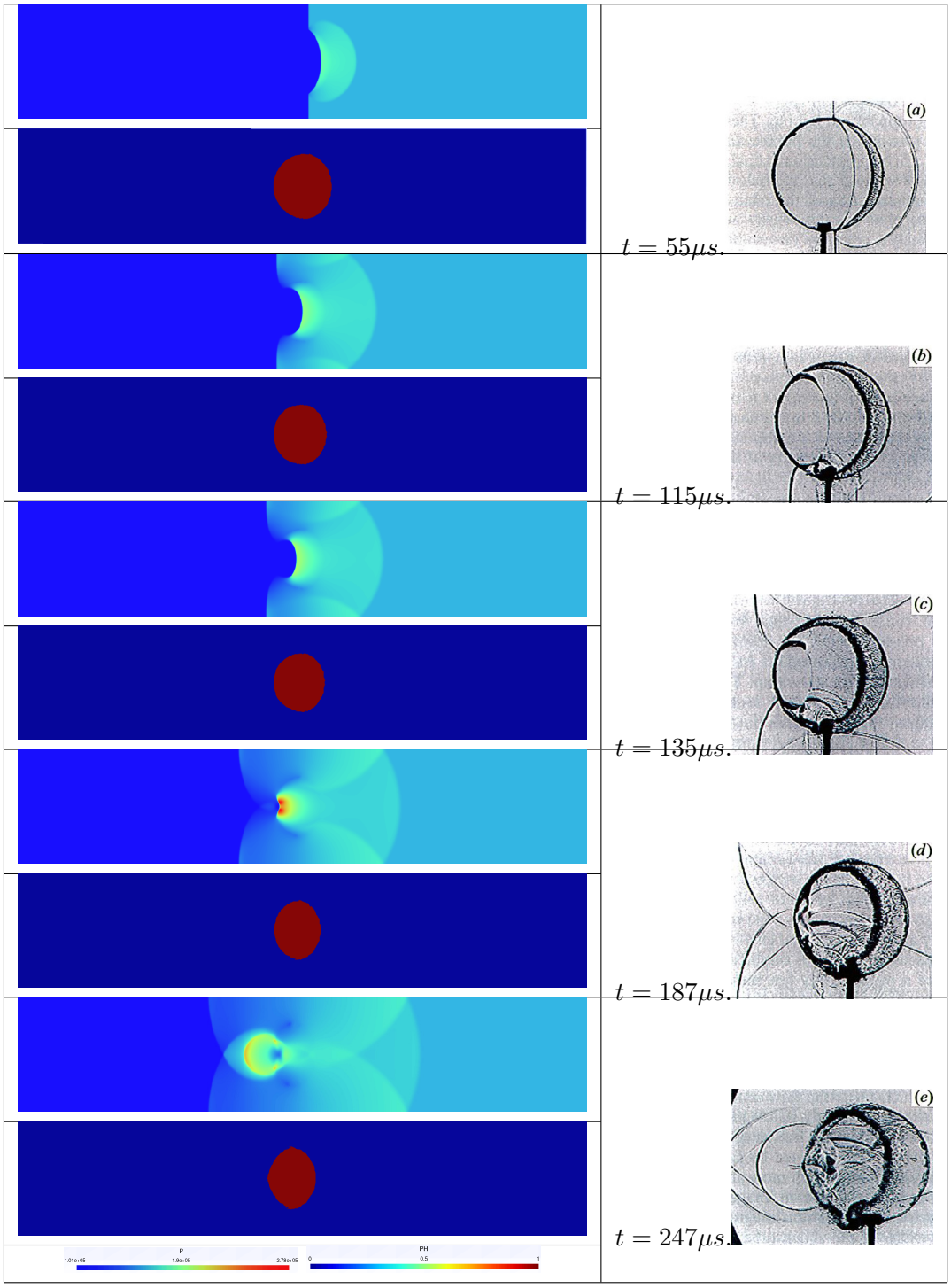


Figure 10: Air-$R22$ shock cylinder interaction test. Pressure field and interface on the left; experience of Haas and Sturtevant [22] on the right.

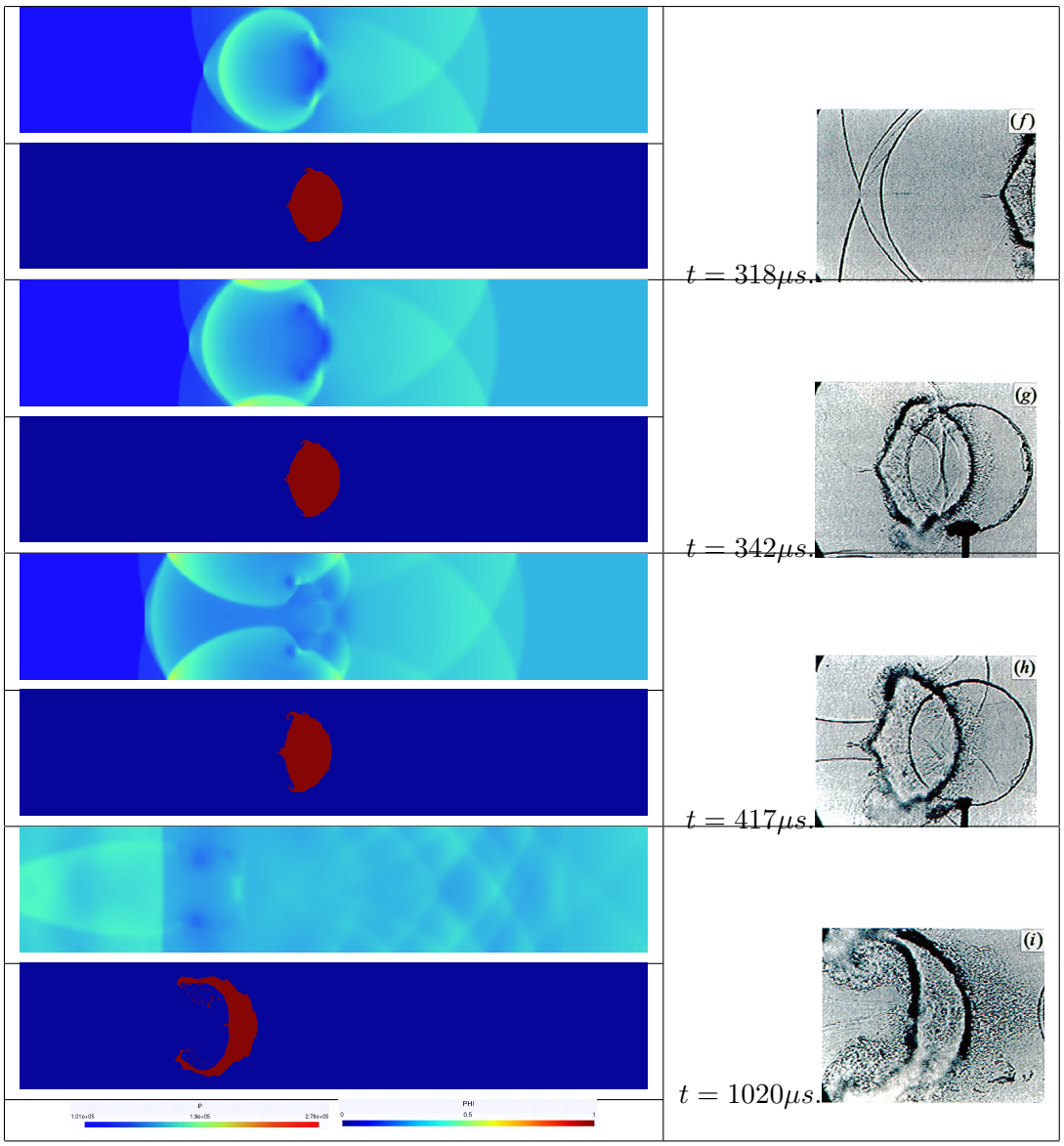


Figure 11: Air-$R22$ shock cylinder interaction test. Pressure field and interface on the left; experience of Haas and Sturtevant [22] on the right.

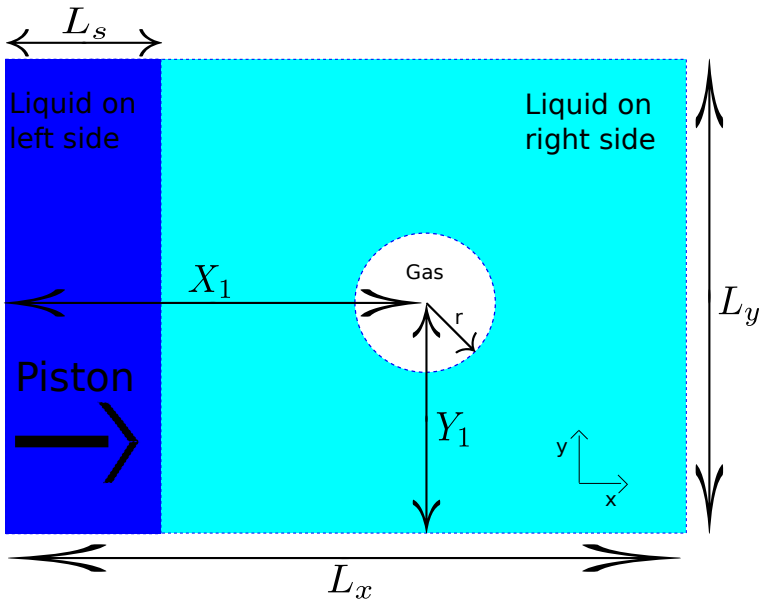


Figure 12: Liquid-gas shock/bubble interaction test. Description of the initial conditions.

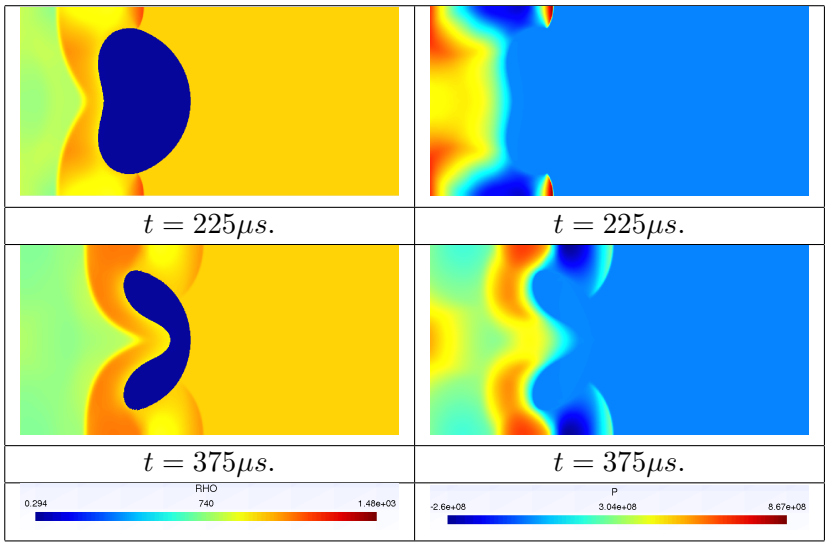


Figure 13: Liquid-gas bubble interaction test. Density on the left and pressure fied on the right.

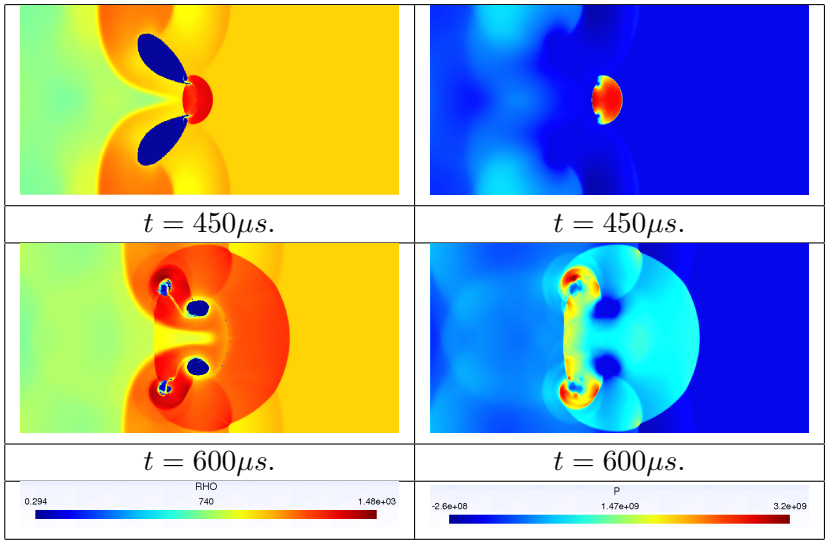


Figure 14: Liquid-gas bubble interaction test. Density on the left and pressure field on the right.

8 Conclusion

We have proposed a new method for computing two-dimensional compressible flows with interface. Our approach is based on a robust relaxation Riemann solver, coupled with a very simple random choice sampling projection at the interface. The resulting scheme has properties that are not observed in other conservative schemes of the literature:

- it preserves velocity and pressure equilibrium at the two-fluid interface;
- it is (statistically) conservative;
- the mass fraction is not diffused at all;
- it handles vacuum situations.

In addition, the algorithm is easy to parallelize on recent multicore architectures. We have implemented the scheme in the OpenCL environment. The efficiency is spectacular: compared to a standard CPU implementation, we observed that the fastest GPU computations are more than hundred times faster.

References

[1] R. Abgrall. How to prevent pressure oscillations in multicomponent flow calculations: a quasi-conservative approach. J. Comput. Phys. 125 (1996), no. 1, 150–160.

[2] M. Bachmann, P. Helluy, J. Jung, H. Mathis, S. Müller. Random sampling remap for compressible two-phase flows, 2010, submitted. `http://hal.archives-ouvertes.fr/hal-00546919`

[3] T. Barberon, P. Helluy. Finite volume simulation of cavitating flows. Computers and Fluids 34, 7, 832-858, 2005.

[4] T. Barberon, P. Helluy and S. Rouy. Practical computation of axisymmetrical multifluid flows. Int. J. Finite Vol., 1(1):34, 2004.

[5] M. Baudin, C. Berthon, F. Coquel, P. Hoche, R. Masson, Q.H. Tran. A relaxation method for two-phase flow models with hydrodynamic closure law. Numer. Math. (2005) 99: 411–440

[6] G.E. Blelloch. Scans as Primitive Parallel Operations. IEEE Transactions on Computers, C-38(11):1526–1538, November 1989.

[7] F. Bouchut. A reduced stability condition for nonlinear relaxation to conservation laws. J. Hyp. Diff. Eq. 1, 149-170, 2004.

[8] F. Bouchut. Nonlinear stability of finite volume methods for hyperbolic conservation laws and well-balanced schemes for sources. Frontiers in Mathematics. Birkhäuser Verlag, Basel, 2004.

[9] C. Chalons, F. Coquel. Capturing infinitely sharp discrete shock profiles with the Godunov scheme. Hyperbolic problems: theory, numerics, applications, 363–370, Springer, Berlin, 2008.

[10] C. Chalons, F. Coquel. Computing material fronts with a Lagrange-Projection approach. 2010. `http://arxiv.org/abs/1012.4561`

[11] C. Chalons, P. Goatin. Transport-equilibrium schemes for computing contact discontinuities in traffic flow modeling. Commun. Math. Sci., 5 (2007), no. 3, 533–551.

[12] A. Champmartin, F. de Vuyst. A low diffusive Lagrange-remap scheme for the simulation of violent air-water free-surface flows. 2013. `http://hal.archives-ouvertes.fr/hal-00798783`

[13] G. Q. Chen, C. D. Levermore, T.-P. Liu. Hyperbolic conservation laws with stiff relaxation terms and entropy. Comm. Pure Appl. Math 47, 787-830, 1994.

[14] P. Colella. Glimm's Method For Gas Dynamics. SIAM, J. Sci. Stat. Comput., Vol. 3, No 1, March 1982.

[15] F. Coquel, E. Godlewski, B. Perthame, A. In, P. Rascle. Some new Godunov and relaxation methods for two-phase flow problems, Godunov methods (Oxford, 1999), 179-188, Kluwer/Plenum, New York, 2001.

[16] F. Coquel, B. Perthame. Relaxation of energy and approximate Riemann solvers for general pressure laws in fluid dynamics. SIAM J. Numer. Anal. 35 (1998), no. 6, 2223–2249

[17] A. Crestetto, P. Helluy. An OpenCL tutorial. 2010. http://www-irma.u-strasbg.fr/~helluy/OPENCL/tut-opencl.html

[18] J.-P. Franc et al. La Cavitation : Mécanismes Physiques et Aspects Industriels. Presses Universitaires de Grenoble, 1995.

[19] T. Gallouët, J.-M. Hérard and N. Seguin. A hybrid scheme to compute contact discontinuities in one-dimensional Euler systems. M2AN Math. Model. Numer. Anal., 36(6) :1133–1159 (2003), 2002.

[20] E. Godlewski, P.-A. Raviart. Hyperbolic systems of conservation laws. Société de mathematiques appliquées et industrielles Marketing (Editeur), 1991.

[21] A. Harten, P. D. Lax, and B. Van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. SIAM Rev., 25(1) :35–61, 1983.

[22] J. F. Haas and B. Sturtevant. Interaction of weak shock waves with cylindrical ans spherical gas inhomogeneities. J. Fluid Mechanics, 181, 41-71, 1987.

[23] S. Jin, Z.-P. Xin. The relaxation schemes for systems of conservation laws in arbitrary space dimensions. Comm. Pure Appl. Math. 48, 235-276, 1995.

[24] S. Karni. Multicomponent flow calculations by a consistent primitive algorithm. J. Comput. Phys. 112 (1994), no. 1, 31–43.

[25] S. Kokh, F. Lagoutière. An anti-diffusive numerical scheme for the simulation of interfaces between compressible fluids by means of the five-equation model. J. Computational Physics, 229, 2773-2809, 2010.

[26] Khronos Group. OpenCL online documentation. http://www.khronos.org/opencl/

[27] T.-P. Liu. Hyperbolic systems with relaxation. Comm. Math. Phys, Vol. 57, 153-175, 1987.

[28] J. J. Quirk and S. Karni. On the dynamics of a shock-bubble interaction. J. Fluid Mechanics, 318, 129-163, 1996.

[29] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. J. Comput. Phys., 43(2) :357–372, 1981.

[30] Ruetsch, G., Micikevicius, P. Optimizing Matrix Transpose in CUDA. NVIDIA GPU Computing SDK, 1 – 24. 2009.

[31] R. Saurel, R. Abgrall. A simple method for compressible multifluid flows. SIAM J. Sci. Comput., 21 (3), 1115-1145, 1999.

[32] K. M. Shyue. A wave-propagation based volume tracking method for compressible multicomponent flow in two space dimensions. J. Comput. Phy., 215, 219-244, 2006.

[33] Strang, Gilbert On the construction and comparison of difference schemes. SIAM J. Numer. Anal. 5 1968 506–517.

[34] J. J. W. van der Vegt, H. van des Ven. Space-Time Discontinuous Galerkin Finite Element Method with Dynamic Grid Motion for Inviscid Compressible Flows. J. Comput. Phy., 182, 546-585, 2002.

[35] E. F. Toro. Riemann solvers and numerical methods for fluid dynamics, 2nd edition. Springer, 1999

[36] Wong, Hon-Cheng; Wong, Un-Hong; Feng, Xueshang; Tang, Zesheng Efficient magnetohydrodynamic simulations on graphics processing units with CUDA. Comput. Phys. Comm. 182 (2011), no. 10, 2132–2160

[37] S. T. Zalesak. Fully Multidimensional Flux-Corrected Algorithms for Fluids. J. Computational Physics, 31, 335-362, 1969.