



HAL
open science

OpenCL numerical simulations of two-fluid compressible flows with a 2D random choice method

Philippe Helluy, Jonathan Jung

► **To cite this version:**

Philippe Helluy, Jonathan Jung. OpenCL numerical simulations of two-fluid compressible flows with a 2D random choice method. *International Journal on Finite Volumes*, 2013, 10, pp.1-38. hal-00759135

HAL Id: hal-00759135

<https://hal.science/hal-00759135>

Submitted on 30 Nov 2012

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

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

OPENCL NUMERICAL SIMULATIONS OF TWO-FLUID COMPRESSIBLE FLOWS WITH A 2D RANDOM CHOICE METHOD

PHILIPPE HELLUY, JONATHAN JUNG

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 classical 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 [13]. 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 multicore processors and GPU, using the OpenCL environment.

1. INTRODUCTION

Compressible two-fluid flows are difficult to numerically simulate. Indeed, as first discovered in [1] and [14], classical 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, 14]. The resulting schemes are generally not conservative. Based on previous works of Chalons and Goatin [9] and Chalons and Coquel [6], we have proposed in [13] 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 [13], we have tested the random scheme on one-dimensional test cases. The random scheme presents interesting properties: (statistical) conservation, faster convergence, it preserves the pressure and velocity equilibrium at the interface, it allows to perform computations that are not feasible with other classical schemes.

In this paper, we extend the method to two-dimensional equations, thanks to Strang splitting.

The simplicity of the approach allows also an easy implementation of the method on recent multicore 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, modelling a liquid-gas compressible flow

$$(2.1) \quad \partial_t W + \partial_x F(W) + \partial_y G(W) = 0,$$

where

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

and

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

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 a stiffened gas pressure law

$$p(\rho, e, \varphi) = (\gamma(\varphi) - 1)\rho e - \gamma(\varphi)\pi(\varphi),$$

where γ and π 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, classical numerical schemes generally produce an artificial diffusion of the mass fraction, and in the numerical approximation we may observe $1 > \varphi > 0$. In classical conservative schemes, the artificial mixing zone implies a loss of the velocity and pressure equilibrium at the interface. It is possible to recover the equilibrium by relaxing the conservation property of the scheme as in [20].

3. RANDOM CHOICE NUMERICAL METHOD

3.1. Directional splitting. We consider an increasing sequence of time 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 can use Strang dimensional splitting [22], which amounts to solve, for a time step Δt the Cauchy problem

$$(3.1) \quad \partial_t W + \partial_x F(W) = 0,$$

$$(3.2) \quad W(x, y, 0) = W^n(x, y).$$

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

$$(3.3) \quad \partial_t V + \partial_y G(V) = 0,$$

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 (2.1) at order 1 in Δt . It is classical and easy to make it second order [22]. In addition, in our application, thanks to the rotational invariance of the Euler equations, the equations (3.1) and (3.3) 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

$$(3.4) \quad \partial_t W + \partial_x F(W) = 0,$$

$$(3.5) \quad W(x, 0) = W_0(x).$$

We use a variant of the scheme developed in [13], which we recall below.

3.2. Random choice approach. In this section we consider the numerical scheme for solving (3.1) for (x, t) in $[a, b] \times \mathbb{R}^+$. We consider a sequence of time t_n , $n \in \mathbb{N}$ such that the time step $\tau_n := t_{n+1} - t_n > 0$. 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 $x_{i\pm 1/2} = x_i \pm h/2$. We look for an approximation of $W(x_i, t_n)$

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

Each time step of our scheme is made of two stages.

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

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

Integrating the conservation law (3.1) on the space time trapezoid

$$\bigcup_{t \in]t_n, t_{n+1}[}]x_{i-1/2} + (t - t_n)\xi_{i-1/2}^n, x_{i+1/2} + (t - t_n)\xi_{i+1/2}^n[\times \{t\},$$

we obtain the following finite volume approximation

$$h_i^{n+1,-} W_i^{n+1,-} - h W_i^n + \tau_n (F_{i+1/2}^n - F_{i-1/2}^n) = 0.$$

The new size of cell i is given by

$$h_i^{n+1,-} = x_{i+1/2}^{n+1,-} - x_{i-1/2}^{n+1,-} = h + \tau_n (\xi_{i+1/2}^n - \xi_{i-1/2}^n).$$

The Arbitrary Lagrangian Eulerian (ALE) numerical flux is of the form

$$F_{i+1/2}^n = F(W_{i+1/2}^n) - \xi_{i+1/2}^n W_{i+1/2}^n.$$

The intermediate state $W_{i+1/2}^n$ is obtained by the resolution of a Riemann problem. More precisely, we consider the entropy solution of

$$\begin{aligned} \partial_t V + \partial_x F(V) &= 0, \\ V(x, 0) &= \begin{cases} V_L & \text{if } x < 0, \\ V_R & \text{if } x > 0, \end{cases} \end{aligned}$$

which is denoted by

$$R(V_L, V_R, x/t) = V(x, t).$$

The intermediate state is then

$$W_{i+1/2}^n = R(W_i^n, W_{i+1}^n, \xi_{i+1/2}^n).$$

In practice, R can be an exact or approximate Riemann solver. We first describe the method for the exact Riemann solver. A more efficient approximate Riemann solver, based on a relaxation approach, is also described below.

3.2.2. Choice of the interface velocity. Several choices are possible for the interface velocity $\xi_{i+1/2}^n$. The standard Eulerian scheme choice is

$$(3.6) \quad \xi_{i+1/2}^n = 0,$$

and the standard Lagrangian consist to choose

$$(3.7) \quad \xi_{i+1/2}^n = u_{i+1/2}^n,$$

where $u_{i+1/2}^n$ is the contact discontinuity velocity in the resolution of the Riemann problem $R(W_i^n, W_{i+1}^n, x/t)$.

A better choice is to take $\xi_{i+1/2}^n = u_{i+1/2}^n$ only at the liquid-gas interface and $\xi_{i+1/2}^n = 0$ elsewhere. We explain below why this choice is better. The cell interface $i + 1/2$ is at the liquid-gas interface if the following condition is satisfied

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

because $\varphi = 0$ in the liquid and $\varphi = 1$ in the gas. In short

$$(3.8) \quad \xi_{i+1/2}^n = \begin{cases} u_{i+1/2}^n & \text{if } (\varphi_i - 1/2)(\varphi_{i+1} - 1/2) < 0, \\ 0 & \text{else.} \end{cases}$$

In the sequel we will denote by the ‘‘Lagrange scheme’’ the scheme corresponding to choice (3.7) and by the ‘‘ALE scheme’’ the scheme corresponding to choice (3.8).

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 the solution $W_i^{n+1,-}$, which is naturally defined on the moved cells $C_i^{n+1,-} =]x_{i-1/2}^{n+1,-}, x_{i+1/2}^{n+1,-}[$. The averaging is different if the cell touches the liquid gas interface or not. More precisely, if the cell is not at the interface, i.e. if

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

then we perform the standard averaging

$$(3.9) \quad \begin{aligned} W_i^{n+1} &= W_i^{n+1,-} - \frac{\tau_n}{h} (\max(\xi_{i-\frac{1}{2}}^n, 0)(W_i^{n+1,-} - W_{i-1}^{n+1,-}) \\ &\quad + \min(\xi_{i+\frac{1}{2}}^n, 0)(W_{i+1}^{n+1,-} - W_i^{n+1,-})). \end{aligned}$$

Remark: if the interface velocity $\xi_{i\pm 1/2}^n = 0$, we simply obtain

$$W_i^{n+1} = W_i^{n+1,-}.$$

On the other hand, if the cell touches the interface

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

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

$$(3.10) \quad W_i^{n+1} = \begin{cases} W_{i-1}^{n+1,-}, & \text{if } \omega_n < \frac{\xi_{i-1/2}^n \tau_n}{h}, \\ W_i^{n+1,-}, & \text{if } \frac{\xi_{i-1/2}^n \tau_n}{h} \leq \omega_n \leq 1 + \frac{\xi_{i-1/2}^n \tau_n}{h}, \\ W_{i+1}^{n+1,-}, & \text{if } \omega_n > 1 + \frac{\xi_{i-1/2}^n \tau_n}{h}. \end{cases}$$

A good choice for the pseudo-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 [23]. In practice, we consider the $(5, 3)$ van der Corput sequence.

Remark: it is not possible to perform the random averaging in all the cells, because the resulting scheme is generally not BV stable (see [13] and also a numerical example below).

3.2.4. *Relaxation solver.* To simplify the presentation, in this part we first consider the Eulerian scheme (see equation (3.6) that can be written as

$$(3.11) \quad W_i^{n+1} = W_i^n - \frac{\tau_n}{h} (F_{i+1/2}^n - F_{i-1/2}^n),$$

where $F_{i+1/2}^n = F(W_i^n, W_{i+1}^n)$ is the interface numerical flux. In the particular case of the Godunov scheme (see[23]), we have

$$(3.12) \quad F_{i+1/2}^n = F(W_{i+1/2}^n) = F(R(W_i^n, W_{i+1}^n, 0)),$$

where $R(W_i^n, W_{i+1}^n, x/t) = W(x, t)$ is the solution of the Riemann problem

$$\begin{aligned} \partial_t W + \partial_x F(W) &= 0, \\ W(x, 0) &= \begin{cases} W_i^n & \text{if } x < 0, \\ W_{i+1}^n & \text{if } x > 0. \end{cases} \end{aligned}$$

The exact solution could be computed (see [3]) but it requires to solve iteratively a non-linear equation. Numerically, this computation is not efficient on GPU, because it involves many different branch tests, which are not easy to parallelize.

It is thus indicated to replace the exact Riemann solver by an approximated one. However we must construct it carefully. Indeed, other classical solvers, such as the Roe or VFRoe Riemann solvers [8, 18], 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 much more robust solver. An obvious idea is then to adapt the Harten-Lax-van Leer (HLL) scheme to our case [11]. It appears that it is not so easy because we are considering two-fluid flows. Then we decided to use an approximate Riemann solver based on a relaxation approach proposed by Coquel [7] and Bouchut [5]. At some points, we have to adapt the Bouchut presentation, because we want to consider two-fluid flows.

The relaxation approach consists in solving the following extended system

$$(3.13) \quad \begin{aligned} \partial_t \rho + \partial_x(\rho u) &= 0, \\ \partial_t(\rho u) + \partial_x(\rho u^2 + \beta) &= 0, \\ \partial_t(\rho v) + \partial_x(\rho uv) &= 0, \\ \partial_t(\rho E) + \partial_x((\rho E + \beta)u) &= 0, \\ \partial_t(\rho \varphi) + \partial_x(\rho \varphi u) &= 0, \\ \partial_t\left(\frac{\rho \beta}{c^2}\right) + \partial_x\left(\frac{\rho \beta u}{c^2} + u\right) &= 0, \\ \partial_t(\rho c) + \partial_x(\rho cu) &= 0, \\ \partial_t(\rho s) + \partial_x(\rho cs) &= 0, \end{aligned} \tag{3.14}$$

with

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

and where there are 8 unknowns $(\rho, u, v, e, \varphi, \beta, c, s)$ for our initial system of 5 unknowns (ρ, u, v, e, φ) . The additional unknown β represents a relaxed pressure, s represents a relaxed entropy and c is a convected pressure law parameter (more precisely c/ρ is homogeneous to a sound speed). One has to take care that in (3.13)-(3.14), ρ, e, s, β and c are understood as independent variables. We can write the system (3.13)-(3.14) in the conservative form

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

where

$$\widetilde{W} = (\rho, \rho u, \rho v, \rho E, \rho \varphi, \frac{\rho \beta}{c^2}, \rho c, \rho s)^T$$

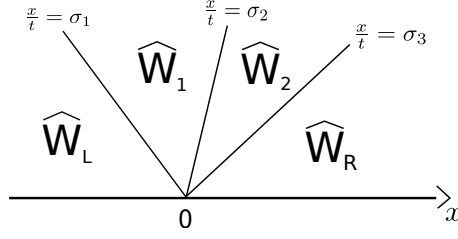


FIGURE 3.1. Structure of the approximate Riemann solver.

and

$$\widetilde{F}(\widetilde{W}) = (\rho u, \rho u^2 + \beta, \rho uv, (\rho E + \beta)u, \rho \varphi u, \frac{\rho \beta u}{c^2} + u, \rho cu, \rho su)^T.$$

We choose \widetilde{W} such that the first five components of \widetilde{W} are exactly the components of W at the beginning of the time step. In addition, we have to set β , s and c and explain how we compute these additional variables from the state W . In other words, we have to duplicate some variables computed from W at the beginning of the time step. The duplication has to be performed in such a way that the resulting solver is consistent with the exact one and that the whole scheme satisfies some stability properties.

Thanks to the additional relaxed variables, the eigenvalues of the new system are linearly degenerated. Thus we can compute easily the exact solution to the relaxed Riemann problem. It has three wave speeds $\sigma_1 = u - \frac{c}{\rho}$, $\sigma_2 = u$, $\sigma_3 = u + \frac{c}{\rho}$, with two intermediate states that we shall index by $._1$ and $._2$ (see Figure 3.1). We notice that $c_1 = c_L$ and $c_2 = c_R$. Then, according to expression of the Riemann invariants for the first and thirch wave and the fact that u and β are two independant Riemann invariants for the central wave, the intermediate states are obtained by the relations

$$\begin{aligned} u_1 &= u_2, & \beta_1 &= \beta_2, \\ v_L &= v_1, & v_2 &= v_R, \\ s_L &= s_1, & s_2 &= s_R, \\ \varphi_L &= \varphi_1, & \varphi_2 &= \varphi_R, \\ (\beta + uc)_L &= (\beta + uc)_1, & (\beta - uc)_2 &= (\beta - uc)_R, \\ \left(\frac{1}{\rho} + \frac{\beta}{c^2}\right)_L &= \left(\frac{1}{\rho} + \frac{\beta}{c^2}\right)_1, & \left(\frac{1}{\rho} + \frac{\beta}{c^2}\right)_2 &= \left(\frac{1}{\rho} + \frac{\beta}{c^2}\right)_R, \\ \left(e - \frac{\beta^2}{2c^2}\right)_L &= \left(e - \frac{\beta^2}{2c^2}\right)_1, & \left(e - \frac{\beta^2}{2c^2}\right)_2 &= \left(e - \frac{\beta^2}{2c^2}\right)_R. \end{aligned}$$

The wave speeds are given by

$$\sigma_1 = u_L - \frac{c_L}{\rho_L}, \quad \sigma_2 = u_1 = u_2, \quad \sigma_3 = u_R + \frac{c_R}{\rho_R}.$$

Then the solution is

$$(3.15) \quad \frac{1}{\rho_1} = \frac{1}{\rho_L} + \frac{c_R(u_R - u_L) + \beta_L - \beta_R}{c_L(c_L + c_R)},$$

$$(3.16) \quad \frac{1}{\rho_2} = \frac{1}{\rho_R} + \frac{c_L(u_R - u_L) + \beta_R - \beta_L}{c_R(c_L + c_R)},$$

$$(3.17) \quad u_1 = u_2 = \frac{\beta_L - \beta_R + c_L u_L + c_R u_R}{c_R + c_L},$$

$$v_1 = v_L, \quad v_2 = v_R,$$

$$(3.18) \quad \beta_1 = \beta_2 = \frac{c_L \beta_R + c_R \beta_L + c_L c_R (u_L - u_R)}{c_L + c_R},$$

$$e_1 = e_L - \frac{\beta_L^2 - \beta_1^2}{2c_L^2},$$

$$e_2 = e_R - \frac{\beta_R^2 - \beta_2^2}{2c_R^2},$$

$$(3.19) \quad \varphi_1 = \varphi_L, \quad \varphi_2 = \varphi_R.$$

Of course, we have to relate the relaxed Riemann solver to our initial set of variables. In practice, as we already said, at the beginning of the time step, we duplicate the relaxed variables, which amounts to set

$$(3.20) \quad \beta_L = p(\rho_L, e_L, \varphi_L), \quad \beta_R = p(\rho_R, e_R, \varphi_R),$$

$$(3.21) \quad s_L = s(\rho_L, e_L, \varphi_L), \quad s_R = s(\rho_R, e_R, \varphi_R).$$

The objective is now to define proper c_L and c_R and the numerical flux in such a way that the Riemann solver satisfies good properties. We will study if our scheme is stable, it means that under some CFL condition, we have

$$W_i^n \in \mathcal{W} \text{ for all } i \Rightarrow W_i^{n+1} \in \mathcal{W} \text{ for all } i,$$

where we denote by \mathcal{W} the convex domain

$$\mathcal{W} = \{(\rho, \rho u, \rho v, \rho E, \rho \varphi)^T : \rho \geq 0; e = E - \frac{u^2 + v^2}{2} \geq 0\}.$$

In order to study the stability of the scheme we will use the right and left numerical fluxes associated to the approximate Riemann solver of the extended system (3.13)-(3.14). These fluxes are defined by [11]

$$(3.22) \quad \tilde{F}_L(\tilde{W}_L, \tilde{W}_R) = \tilde{F}(\tilde{W}_L) - \int_{-\infty}^0 (\tilde{R}(\tilde{W}_L, \tilde{W}_R, \omega) - \tilde{W}_L) d\omega,$$

$$(3.23) \quad \tilde{F}_R(\tilde{W}_L, \tilde{W}_R) = \tilde{F}(\tilde{W}_R) + \int_0^{+\infty} (\tilde{R}(\tilde{W}_L, \tilde{W}_R, \omega) - \tilde{W}_R) d\omega.$$

The conservativity identity $\tilde{F}_L(\tilde{W}_L, \tilde{W}_R) = \tilde{F}_R(\tilde{W}_L, \tilde{W}_R)$ becomes

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

Conservativity thus enables to define the intermediate fluxes \tilde{F}_1 and \tilde{F}_2 by

$$\begin{aligned} \tilde{F}_1 &= \tilde{F}(\tilde{W}_L) + \sigma_1(\tilde{W}_1 - \tilde{W}_L), \\ \tilde{F}_2 &= \tilde{F}(\tilde{W}_R) - \sigma_3(\tilde{W}_R - \tilde{W}_2), \end{aligned}$$

which can be considered as a generalization of the Rankine-Hugoniot relations. The numerical flux of the extended system (3.13)-(3.14) is then given by

$$\tilde{F}(\tilde{W}_L, \tilde{W}_R) = \tilde{F}_L(\tilde{W}_L, \tilde{W}_R) = \tilde{F}_R(\tilde{W}_L, \tilde{W}_R) = \begin{cases} \tilde{F}(\tilde{W}_L), & \text{if } 0 \leq \sigma_1, \\ \tilde{F}_1, & \text{if } \sigma_1 \leq 0 \leq \sigma_2, \\ \tilde{F}_2, & \text{if } \sigma_2 \leq 0 \leq \sigma_3, \\ \tilde{F}(\tilde{W}_R), & \text{if } \sigma_3 \leq 0. \end{cases}$$

The computation of the intermediate fluxes \tilde{F}_1 and \tilde{F}_2 gives us

$$\begin{aligned} \tilde{F}_1 &= (\rho_1 u_1, \rho_1 u_1^2 + \beta_1, \rho_1 u_1 v_1, (\rho_1 (e_1 + \frac{u_1^2 + v_1^2}{2}) + \beta_1) u_1, \rho_1 u_1 \varphi_1, \frac{\rho_1 \beta_1 u_1}{c_1^2} + u_1, \rho_1 c_1 u_1, \rho_1 s_1 u_1)^T, \\ &= \tilde{F}(\tilde{W}_1), \\ \tilde{F}_2 &= (\rho_2 u_2, \rho_2 u_2^2 + \beta_2, \rho_2 u_2 v_2, (\rho_2 (e_2 + \frac{u_2^2 + v_2^2}{2}) + \beta_2) u_2, \rho_2 u_2 \varphi_2, \frac{\rho_2 \beta_2 u_2}{c_2^2} + u_2, \rho_2 c_2 u_2, \rho_2 s_2 u_2)^T, \\ &= \tilde{F}(\tilde{W}_2). \end{aligned}$$

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. $\tilde{F}(\tilde{W}_L, \tilde{W}_R) = \tilde{F}(\tilde{W}_*)$ for some $\tilde{W}_* \in \mathbb{R}^8$).

This property is generally not true for the class of so-called ‘‘Godunov type finite volume schemes’’ as defined by Harten in [11]. For this class of schemes, the numerical flux is defined by integrating the approximate Riemann solver as in (3.22) and (3.23). And there is no reason that the numerical flux can be expressed in the form (3.12).

The numerical flux $F(W_L, W_R)$ for the original system (3.1) is obtain by keeping only the first five components of our relaxed fluxes, we deduce

$$(3.24) \quad F(W_L, W_R) = \begin{cases} F(W_L), & \text{if } 0 \leq \sigma_1, \\ F_1, & \text{if } \sigma_1 \leq 0 \leq \sigma_2, \\ F_2, & \text{if } \sigma_2 \leq 0 \leq \sigma_3, \\ F(W_R), & \text{if } \sigma_3 \leq 0, \end{cases}$$

where

$$\begin{aligned} F_1 &= (\rho_1 u_1, \rho_1 u_1^2 + \beta_1, \rho_1 u_1 v_1, (\rho_1 (e_1 + \frac{u_1^2 + v_1^2}{2}) + \beta_1) u_1, \rho_1 u_1 \varphi_1)^T, \\ F_2 &= (\rho_2 u_2, \rho_2 u_2^2 + \beta_2, \rho_2 u_2 v_2, (\rho_2 (e_2 + \frac{u_2^2 + v_2^2}{2}) + \beta_2) u_2, \rho_2 u_2 \varphi_2)^T. \end{aligned}$$

Generally $\beta_1 \neq p(\rho_1, e_1, \varphi_1)$ and $\beta_2 \neq p(\rho_2, e_2, \varphi_2)$, and it is not possible to write the $F(W_L, W_R) = F(W^*)$ for some $W^* \in \mathbb{R}^5$.

Remark: The positivity of ρ_1 and ρ_2 is not guaranteed from (3.15)-(3.16). This is a requirement that constraints c_L and c_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 $\sigma_2 - \sigma_1 = \frac{c_L}{\rho_1}$ and $\sigma_3 - \sigma_2 = \frac{c_R}{\rho_2}$.

Property: In the Riemann problem (3.13)-(3.14) with initially (3.20) and (3.21), if $\rho_L > 0$ and $\rho_R > 0$, we define the relaxation speeds c_L and c_R by

$$\begin{aligned} \text{if } p_R - p_L \geq 0, \quad & \begin{cases} \frac{c_L}{\rho_L} = \sqrt{\frac{\partial p}{\partial \rho}(\rho_L, s_L, \varphi_L)} + \alpha \max\left(\frac{p_R - p_L}{\rho_R \sqrt{\frac{\partial p}{\partial \rho}(\rho_R, s_R, \varphi_R)}} + u_L - u_R, 0\right), \\ \frac{c_R}{\rho_R} = \sqrt{\frac{\partial p}{\partial \rho}(\rho_R, s_R, \varphi_R)} + \alpha \max\left(\frac{p_L - p_R}{c_L} + u_L - u_R, 0\right), \end{cases} \\ \text{if } p_R - p_L \leq 0, \quad & \begin{cases} \frac{c_R}{\rho_R} = \sqrt{\frac{\partial p}{\partial \rho}(\rho_R, s_R, \varphi_R)} + \alpha \max\left(\frac{p_L - p_R}{\rho_L \sqrt{\frac{\partial p}{\partial \rho}(\rho_L, s_L, \varphi_L)}} + u_L - u_R, 0\right), \\ \frac{c_L}{\rho_L} = \sqrt{\frac{\partial p}{\partial \rho}(\rho_L, s_L, \varphi_L)} + \alpha \max\left(\frac{p_R - p_L}{c_R} + u_L - u_R, 0\right), \end{cases} \end{aligned}$$

where $\alpha = \max(\frac{\gamma_L + 1}{2}, \frac{\gamma_R + 1}{2})$, $s = \frac{p + \pi}{\rho^\gamma}$ then $\frac{\partial p}{\partial \rho}(\rho, s, \varphi) = \gamma(\varphi)s\rho^{\gamma(\varphi)-1}$.

If we assume that the time step τ_n satisfies the CFL condition

$$(3.25) \quad \begin{cases} \tau_n \cdot (\sigma_1)_i^n \geq -\frac{h}{2}, \\ \tau_n \cdot (\sigma_2)_i^n \leq \frac{h}{2}, \end{cases} \quad \forall i$$

then, the Eulerian scheme (3.11) with the interface fluxes given by (3.24) has now the following properties

- (1) it preserves the nonnegativity of ρ ,
- (2) it preserves the positivity of e ,
- (3) it satisfies discrete entropy inequalities for single fluid flows,
- (4) stationary contact discontinuities where $u = 0$, $p = cst$ are exactly resolved,
- (5) it handles data with vacuum.

For a proof, we refer to [5]¹.

In practice, we do not exactly apply the CFL condition (3.25). We rather compute a local time step

$$(3.26) \quad \tau_{i,n} = \frac{h}{2 \max(|(\sigma_1)_i^n|, |(\sigma_3)_i^n|)},$$

and deduce an approximation of the stability time step

$$(3.27) \quad \tau_n = \delta \min_i \tau_{i,n},$$

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

The lagrangian scheme (3.7) can be treated in a similar way, in this case the interface flux is given by a formula that depends only on an intermediate state

$$F(W_L, W_R) = (0, \beta_1, 0, u_1 \beta_1, 0)^T = (0, \beta_2, 0, u_2 \beta_2, 0)^T,$$

where β_1 and u_1 are given by the same formula (3.17)-(3.18). We can prove similarly at it was done in [5] that under the CFL condition (3.25) the Lagrangian scheme satisfies the properties 1, 2, 4 and 5 previously given.

We are able to compute a Eulerian and a Lagrangian fluxes that have good properties. We have thus constructed an ALE flux too.

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 stability and convergence of the scheme, because the projection step, which mixes deterministic and random averaging, is rather untypical.

¹Actually the proof has to be adapted, because we deal with two-fluid flows, but it is very similar.

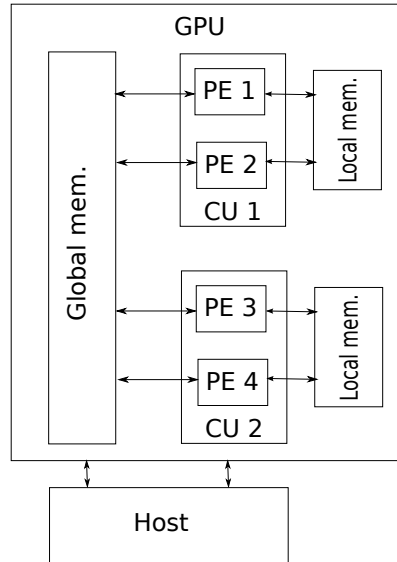


FIGURE 4.1. A (virtual) GPU with 2 Compute Units and 4 Processing Elements

4. GPU AND OPENCL IMPLEMENTATION

4.1. **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 4.1)

- Global memory (typically 1 Gb²)
- Compute units (typically 27).

Each compute unit is made of:

- Processing elements (typically 8).
- Local (or cache) memory (typically 16 kb)

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 to drive the NVIDIA GPUs. OpenCL is a recent set of tools, which allow to program many kind 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 practically available since september 2009 [16]. The specification is managed by the Khronos Group, which is also responsible of the OpenGL API design and evolutions. Virtually,

²the typical values are given for a NVIDIA GeForce GTX 280 GPU

it allows to have 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 [10].

4.2. 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 (3.1), the data are well aligned into the memory and two successive processing elements access two neighbouring memories, which allows to achieve optimal memory bandwidth (coalescing access). For the y -direction step (3.3), 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, described for instance in [19].

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 time step for each cell i (see (3.26)). If the local time step is lower 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 4.1). 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 reduction algorithm for computing the time step [4], but it is a little bit more complicated and in practice our 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 the 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.
- we employ a subdomain strategy in order to retain as much data as possible into the local cache memory of the work-group. When the rows are split, a covering of two cells is then necessary between the subdomain for ensuring the correctness at the boundary values. Indeed, the local cache memory is not shared across work-groups.
- 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 access are optimal.
- we transpose again the grid.

In Table 1, we compare our OpenCL code when it is run on one core or four cores of a multicore CPU, or on recent GPU. We observe spectacular speedups for the GPU simulations compared to the one-core simulation. The test case corresponds to the computation of 300 time steps of the algorithm on a 1024x512 grid.

5. NUMERICAL RESULTS

5.1. One-dimensional results. Firstly, we present some numerical results on the one dimensional Euler equations (3.1) where we do not take into account the y -velocity, it means that

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

Quantities	Left	Right
ρ ($kg.m^{-3}$)	10	1
u ($m.s^{-1}$)	50	50
p (Pa)	1.1×10^5	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 to illustrate the non convergence of the averaging projection.

$v = 0$ at any time. We compare different choices of projections for the lagrangian approach (see condition (3.7)). A classical choice is to use the averaging projection (3.9), the advantage of this projection is to obtain a conservative scheme but the problem is that the fraction of mass φ will be diffused. Indeed, the fourth equation of system (3.1) gives us the transport equation

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

then if at initial time we have $\varphi \in 0, 1$, theoreticaly we have $\varphi = 0$ or $\varphi = 1$ at any time. With the averaging projection this is no more the case numerically and we have then to define mixture parameters law. In [20], it is given by

$$\begin{aligned} \frac{1}{\gamma(\varphi) - 1} &= \varphi \frac{1}{\gamma_2 - 1} + (1 - \varphi) \frac{1}{\gamma_1 - 1}, \\ \frac{\gamma(\varphi)\pi(\varphi)}{\gamma(\varphi) - 1} &= \varphi \frac{\gamma_2 \pi_2}{\gamma_2 - 1} + (1 - \varphi) \frac{\gamma_1 \pi_1}{\gamma_1 - 1}, \end{aligned}$$

where (γ_1, π_1) and (γ_2, π_2) correspond respectively to the pure liquid phase $\varphi = 0$ and the pure gas phase $\varphi = 1$ at the initial time. The resulting scheme has a very poor precision: for instance, if we consider the Riemann problem where the interface between the two fluids is located at time $t_0 = 0$ s at position $x = 1$ m with the left and right states recorded in Table 2, we observe pressure oscillations for the numerical solution (see figure 5.1).

Another approach is to consider the Glimm projection (3.10) in every cell. The resulting scheme is not conservative and in some case it does not converge. For instance, if we consider the Riemann problem where the interface between the two fluids is located at time $t_0 = 0$ s at position $x = 0$ m. Left and right initial states are given in Table 3. A typical plot is given in Figure 5.2, where we compare the exact and the approximated densities at time $t = 0.5$ s.

We also provide in Figure 5.3 a comparison of the two lagrangian schemes (mixed and averaging projection schemes) and the ALE scheme for the densities for a uniform mesh of 500 points. 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. The graph obtained with the mixed projection and with the ALE scheme seems to be superposed. In order to determine the less diffusive scheme we perform a convergence study. In Figure 5.4,

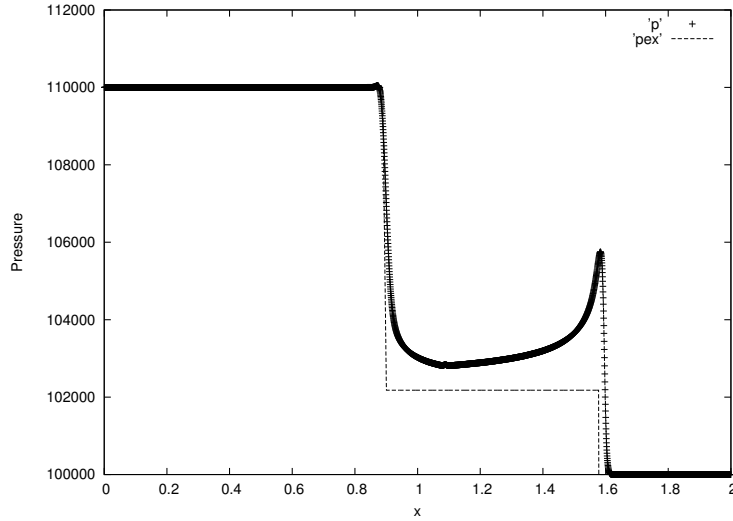


FIGURE 5.1. Oscillations on the pressure with the averaging projection for initial parameters recorded in Table 2.

Quantities	Left	Right
$\rho (kg.m^{-3})$	3.488	1
$u (m.s^{-1})$	1.13	-1
p (Pa)	23.33	2
φ	1	0
γ	2	1.4
π	7	0

TABLE 3. Initial left and right states of the Riemann problem to illustrate the non convergence of the Glimm projection.

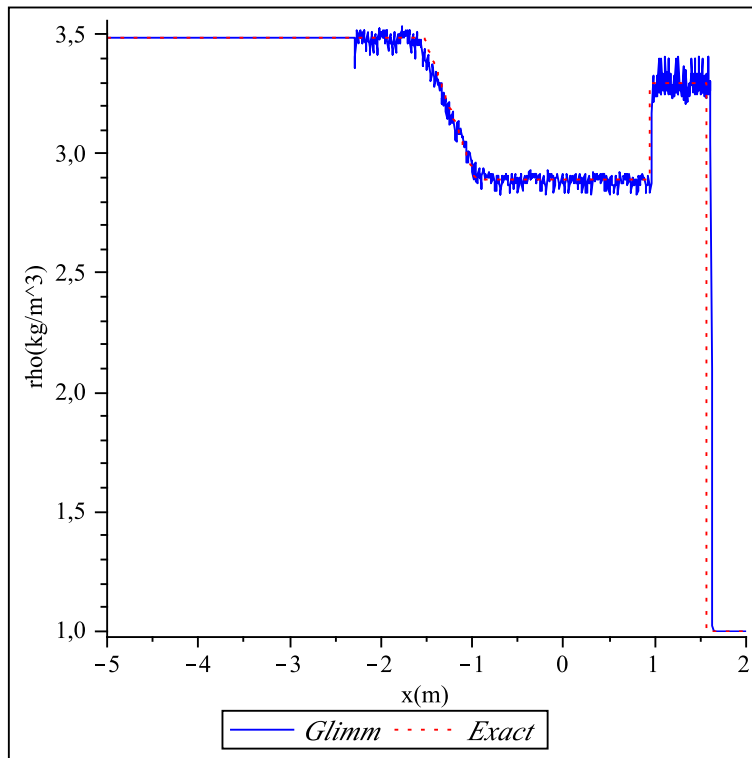


FIGURE 5.2. Non convergence of the Glimm projection for initial parameters recorded in Table 3.

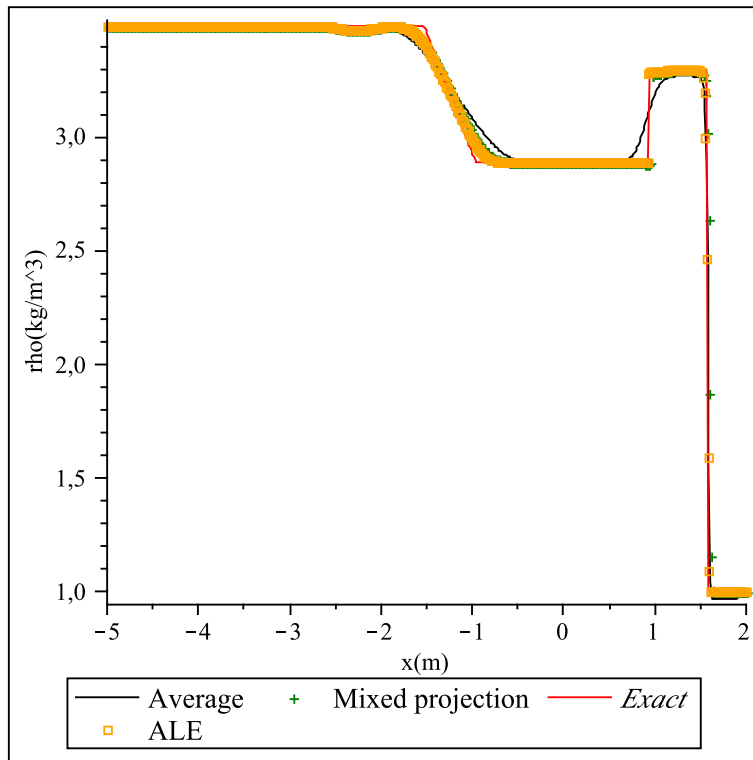


FIGURE 5.3. Comparison of the mixed and averaging projection schemes for initial parameters recorded in Table 3.

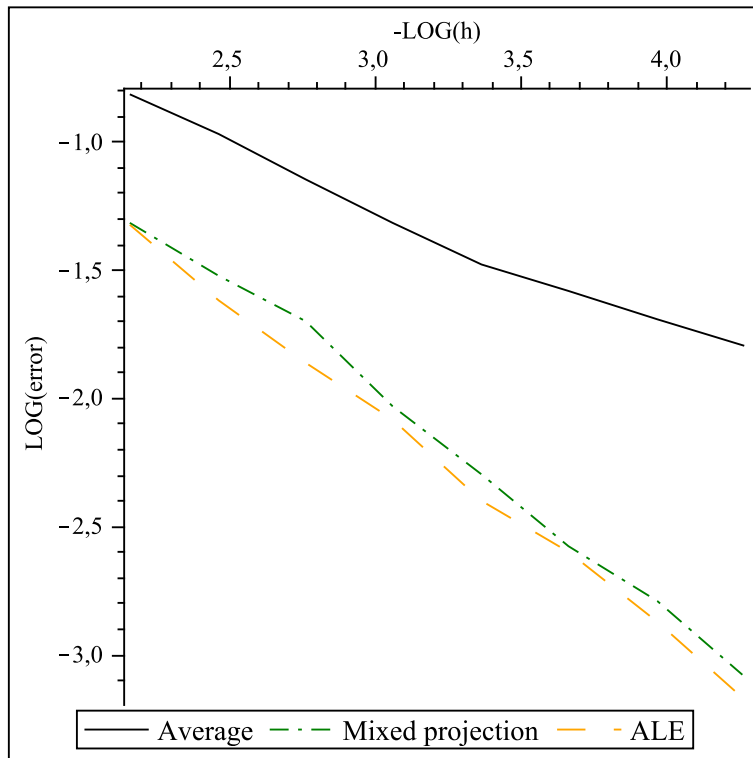


FIGURE 5.4. Convergence study in L^1 norm for initial parameters recorded in Table 3.

we compare the convergence in L^1 norm of the different schemes: the ALE scheme is a little bit less diffuse than the mixed projection scheme.

The convergence rate for the averaging projection is approximately 0.5, while it is 0.8 for the mixed projection and the ALE scheme.

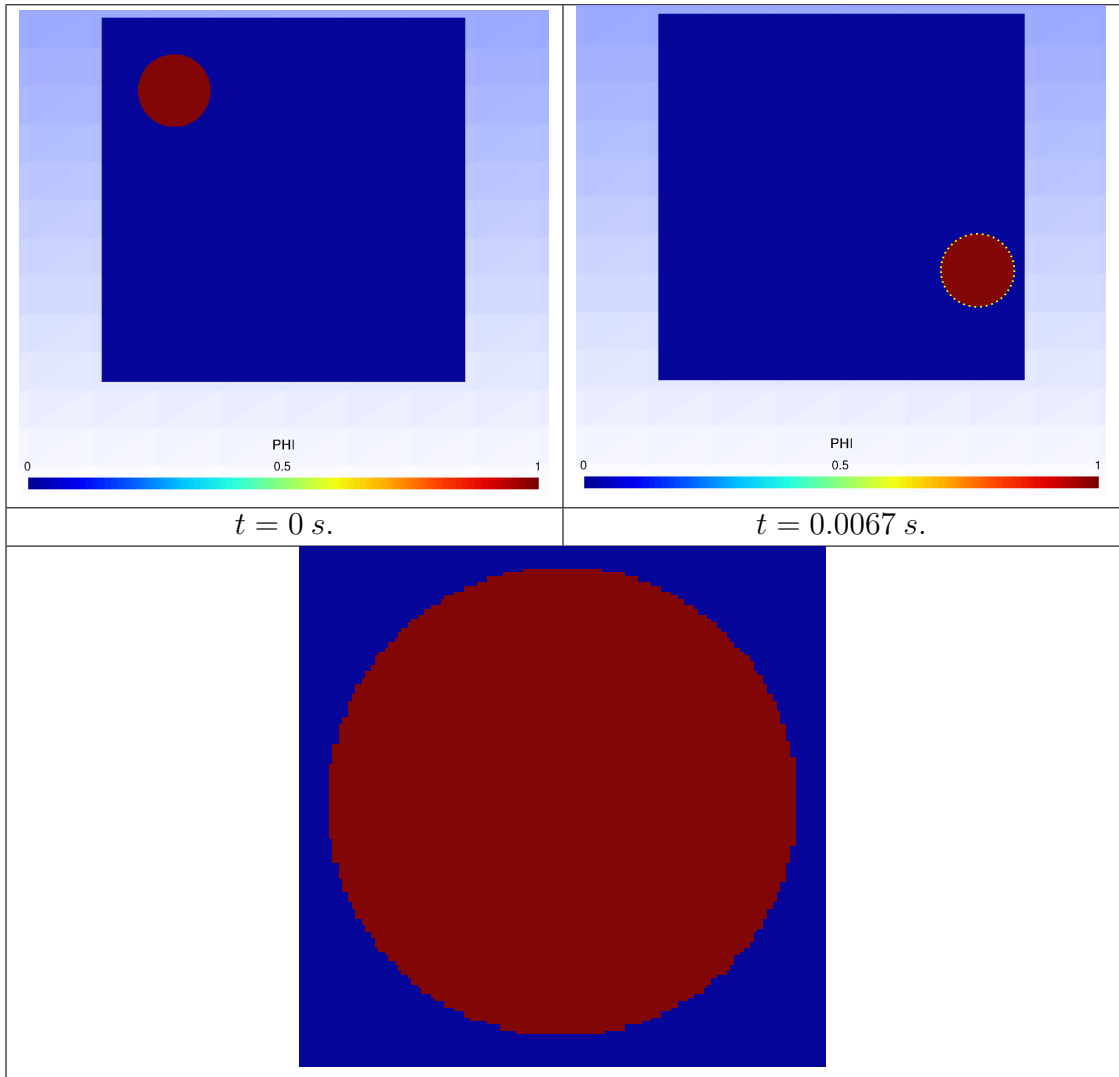


FIGURE 5.5. Initial place of the bubble on top left, final place of the bubble on the top right and zoom on the bubble interface on the bottom.

5.2. Pure convection. Now, we consider two-dimensional numerical tests for which we use the Strang dimensional splitting approach. The first test is to consider the convection of a spherical bubble of gas in a liquid phase. At time $t = 0$ the bubble is in the left top corner (see Figure 5.5).

The initial parameters are presented in Table 4, the bubble moves to the bottom right corner. The results was performed with the ALE scheme and a uniform mesh of 512×512 points on the domain $[0, 1] \times [0, 1]$. On this test case without pressure waves, the results are the same with the mixed projection scheme. The radius of the bubble is 0.1 m and the final time of computation is $t_{final} = 0.0067 \text{ s}$. For the boundary conditions, we impose the initial data at every time. As our algorithms do not diffuse the fraction of mass of gas φ , in fact $\varphi = 1$ in the bubble and $\varphi = 0$ outside the bubble at every time, we choose to plot φ to localize the bubble interface. We see in Figure 5.5 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 5.5 shows that the iterface bubble is not smooth. This lack of smoothness is of course due to the pseudo-random nature of the Glimm projection.

5.3. Test of Zalesak. Now, we consider another classical two dimensional numerical test. Firstly, we want to test dimensional splitting coupled with the Glimm projection. It is the test proposed by Zalesak [24] and consists of a computing the rotation of a complex solid shape. In

Quantities	Inside the bubble	Outside the bubble
ρ ($kg.m^{-3}$)	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. Inital datas for the test of a convected bubble.

fact, we just solve the equation

$$(5.1) \quad \partial_t \rho + \partial_x(\rho u) + \partial_y(\rho v) = 0,$$

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, then we just precise the numerical scheme used to solve

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

In fact, we use the ALE numerical scheme

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

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

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

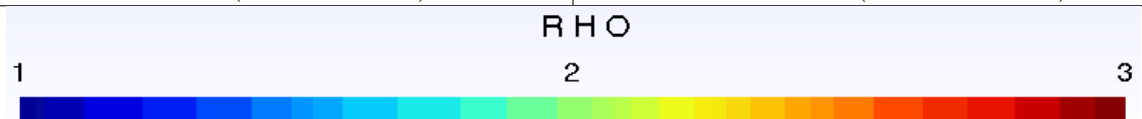
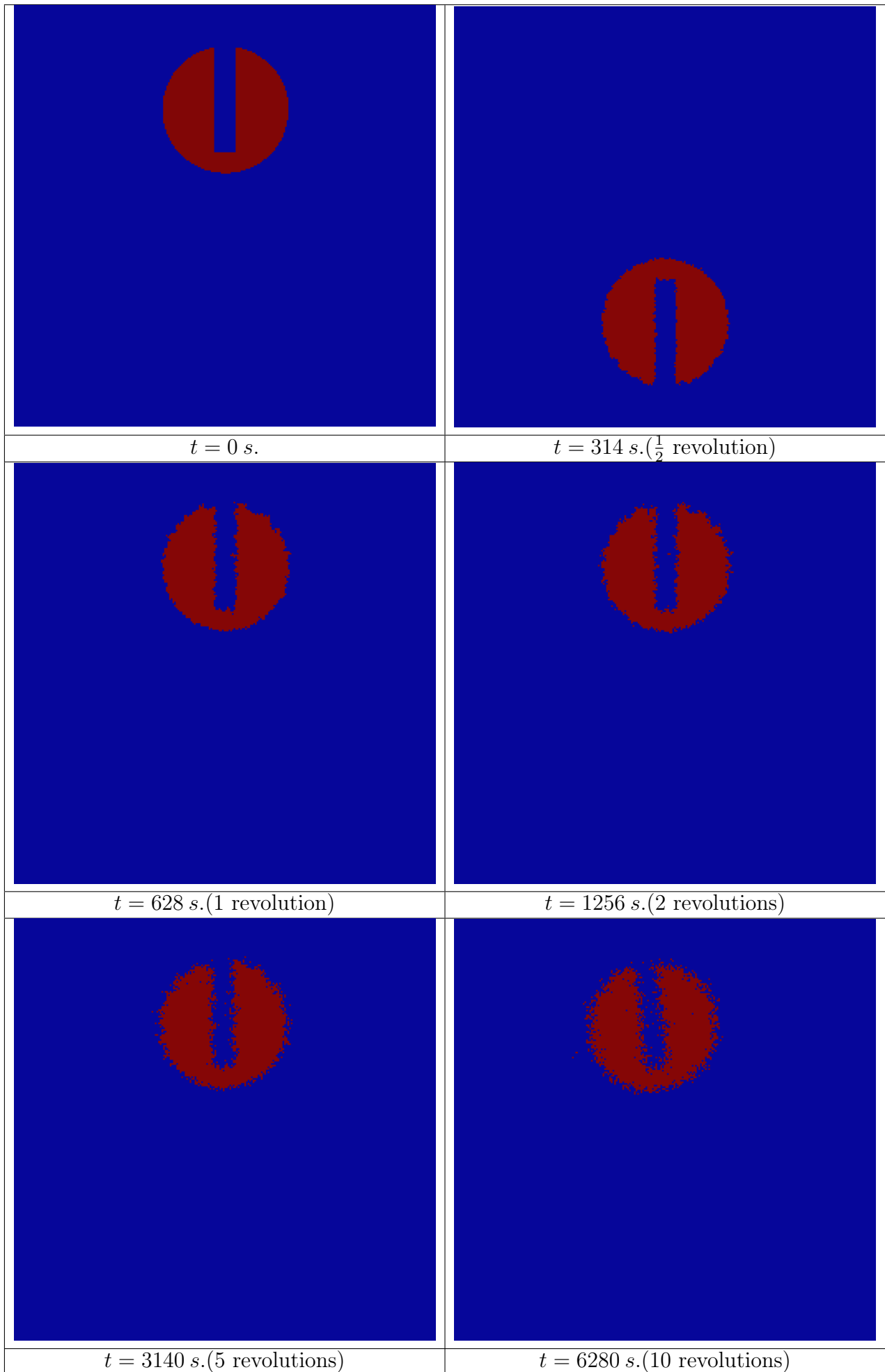
The domain is $[0, 1] \times [0, 1]$ and the solid considered is a cylinder of radius $0.15 m$, through which a slot has been cut of width $0.05m$. We assume that the density inside the cuted cylinder is $\rho = 3$ and outside we have $\rho = 1$. The rotational speed is chosen such that after $628 s$ the cylinder will effect one complete revolution about the central point, i.e. $\Omega = \frac{2\pi}{628} rad.s^{-1}$. The results was performed with the ALE scheme and a uniform mesh of 1000×1000 points, the results were the same with the mixed projection scheme. The time step is chosen such that the quantity $h_i^{n+1,-}$ stay positive, we choose

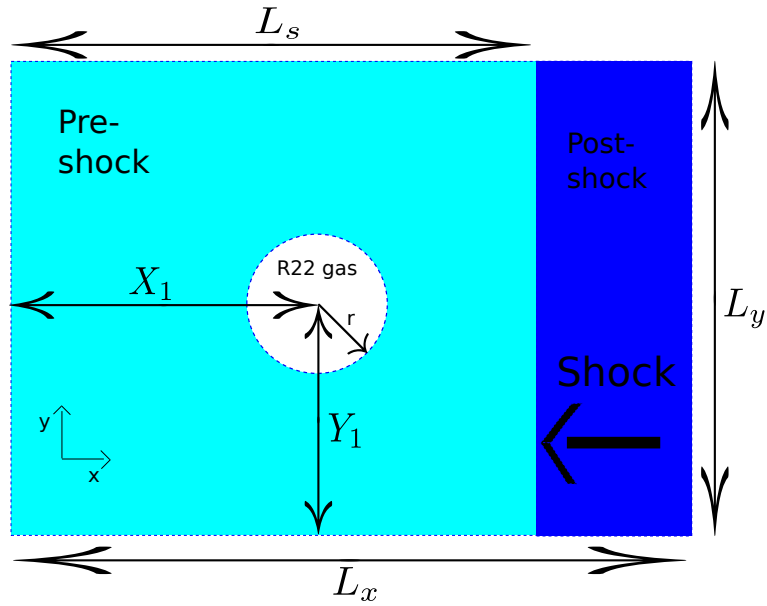
$$\Delta t \leq \frac{1}{2} \frac{\Delta x}{\max_i \sqrt{u_i^2 + v_i^2}}.$$

Remark: As we just consider the equation (5.1), the system is hyperbolic with only one wave speed $\sigma = u$ then the CFL condition is less restrictive than for system (2.1).

In Figure 5.6, we compare the shape of the cuted cylinder after different time. Features to be compared are the filling-in of the gap, erosion of the ‘‘bridge,’’ and the relative sharpness of the profiles decking the front surface of the cylinder. 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 is not so badly solved, considering the simplicity of the Glimm approach, even if after 2 revolutions we observe that there are some part of the solid in the bridge.

5.4. Shock-droplet interaction. We now present a two-dimensional test that consists in simulating the impact of a Mach 1.22 shock travelling though air onto a cylinder of *R22* gas. The shock speed is $\sigma = 415 m.s^{-1}$. This test aims at simulating the experiment of [12] and has been considered by several authors such as [17, 21, 15]. The initial conditions are depicted in Figure 5.7: a cylinder of *R22* is surrounded by air within a $L_x \times L_y$ rectangular domain. At $t = 0$, the cylinder is at rest and its center is located at (X_1, Y_1) . We denote by r the initial




 FIGURE 5.7. Air-*R22* shock/cylinder interaction test. Description of the initial conditions.

Quantities	Air (post-shock)	Air (pre-shock)	<i>R22</i>
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.

radius of the cylinder. The planar shock is initially located at $x = L_s$ and moves from right to left towards the cylinder. 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 modelled 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 5.8 and Figure 5.9 display the evolution of the cylinder shape obtained with the ALE scheme and the experience of Hass and Sturtevant [12]. We can also compare our results to the computations of Kokh and Lagoutiere [15] because we consider the same test-case with the same initial datas. The profiles are obtained thank 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 of the bulk is quite similar to the experimental results. The shape of the two vortices is not exactly the same on the top and the bottom, it comes from the random part of the Glimm projection. In fact, the Glimm projection in the y -direction use the velocity v and this velocity field is symmetric with respect to the axis (O, x) where O is the center of the domain.

5.5. Shock-bubble. We perform another shock/interface interaction test proposed in [20, 15] that involves a gas bubble surrounded by a liquid. The geometry of the initial condition is

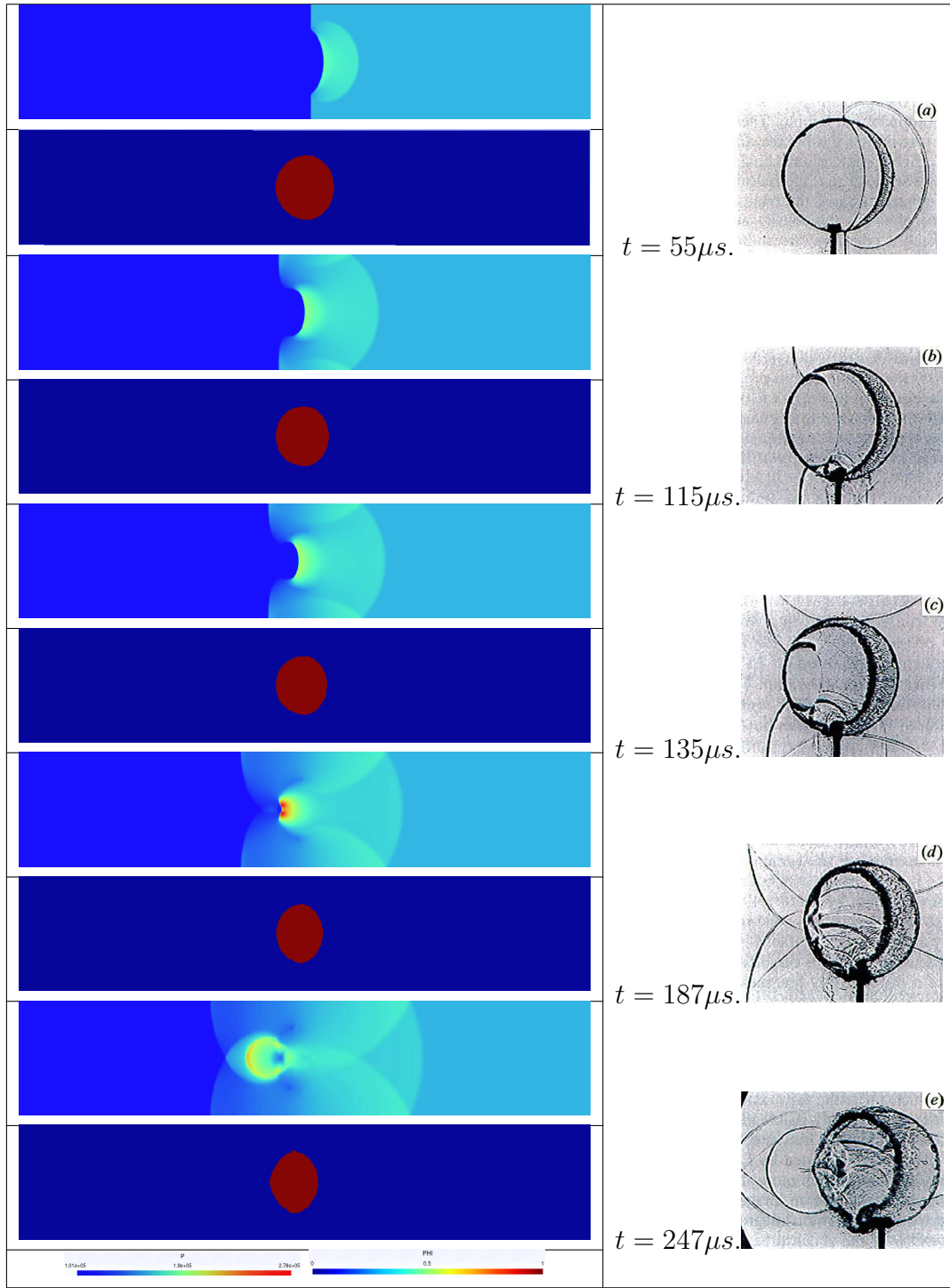


FIGURE 5.8. Air-R22 shock cylinder interaction test. Pressure field and interface on the left; experience of Hass and Sturtevant [12] on the right.

depicted in Figure 5.7 with the following parameters values:

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

The gas within the bubble is governed by a perfect gas law while the liquid is modelled with the stiffened gas law. The EOS parameters and initial states are given in Table 6.

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. The speed of the shock is $\sigma = 10008 \text{ m.s}^{-1}$.

Figure 5.11 and 5.12 display the mapping of respectively the density and the pressure at several instants. For the graph of pressure, as the maximum pressure increase with the time,

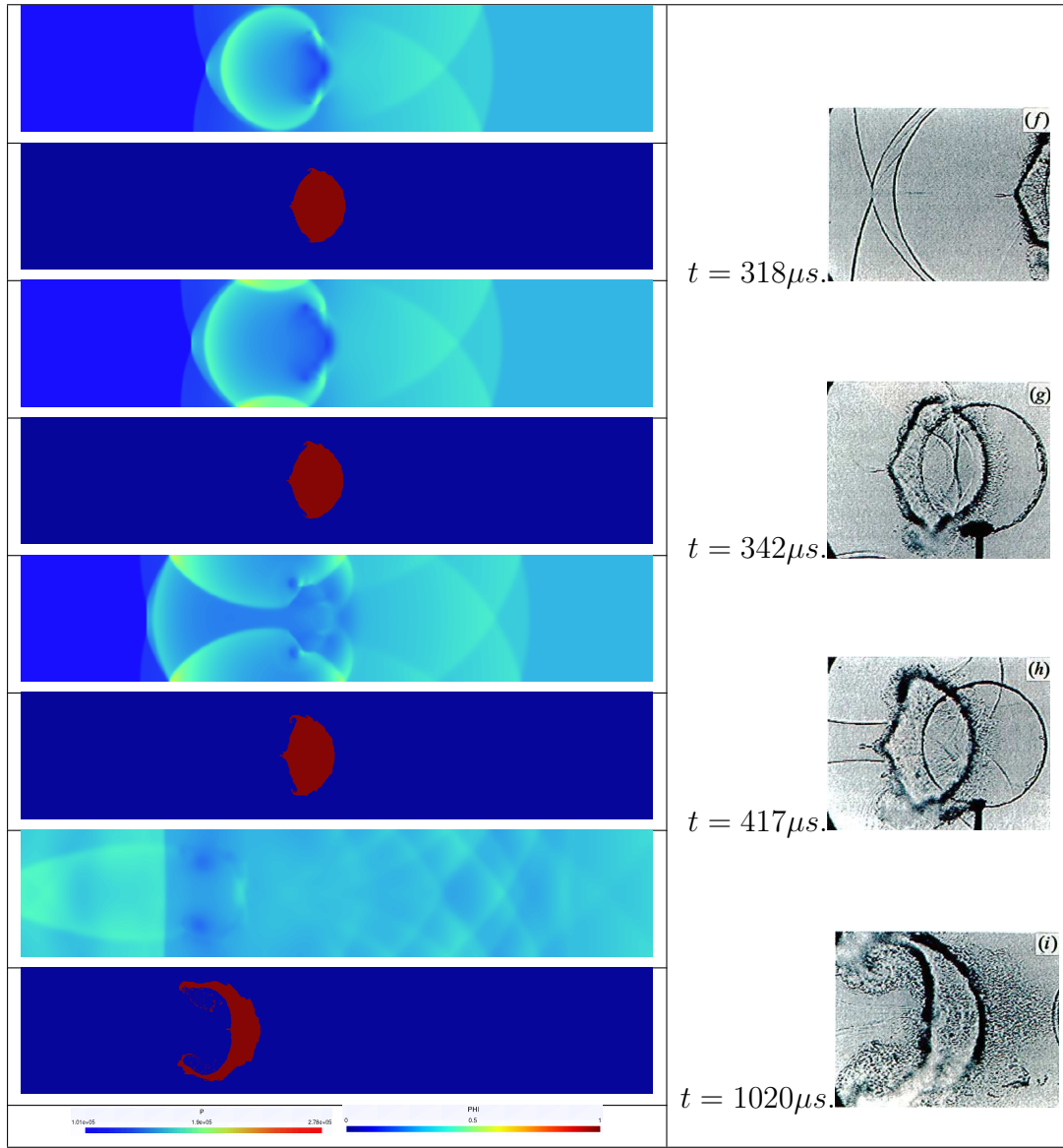


FIGURE 5.9. Air-R22 shock cylinder interaction test. Pressure field and interface on the left; experience of Hass and Sturtevant [12] on the right.

Quantities	Liquid (post-shock)	Liquid (pre-shock)	Gas
ρ ($kg.m^{-3}$)	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.

we can not use the same scale to plot all the graph: the first two graph are plot with different scales from the last two. As the difference of density inside and outside the bubble is huge, we can not see the shock position, but we can easily see the interface bubble by observing the density field. One can clearly see the important variation 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.

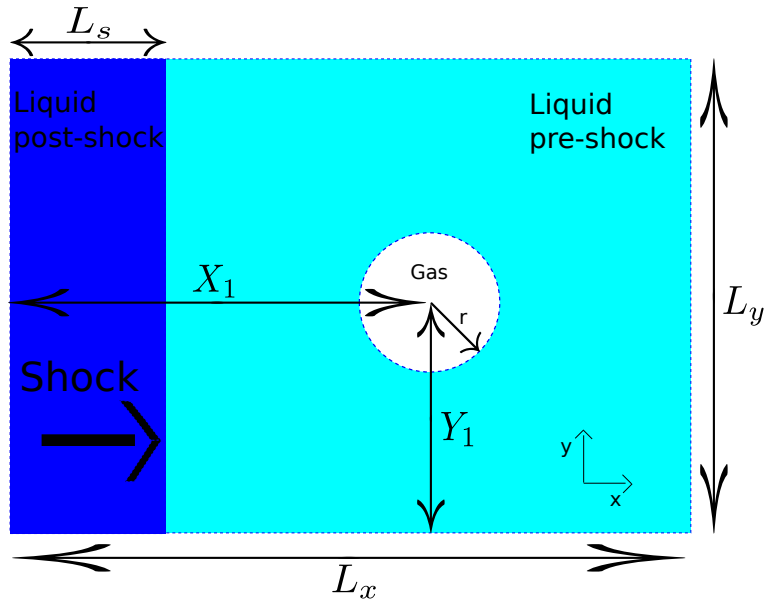


FIGURE 5.10. Liquid-gas shock/bubble interaction test. Description of the initial conditions.

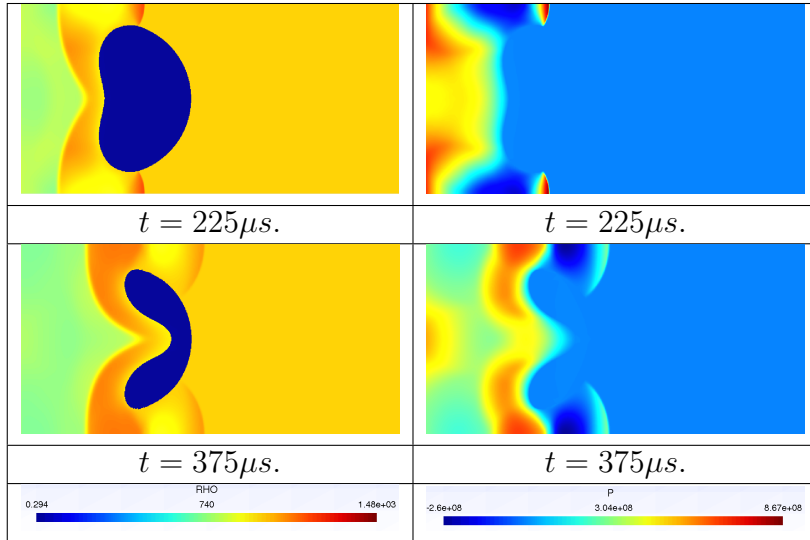


FIGURE 5.11. Liquid-gas/bubble interaction test. Density on the left and pressure field on the right.

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

6. CONCLUSION

We have proposed a new method for computing 2D 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 properties of the resulting scheme are very interesting :

- it preserves constant velocity and pressure states;
- the mass fraction is not diffused at all;
- it allows vacuum.

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

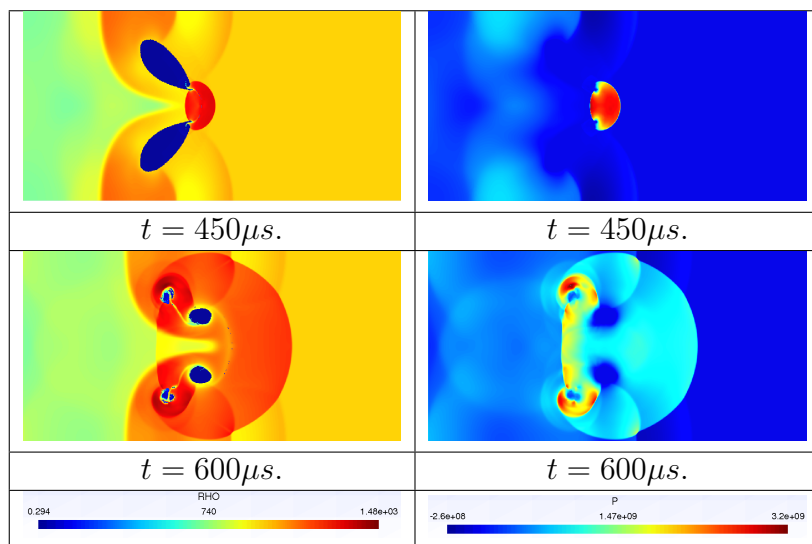


FIGURE 5.12. Liquid-gas/bubble interaction test. Density on the left and pressure field on the right.

to a standard CPU implementation, we observed that the GPU computations are more than hundred times faster.

REFERENCES

- [1] Abgrall, Rémi 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
- [3] Thomas Barberon, Philippe Helluy, and Sandra Rouy. Practical computation of axisymmetrical multifluid flows. *Int. J. Finite Vol.*, 1(1):34, 2004.
- [4] Belloch, G.E., Scans as Primitive Parallel Operations. *IEEE Transactions on Computers*, C-38(11):1526–1538, November 1989.
- [5] Bouchut, François Nonlinear stability of finite volume methods for hyperbolic conservation laws and well-balanced schemes for sources. *Frontiers in Mathematics*. Birkhäuser Verlag, Basel, 2004.
- [6] C. Chalons, F. Coquel. Capturing infinitely sharp discrete shock profiles with the Godunov scheme. *Hyperbolic problems: theory, numerics, applications*, 363–370, Springer, Berlin, 2008.
- [7] Coquel, Frédéric; Perthame, Benoît Relaxation of energy and approximate Riemann solvers for general pressure laws in fluid dynamics. *SIAM J. Numer. Anal.* 35 (1998), no. 6, 2223–2249
- [8] Thierry Gallouët, Jean-Marc Hérard, and Nicolas Seguin. A hybrid scheme to compute contact discontinuities in one-dimensional Euler systems. *M2AN Math. Model. Numer. Anal.*, 36(6):1133–1159 (2003), 2002.
- [9] C. Chalons, P. Goatin. Transport-equilibrium schemes for computing contact discontinuities in traffic flow modeling. *Commun. Math. Sci.*, 5 (2007), no. 3, 533–551.
- [10] Crestetto A., Helluy P. An OpenCL tutorial. 2010. <http://www-irma.u-strasbg.fr/~helluy/OPENCL/tut-openc1.html>
- [11] 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.
- [12] J. F. Hass and B. Sturtevant. Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities. *J. Fluid Mechanics*, 181, 41–71, 1997.
- [13] Mathieu Bachmann, Philippe Helluy, Jonathan Jung, Hélène Mathis, Siegfried Müller. Random sampling remap for compressible two-phase flows, 2011, submitted. Preprint at <http://hal.archives-ouvertes.fr/hal-00546919>
- [14] Karni, Smadar Multicomponent flow calculations by a consistent primitive algorithm. *J. Comput. Phys.* 112 (1994), no. 1, 31–43.
- [15] 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.
- [16] Khronos Group. OpenCL online documentation. <http://www.khronos.org/openc1/>
- [17] J. J. Quirk and S. Karni. On the dynamics of a shock-bubble interaction. *J. Fluid Mechanics*, 318, 129–163, 1996.

- [18] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43(2) :357–372, 1981.
- [19] Ruetsch, G., Micikevicius, P. Optimizing Matrix Transpose in CUDA. *NVIDIA GPU Computing SDK*, 1 – 24. 2009.
- [20] R. Saurel, R. Abgrall. A simple method for compressible multifluid flows. *SIAM J. Sci. Comput.*, 21 (3), 1115-1145, 1999.
- [21] K. M. Shyue. A wave-propagation based volume tracking method for compressible multicomponent flow in two space dimensions. *J. Comput. Phys.*, 215, 219-244, 2006.
- [22] Strang, Gilbert On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.* 5 1968 506–517.
- [23] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics*, 2nd edition. Springer, 1999
- [24] Fully Multidimensional Flux-Corrected Algorithms for Fluids. *J. Computational Physics*, 31, 335-362, 1969.

IRMA, UNIVERSITÉ DE STRASBOURG, 7 RUE DESCARTES, 67000 STRASBOURG, FRANCE, PHILIPPE.HELLUY@MATH.UNISTRA.FR
JONATHAN.JUNG@MATH.UNISTRA.FR