Numerical simulation of two-fluid flow on multicores accelerator

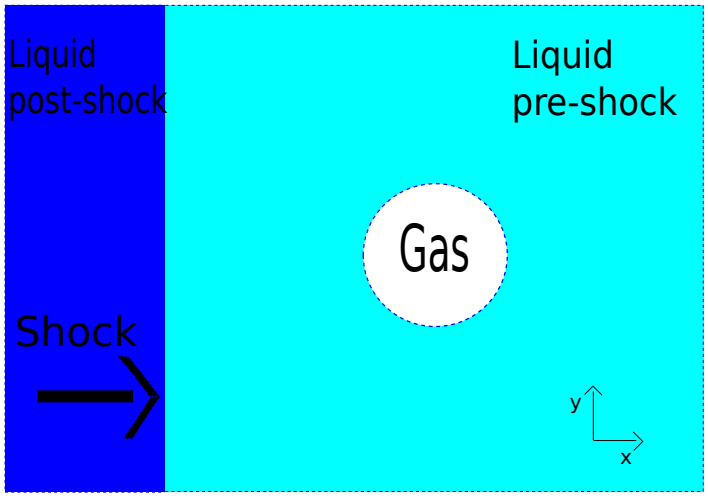
Jonathan Jung (LJLL, Paris VI)

Collaborators : Philippe Helluy (IRMA, Strasbourg)
Jean-Marc Hérard (EDF, Chatou)

LRC Manon, January 21th, 2014

Example of computation

Study case: a shock wave in a liquid interacting with a gas bubble.



Properties of the flow:

- two immiscible fluids,
- the gas and the water are compressible.

Aim: develop a robust and efficient numerical scheme. Adaptation to GPU cluster.

Model

- We consider the 2D compressible Euler equations:

$$\begin{aligned}
\partial_t(\rho) + \partial_x(\rho u) + \partial_y(\rho v) &= 0,\\
\partial_t(\rho u) + \partial_x(\rho u^2 + p) + \partial_y(\rho uv) &= 0,\\
\partial_t(\rho v) + \partial_x(\rho uv) + \partial_y(\rho v^2 + p) &= 0,\\
\partial_t(\rho E) + \partial_x\big((\rho E + p)u\big) + \partial_y\big((\rho E + p)v\big) &= 0,\\
\partial_t(\rho \varphi) + \partial_x(\rho u \varphi) + \partial_y(\rho v \varphi) &= 0,
\end{aligned}$$

  where ρ is the density, (u, v) the velocity vector, E the total energy and φ the gas mass fraction.
- The liquid-gas interface is located at the discontinuities of the gas fraction φ. In the liquid, we have $\varphi = 0$ and in the gas $\varphi = 1$.
- We have 5 equations for 6 unknowns.

Equation of states

- Galilean invariance of the Euler equations:

$$p = p(\tau, e, \varphi),$$

where

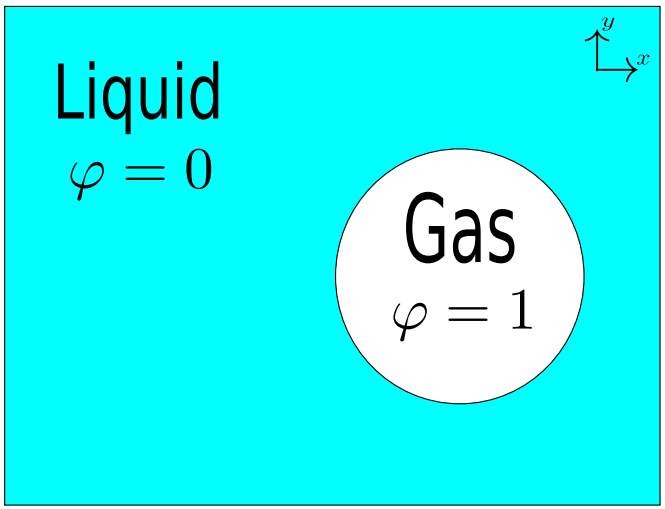
$$\tau = \frac{1}{\rho},$$
$$e = E - \frac{u^2 + v^2}{2}.$$

- The pressure p can be given by a stiffened gas law whose coefficients depend on the gas fraction φ

$$p(\tau, e, \varphi) = \big(\gamma(\varphi) - 1\big)\frac{e}{\tau} - \gamma(\varphi)p_\infty(\varphi).$$

Mixture

- At initial time,



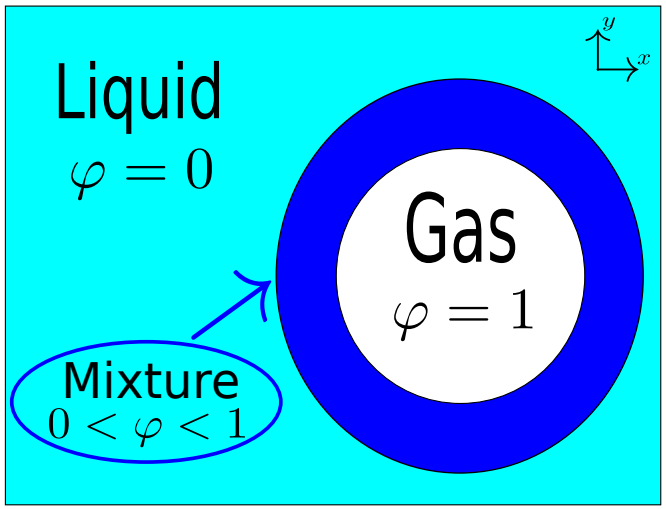
- Theoretically,

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

$\rightarrow \varphi = 0$ or $\varphi = 1$ at any time.

- Numerically, we have diffusion

$$0 < \varphi < 1.$$



- In the mixture, [SA99]

$$\frac{1}{\gamma(\varphi) - 1} = \varphi \frac{1}{\gamma_{gas} - 1} + (1 - \varphi) \frac{1}{\gamma_{liq} - 1},$$

$$\frac{\gamma(\varphi) p_\infty(\varphi)}{\gamma(\varphi) - 1} = \varphi \frac{\gamma_{gas} p_{\infty,gas}}{\gamma_{gas} - 1} + (1 - \varphi) \frac{\gamma_{liq} p_{\infty,liq}}{\gamma_{liq} - 1}.$$

Conservative form

The system admits a conservative form:

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

where

$$\begin{aligned} W &= (\rho, \rho u, \rho v, \rho E, \rho \varphi)^T, \\ F(W) &= \left(\rho u, \rho u^2 + p, \rho uv, (\rho E + p)u, \rho \varphi u\right)^T, \\ G(W) &= \left(\rho v, \rho uv, \rho v^2 + p, (\rho E + p)v, \rho \varphi v\right)^T. \end{aligned}$$

Hyperbolicity of the system

The system is hyperbolic in each direction [GR91]. For each $\mathbf{n} = (n_1, n_2)^T$, the matrix $n_1 D_W F(W) + n_2 D_W G(W)$ is **diagonalizable** with real eigenvalues on the domain

$$\Omega := \left\{ W \in \mathbb{R}^5,\ \rho > 0,\ \varphi \in [0;1],\ p\left(\tau, e, \varphi\right) + p_\infty(\varphi) > 0 \right\},$$

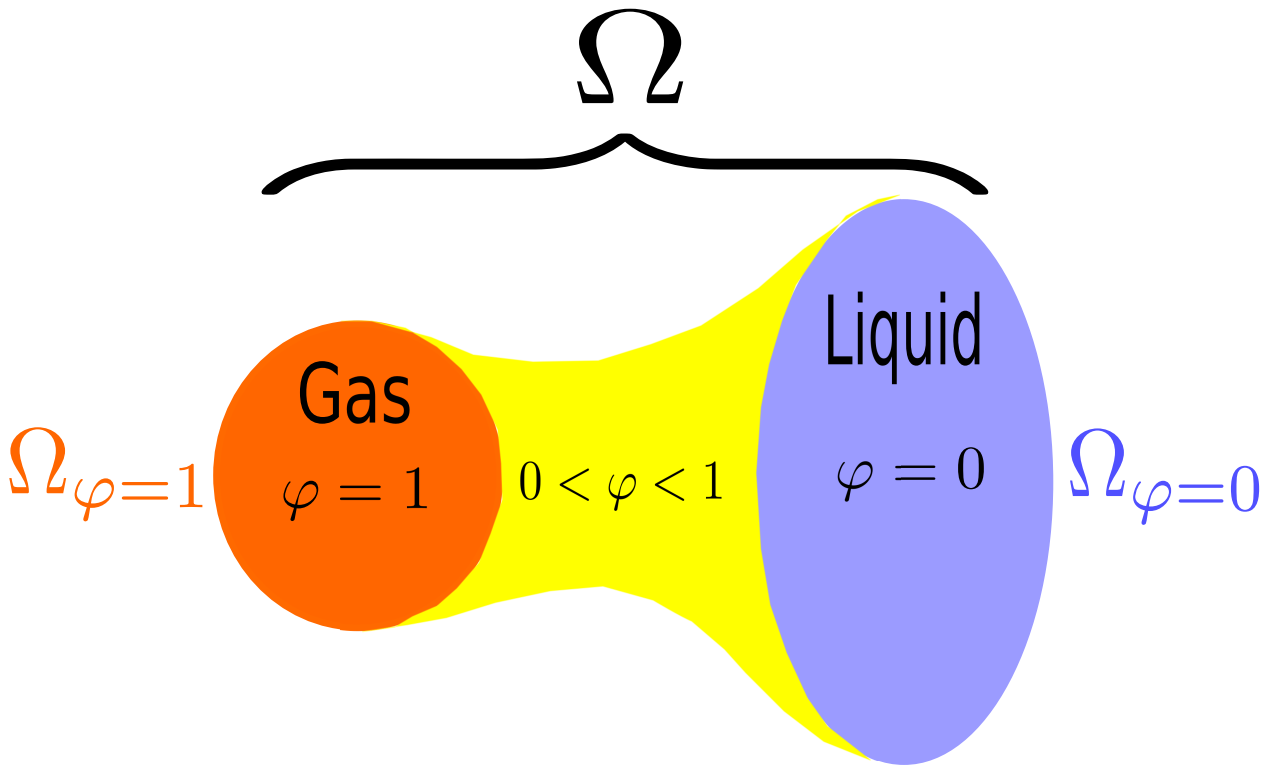
and admits the **eigenvalues**:

$$\lambda_1 = (u,v)^T.\mathbf{n} - c, \qquad \lambda_2 = \lambda_3 = \lambda_4 = (u,v)^T.\mathbf{n}, \qquad \lambda_5 = (u,v)^T.\mathbf{n} + c,$$

where for $W \in \Omega$, the **sound speed** c is given by

$$c(\tau, e, \varphi) = \sqrt{\gamma(\varphi)\tau\big(p\left(\tau, e, \varphi\right) + p_\infty(\varphi)\big)}.$$

Non convexity of the hyperbolic set Ω



- Ω is not a convex set.
- $\Omega_{\varphi=0}$ and $\Omega_{\varphi=1}$, defined by

$$\Omega_{\varphi=\varphi_0} := \left\{ W \in \mathbb{R}^5,\ \rho > 0,\ \varphi = \varphi_0,\ p\left(\tau, e, \varphi_0\right) + p_\infty(\varphi_0) > 0 \right\},$$

are convex sets.

Aim

- For approximating the 2D system, we consider directional splitting (Godunov [G59], Strang [S68]).
  $\rightarrow$ We solve successively the 1D systems

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

- From rotational invariance we only have to construct an approximation of

$$\partial_t W + \partial_x F(W) = 0,\ x \in [a;b],\ t > 0,$$
$$W(x,0) = W_0(x),$$

with some boundary conditions at a and b.

Riemann problem

- Riemann problem

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

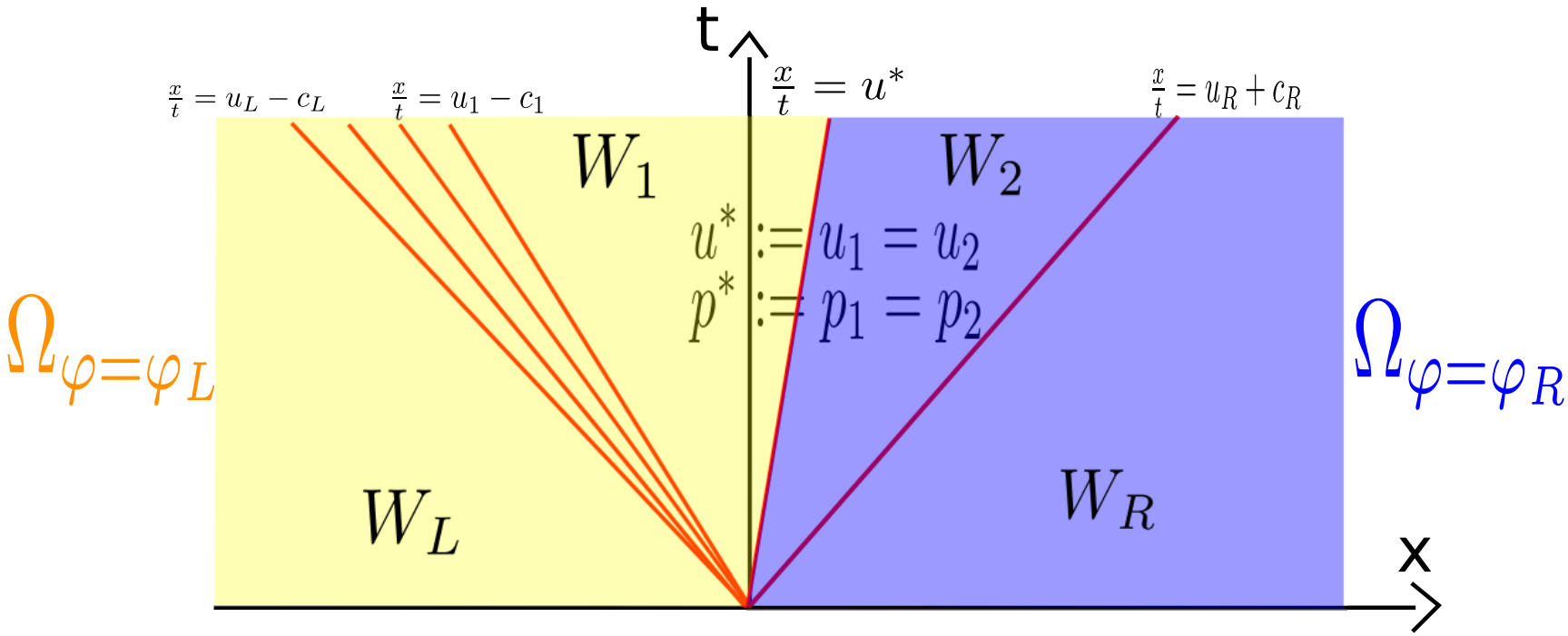
$$W(x,0) = \begin{cases} W_L, & \text{if } x < 0, \\ W_R, & \text{otherwise.} \end{cases}$$

- Even if Ω is not convex, there exists only one, self-similar, global, entropy solution [GR91]

$$W(x,t) = R(W_L, W_R, x/t).$$

- It is entirely defined by 4 constant states W_L, $W_1 \in \Omega_{\varphi=\varphi_L}$, $W_2 \in \Omega_{\varphi=\varphi_R}$ and W_R separated by self-similar waves: rarefaction, shock or contact.

Structure of the solution to the Riemann problem



Example of solution to the Riemann problem $R(W_L, W_R, x/t)$:
a 1-rarefaction, a 2,3,4-contact and a 5-shock.
The exact solution lies in the non convex set $\Omega_{\varphi=\varphi_L} \cup \Omega_{\varphi=\varphi_R}$.

Godunov scheme

- We consider a space step $\Delta x = \frac{b-a}{N}$ and points

$$x_{i-\frac{1}{2}} = a + i \times \Delta x.$$

- The cell (or finite volume) C_i is the interval $]x_{i-\frac{1}{2}}; x_{i+\frac{1}{2}}[$.
- We note x_i the middle of the cell C_i

$$x_i = \frac{x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}}{2}.$$

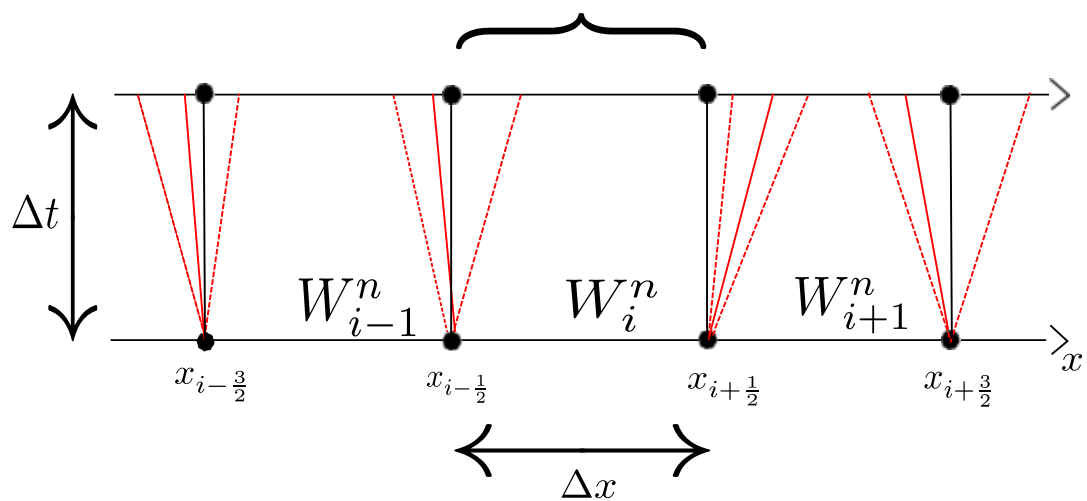
- We start with a piecewise constant approximation of $W(x,0)$ made of cell averages

$$W_i^0 = \frac{1}{\Delta x} \int_{C_i} W(x,0) dx.$$

Time evolution of the Godunov scheme

$$W_i^{n+1} = \frac{1}{\Delta x} \int_{C_i} V(x, \Delta t) dx$$

Averaging step



First step :
We compute the exact solution of

$$\begin{cases} \partial_t V + \partial_x F(V) = 0, \\ V(x,0) = W_i^n, \ x_{i-\frac{1}{2}} < x < x_{i-\frac{1}{2}} \end{cases}$$

$\rightarrow$ if $x_i < x < x_{i+1}$,

$$V(x, \Delta t) = R(W_i^n, W_{i+1}, \frac{\Delta x}{\Delta t}) \in \Omega.$$

Stability issue: Ω is not convex

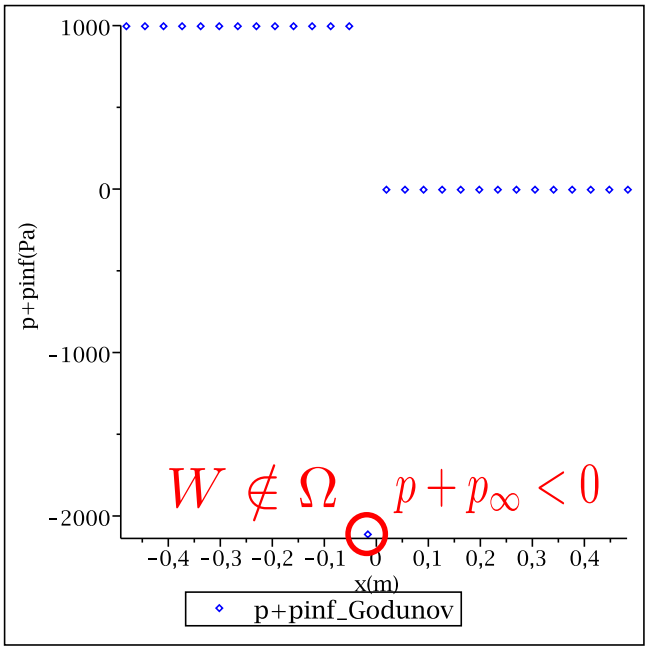
- The exact solution $V(x, \Delta t) \in \Omega$ for all $x \in [a; b]$.
- As Ω is not convex, the averaging

$$W_i^{n+1} = \frac{1}{\Delta x} \int_{C_i} V(x, \Delta t) dx,$$

may produce values outside Ω.

- For example:

Quantities	$x < 0$	$x > 0$
$\rho(kg.m^{-3})$	10	1
$u(m.s^{-1})$	-50	0
$p(Pa)$	1	1
φ	1	0
γ	4.4	3
$p_\infty(Pa)$	1000	0



see also [MHB10].

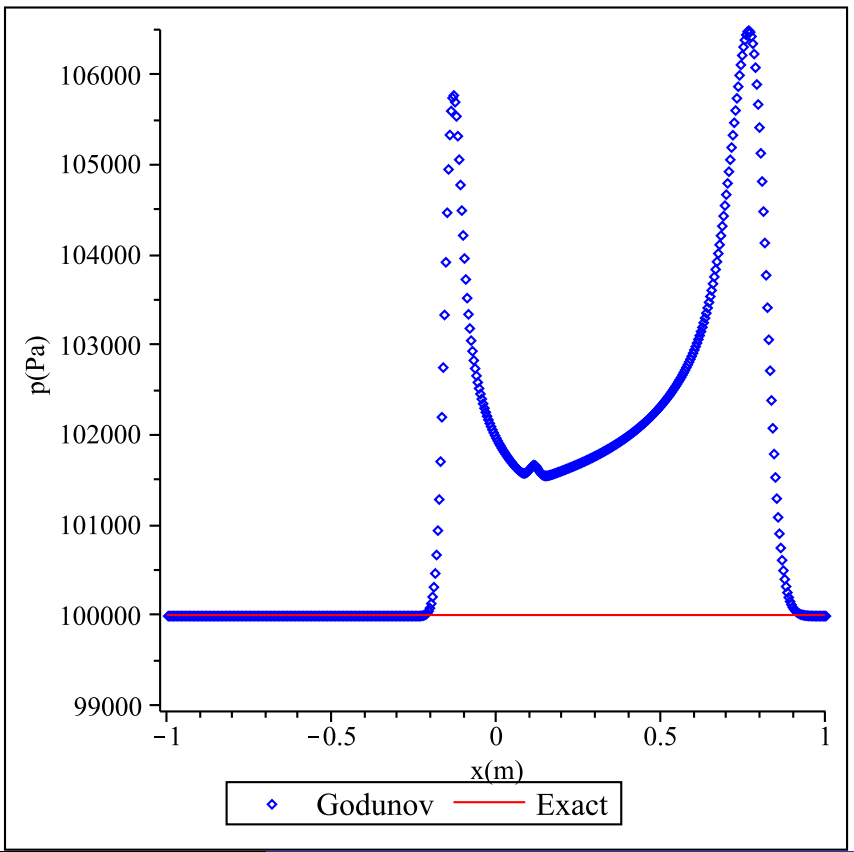
Spurious pressure oscillations at the two-fluid interface

Regardless of the stability problem, we observe velocity-pressure spurious oscillations at the two-fluid interface [A88].

For example, with

Quantities	$x < 0$	$x > 0$
$\rho(kg.m^{-3})$	10	1
$u(m.s^{-1})$	50	50
$p(Pa)$	1e5	1e5
φ	1	0
γ	1.4	1.1
$p_\infty(Pa)$	0	0

we obtain:



Aim of the thesis

- Solve the stability problems due to the non convexity of Ω for a stiffened gas pressure law.
- Solve the problem of the pressure oscillations at the two-fluid interface.
- Extend the method to study the oscillations of a spherical bubble of gas in a liquid phase (well-balanced solver).
  $\rightarrow$ Not presented here.
- Develop a robust and efficient 2D numerical scheme.
- Implement the numerical scheme on Graphics Processing Units (GPU) and on GPU clusters.

How to remove the spurious pressure oscillations?

Previous works on the spurious pressure oscillations issue:

- use a Eulerian approach with a **non conservative flux** at the two-fluid interface (Abgrall [A88], Karni [Ka94], Fedkiw [F99], Saurel-Abgrall [SA99], Abgrall-Karni [AK01], *etc*),
- use a Lagrangian-projection scheme and project the pressure at interface (Barberon-Helluy-Rouy [BHR07], Chalons-Coquel [CC10]). The projection is **non conservative**.

$\rightarrow$ We obtain non conservative schemes.

How to remove the stability problems?

Previous works on the lack of stability caused by the non-convexity of the hyperbolic set:

- Other pressure law (Callen [C85], Croisille [C91], Chanteperdrix, Villedieu and Vila [CVV02], *etc*).
  $\rightarrow$ The proof of convexity is not provided.
- Chalons and Goatin [CG08] proposed a numerical method with a random sampling for computing phase transitions in traffic flow modeling.

How to solve these two issues?

- First approach: construct another pressure law $\tilde{p}$ to obtain a convex domain of hyperbolicity $\tilde{\Omega}$.
  $\rightarrow$ Stable and conservative numerical scheme **but** the spurious pressure oscillations are not removed.

- Second approach: keep a stiffened gas pressure law for the liquid and the gas and use a numerical strategy in order to avoid diffusion on φ and to preserve the non convex domain Ω.
  $\rightarrow$ Stable and conservative numerical scheme **without** spurious pressure oscillations.

First approach: construct an other pressure law $\tilde{p}(\tau, e, \varphi)$

Procedure

- The gas (resp. the liquid) is described by the **massic entropy** $s_1 = s_1(\tau_1, e_1)$ (resp. $s_2 = s_2(\tau_2, e_2)$) linked to the pressure p_i of phase i

$$p_i(\tau_i, e_i) = \frac{\partial_{\tau_i} s_i}{\partial_{e_i} s_i}.$$

- The mixture pressure $\tilde{p}(\tau, e, \varphi)$ is derived from the mixture entropy $s = s(\tau, e, \varphi)$ by the relation

$$\tilde{p}(\tau, e, \varphi) = \frac{\partial_\tau s}{\partial_e s}.$$

$\rightarrow$ We need a procedure for constructing the mixture entropy s from s_1 and s_2.

Mixture entropy and pressure law

- At equilibrium, the immiscible mixture entropy $s(\tau, e, \varphi)$ is given by

$$s(\tau, e, \varphi) = \sup_{0 \leq \alpha, z \leq 1} \left(\varphi s_1 \left(\frac{\alpha}{\varphi} \tau, \frac{z}{\varphi} e \right) + (1 - \varphi) s_2 \left(\frac{1 - \alpha}{1 - \varphi} \tau, \frac{1 - z}{1 - \varphi} e \right) \right).$$

- From $\tilde{p} = \frac{\partial_\tau s}{\partial_e s}$, we deduce the pressure law

$$\tilde{p}(\tau, e, \varphi) = \big(\gamma(\tau, e, \varphi) - 1\big) \frac{e}{\tau} - \gamma(\tau, e, \varphi) p_\infty(\tau, e, \varphi).$$

- In the liquid ($p_{\infty,liq} > 0$, $\varphi = 0$), the pressure law is modified, even if there is no gas. Indeed

$$\tilde{p}(\tau, e, 0) = \begin{cases} 0 & , \text{ if } e \leq \frac{\gamma_{liq} p_{\infty,liq}}{\gamma_{liq} - 1} \tau, \\ (\gamma_{liq} - 1) \frac{e}{\tau} - \gamma_{liq} p_{\infty,liq}, & \text{otherwise.} \end{cases}$$
$$\neq p_{liq}(\tau, e).$$

$\rightarrow$ Negative pressure are replaced by 0.

Properties of the system with the mixture pressure $\tilde{p}$

- With the mixture entropy s, we construct a Lax entropy

$$\tilde{\Omega} \to \mathbb{R},$$

$$W = (\rho, \rho u, \rho v, \rho E, \rho\varphi) \mapsto -\rho s\Big(\frac{1}{\rho}, \frac{2\rho E - (\rho u)^2 + (\rho v)^2}{2\rho}, \varphi\Big),$$

for the system $\partial_t W + \partial_x F(W) = 0$ coupled with $\tilde{p}$.
Here, $\tilde{\Omega}$ is a convex set.

- From Mock theorem, we deduce the hyperbolicity of the system

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

coupled with $\tilde{p}(\tau, e, \varphi)$ on the convex set $\tilde{\Omega}$.

Advantages and drawbacks of $\tilde{p}$

Advantage:

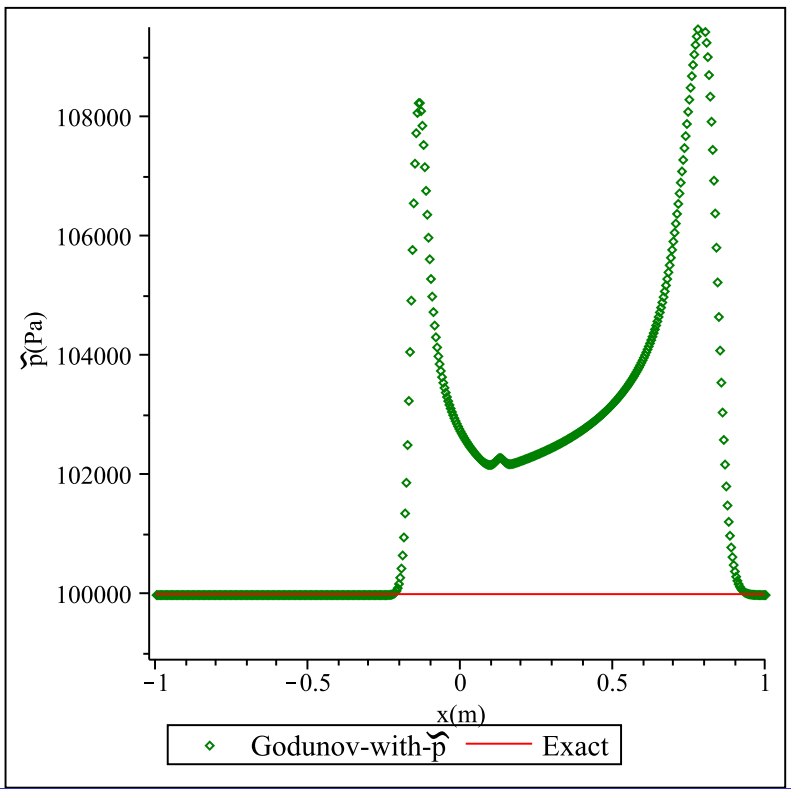
- $\tilde{\Omega}$ is convex
  $\rightarrow$ the Godunov scheme is conservative and stable.

Drawback:

- For some parameters, the liquid degenerates to a pressureless gas.
  $\rightarrow$ Theoretical and numerical difficulties.
- The problem of the spurious oscillations at the two-fluid interface is not solved.

$\rightarrow$ We decide to concentrate on the second idea.

$\longrightarrow$



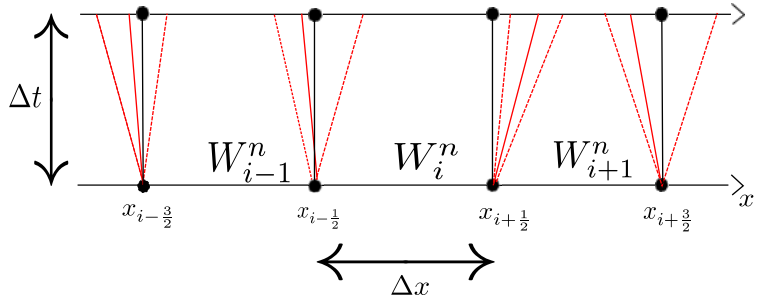
Second approach: construct a numerical method to preserve the non convex domain Ω

Problem of the Godunov scheme

- Now, the gas and the liquid satisfy stiffened gas pressure law again.
- Godunov scheme:

$$W_i^{n+1} = \frac{1}{\Delta x} \int_{C_i} V(x, \Delta t) dx$$

Averaging step



First step :
We compute the exact solution of

$$\begin{cases} \partial_t V + \partial_x F(V) = 0, \\ V(x,0) = W_i^n, \ x_{i-\frac{1}{2}} < x < x_{i-\frac{1}{2}} \end{cases}$$

$\rightarrow$ if $x_i < x < x_{i+1}$,

$$V(x, \Delta t) = R(W_i^n, W_{i+1}, \frac{\Delta x}{\Delta t}) \in \Omega.$$

$\rightarrow V(x, \Delta t)$ lies in the non convex set Ω.

$\rightarrow$ spurious pressure oscillations and lack of stability arise from the averaging step.

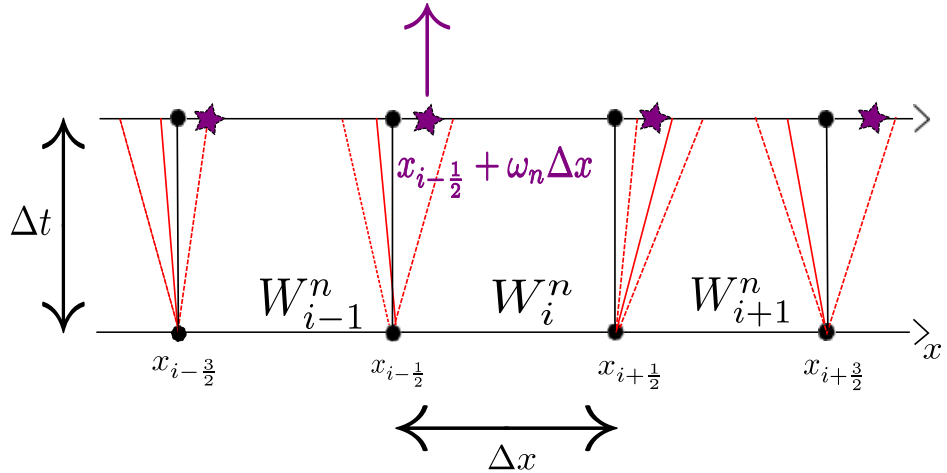
- Conclusion: replace the averaging step.

Glimm scheme [G65]

We take a pseudo-random number $\omega_n \in [0;1[$ and we replace the averaging step of the Godunov scheme.

$$W_i^{n+1} = V(x_{i-\frac{1}{2}} + \omega_n \Delta x, \Delta t)$$

Averaging step



First step :
We compute the exact solution of

$$\begin{cases} \partial_t V + \partial_x F(V) = 0, \\ V(x,0) = W_i^n, \ x_{i-\frac{1}{2}} < x < x_{i-\frac{1}{2}} \end{cases}$$

$\rightarrow$ if $x_i < x < x_{i+1}$,

$$V(x, \Delta t) = R(W_i^n, W_{i+1}, \frac{\Delta x}{\Delta t}) \in \Omega.$$

Advantages and drawbacks of the Glimm scheme

Advantages:

- It does not introduce numerical mixture.
- No spurious pressure oscillations at the two-fluid interface.
- It is **stable**, it preserves the hyperbolic set without diffusion

$$\Omega_{\varphi=0} \cup \Omega_{\varphi=1}.$$

Drawbacks:

- Exact Riemann solver needed.
- Noisy rarefaction waves.
- No extension to higher dimensions (Colella [C78]).

→ We do not follow this approach and introduce the class of the ALE-projection scheme.

ALE-projection scheme

The Arbitrary Lagrangian Eulerian (ALE)-projection scheme includes two steps:

- the ALE-step: we solve the problem on a moving mesh between time t^n and $t^{n+1,-}$.
  - The boundary $x_{i+\frac{1}{2}}$ moves at an arbitrary velocity $\xi^n_{i+\frac{1}{2}}$ between t^n and $t^{n+1,-}$

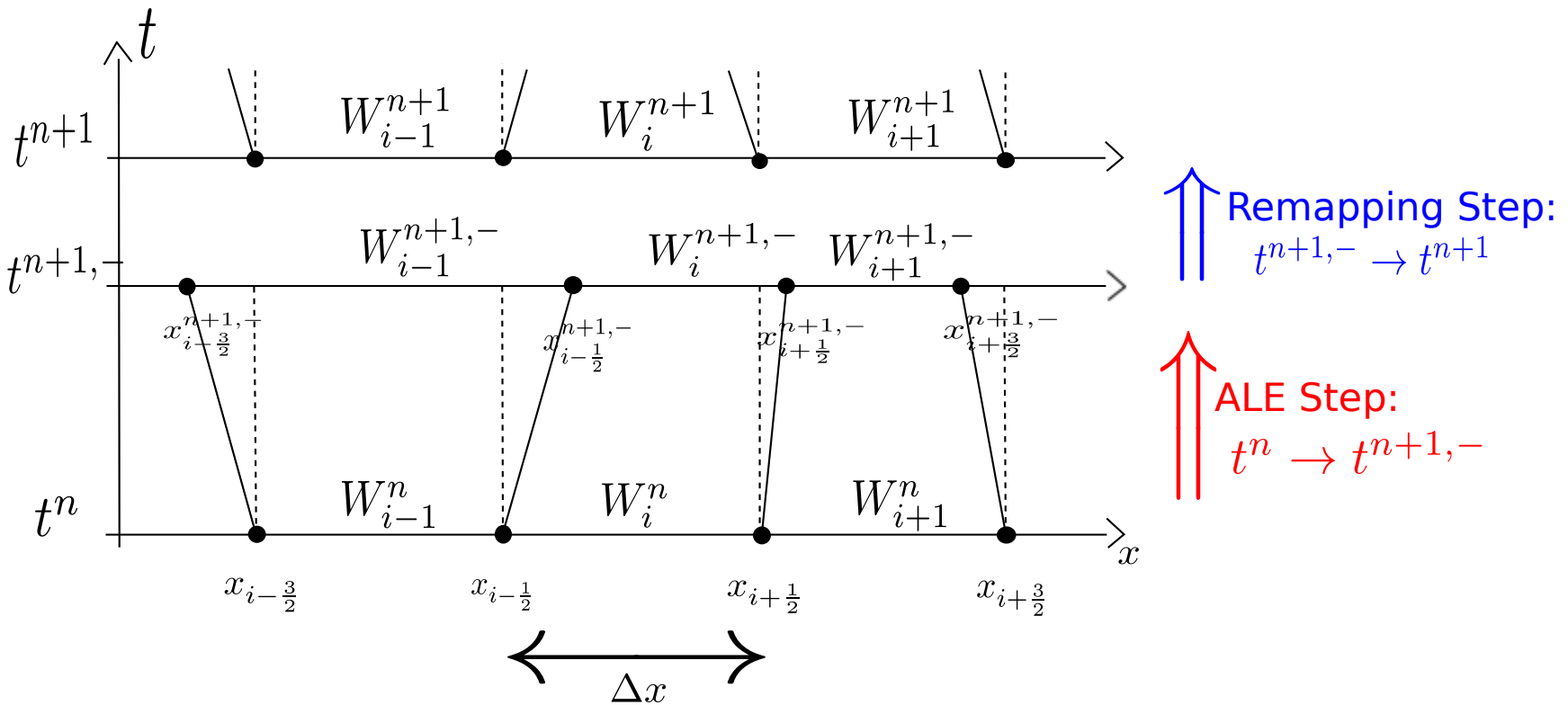
$$x^{n+1,-}_{i+\frac{1}{2}} = x_{i+\frac{1}{2}} + \Delta t\, \xi^n_{i+\frac{1}{2}}.$$

  - We obtain a constant piecewise approximation $W^{n+1,-}$ of $W(\cdot, t_{n+1})$ on cells

$$C^{n+1,-}_i = \left]x^{n+1,-}_{i+\frac{1}{2}}; x^{n+1,-}_{i+\frac{1}{2}}\right[,$$

- the projection step: we project $W^{n+1,-}$ on the original cell C_i.

Structure of ALE-projection scheme



ALE step: expression of $W_i^{n+1,-}$

$W_i^{n+1,-}$ is given by the finite volume scheme

$$\Delta x_i^{n+1,-} W_i^{n+1,-} = \Delta x W_i^n - \Delta t \Big(F(W_i^n, W_{i+1}^n, \xi_{i+\frac{1}{2}}^n) - F(W_{i-1}^n, W_i^n, \xi_{i-\frac{1}{2}}^n) \Big)$$

where Δt satisfies some CFL condition and $F(W_L, W_R, \xi)$ is the numerical flux, given by

$$F(W_L, W_R, \xi) = F\big(R(W_L, W_R, \xi)\big) - \xi R(W_L, W_R, \xi).$$

We have to choose:

- the boundary velocity $\xi_{i+\frac{1}{2}}^n$,
- the projection to go back to the original grid.

Lagrange + Glimm remap

- We choose $\xi^n_{i+\frac{1}{2}} = u^*$ where u^* is the contact discontinuity velocity in the resolution of the Riemann problem $R(W^n_i, W^n_{i+1}, \cdot)$.
- We perform a random sampling

$$W^{n+1}_i = W^{n+1,-}(x_{i-\frac{1}{2}} + \omega_n \Delta x, \Delta t),$$

where $\omega_n \in [0; 1[$ is a pseudo-random number.

- Properties of the scheme:
  - constant velocity-pressure states are preserved,
  - stability of the non convex hyperbolic set $\Omega_{\varphi=0} \cup \Omega_{\varphi=1}$,

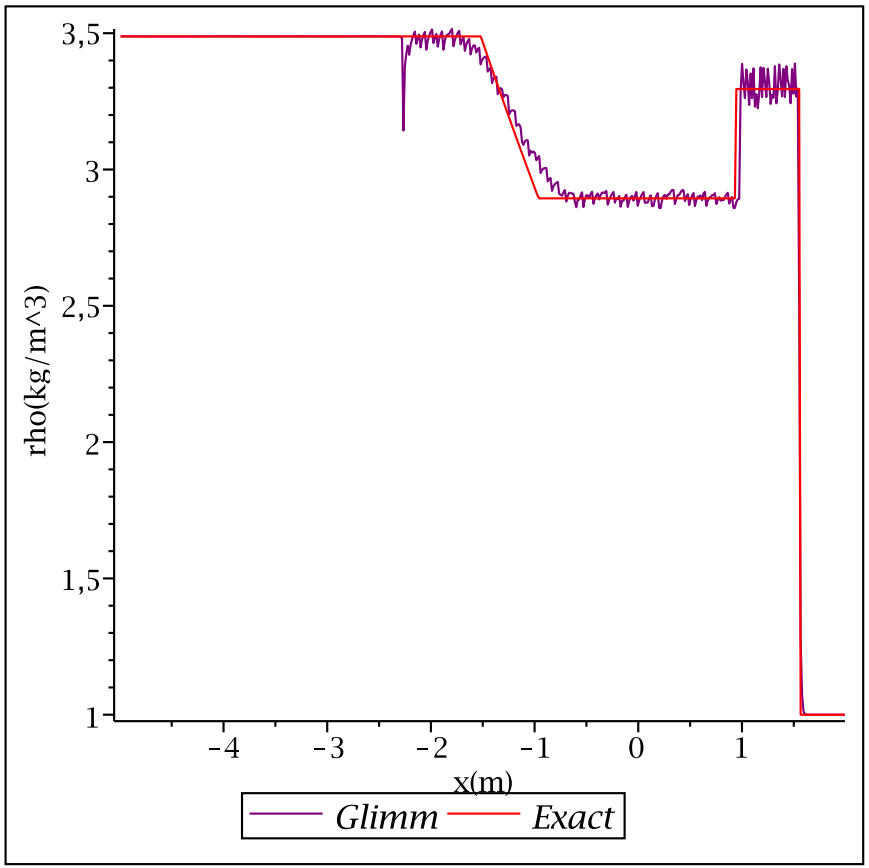
BUT...

Lagrange + Glimm remap: BV instabilities

We consider the following shock-interface interaction

Quantities	$x < -4$	$-4 < x < 1$	$x > 1$
$\rho(kg.m^{-3})$	3.488	2	1
$u(m.s^{-1})$	1.13	-1	-1
$p(Pa)$	23.33	2	2
φ	1	1	0
γ	1.4	1.4	2
$p_\infty(Pa)$	0	0	7

We observe BV instabilities due to the strong shock:



ALE+Glimm projection

- We choose to move the boundary $x_{i+\frac{1}{2}}$ at the speed of the fluid only at the two-fluid interface

$$\xi^n_{i+\frac{1}{2}} = \begin{cases} u^*_{i+\frac{1}{2}} & \text{if} \quad \left(\varphi^n_i - \frac{1}{2}\right)\left(\varphi^n_{i+1} - \frac{1}{2}\right) < 0, \\ 0 & \text{otherwise.} \end{cases}$$

- We perform a random sampling

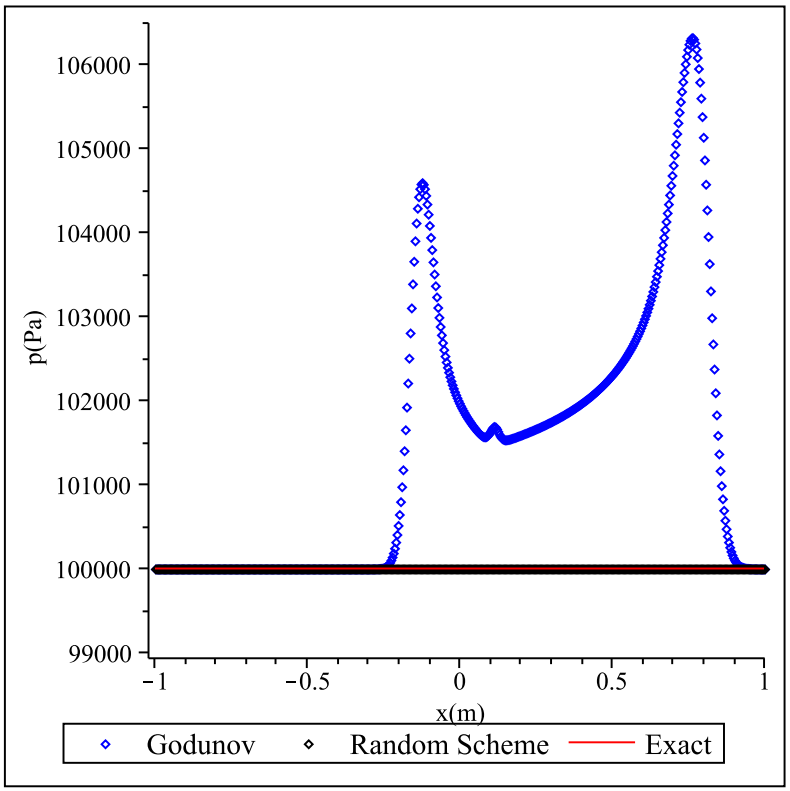
$$W^{n+1}_i = W^{n+1,-}(x_{i-\frac{1}{2}} + \omega_n \Delta x, \Delta t),$$

where $\omega_n \in [0; 1[$ is a pseudo-random number.

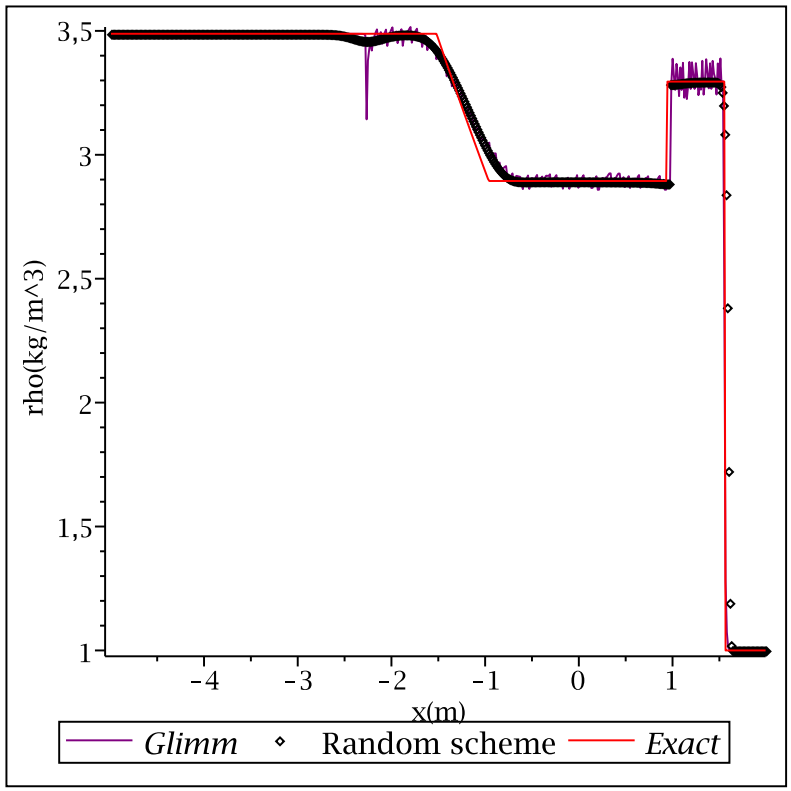
$\rightarrow$ We denote this scheme the "Random Scheme" (RS).

Results with the Random Scheme (RS)

No spurious pressure oscillations at the two-fluid interface:



No BV instabilities:



Properties of the Random Scheme

The Random Scheme has the following properties:

- it preserves hyperbolic set without diffusion $\Omega_{\varphi=0} \cup \Omega_{\varphi=1}$,
- it does not diffuse the gas mass fraction φ,
- it is statistically conservative,
- it satisfies statistically a discrete entropy inequality,
- it does not introduce spurious oscillations at the two-fluid interface,
- it handles vacuum.

A 1D test: comparison with other schemes

We compare the Random Scheme with other schemes:

- the "Ghost Fluid for the poor" (GF) algorithm of Abgrall-Karni [AK01],
- the "SA" algorithm of Saurel-Abgrall [SA99],
- the "NonConsPhi" scheme that includes two steps:
  - a Lagrangian step: as the Lagrange + Glimm scheme,
  - a projection step: we use a conservative projection for ρ, ρu, ρv and ρE and for φ, we take:

$$\varphi_i^{n+1} = \varphi_i^{n+1,-} - \frac{\Delta t}{\Delta x} \Big(\max(u_{i-\frac{1}{2}}, 0)(\varphi_i^{n+1,-} - \varphi_{i-1}^{n+1,-}) + \min(u_{i+\frac{1}{2}}, 0)(\varphi_{i+1}^{n+1,-} - \varphi_i^{n+1,-}) \Big).$$

- the "CC" algorithm of Chalons-Coquel [CC10]. Lagrange-projection scheme with a projection of the pressure and a random sampling on φ at the two-fluid interface.

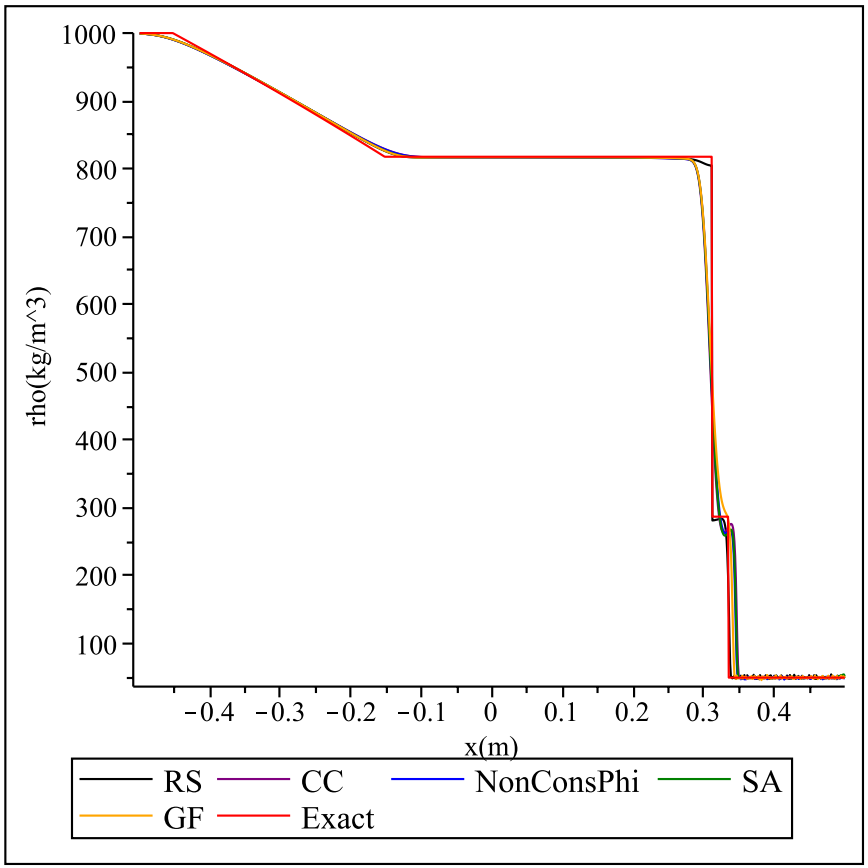
A 1D test

We test the Random Scheme on a water-air shock tube with discontinuous normal velocity.

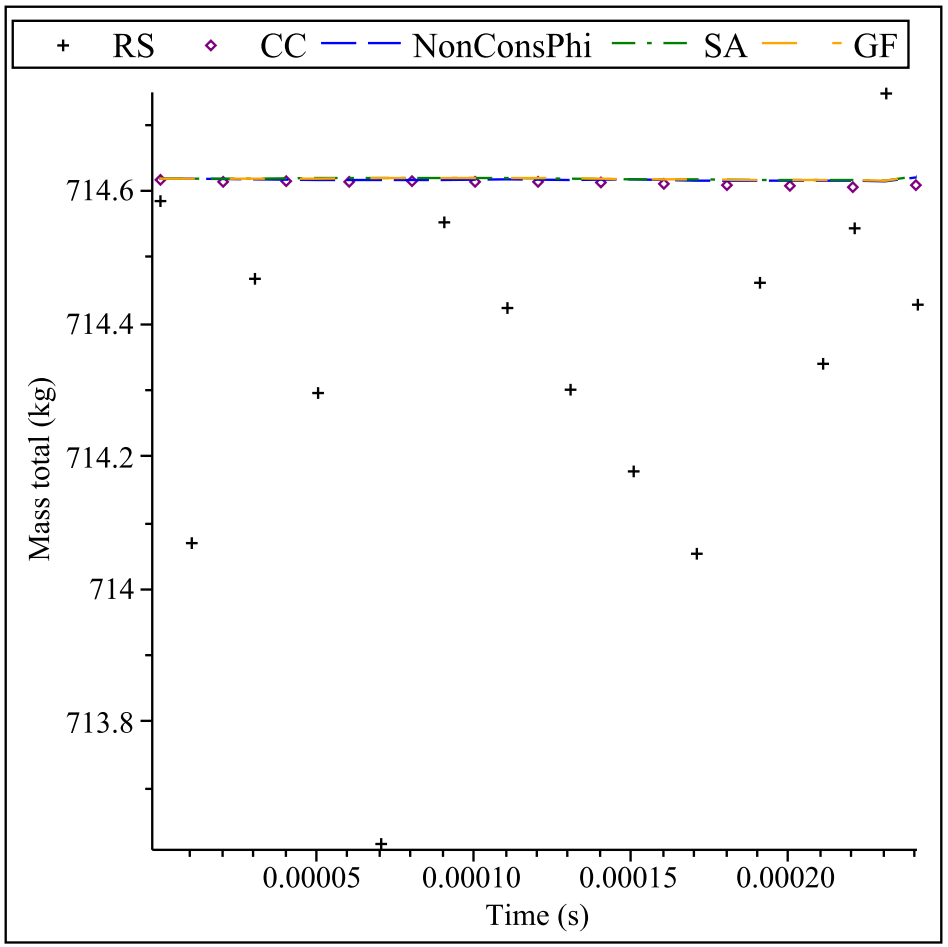
Initial condition:

Quantities	x<0.2	x>0.2
$\rho(kg.m^{-3})$	1000	50
$u(m.s^{-1})$	0	0
$v(m.s^{-1})$	1000	-5000
$p(Pa)$	$1e9$	$1e5$
φ	0	1
γ	4.4	1.4
$p_\infty(Pa)$	$6.8e8$	0

We plot the numerical and the exact solution at time $t = 240\mu m$ with 1000 cells:

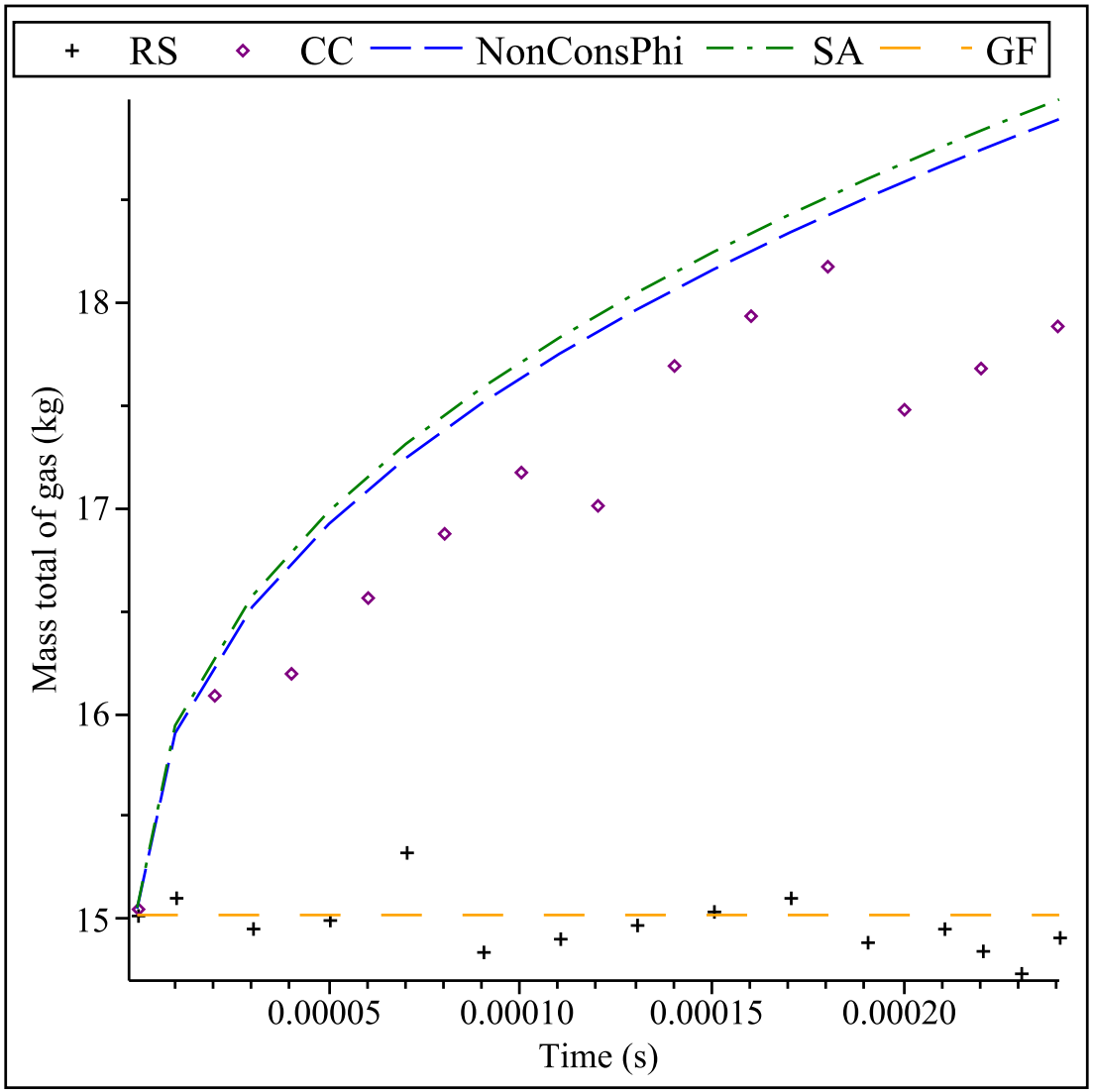


Conservation of the total mass (gas+liquid) until 240μs

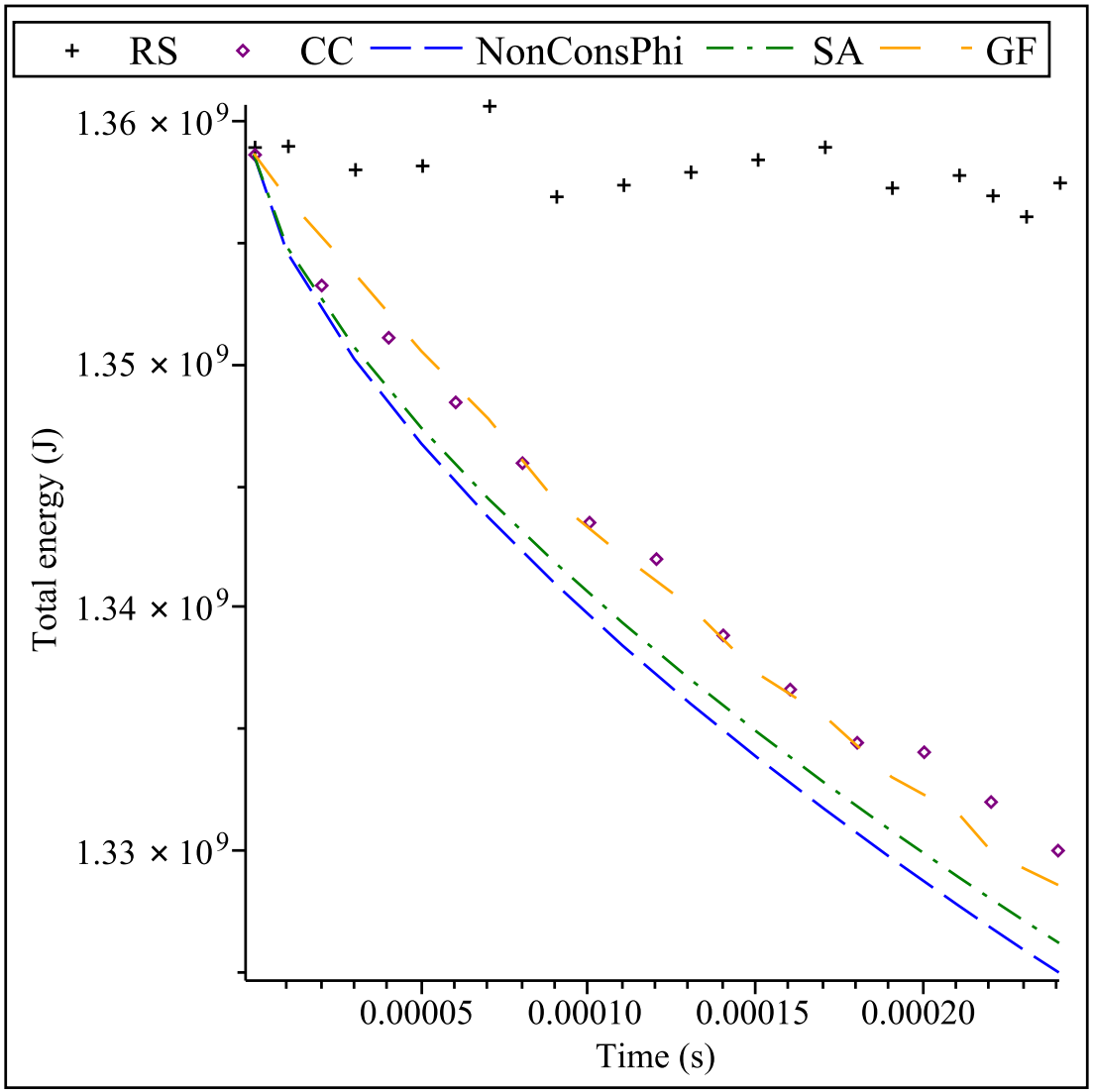


$\rightarrow$ Error less than 0.1%.

Conservation of the mass of gas until $240\mu s$



Conservation of the total energy until $240\mu s$



Extension to 2D with directional splitting

- For solving

$$\begin{cases} \partial_t W + \partial_x F(W) + \partial_y G(W) = 0, \\ W(x, y, t = 0) = W_0(x, y), \end{cases}$$

between time $t = 0$ and $t = \Delta t$, we use directional splitting (Godunov [G59] or Strang [S68]).

- Colella [C78] showed that the directional splitting coupled with the Glimm scheme does not converge (because of the nonlinear waves).
  → Does the directional splitting work with the Random Scheme ?

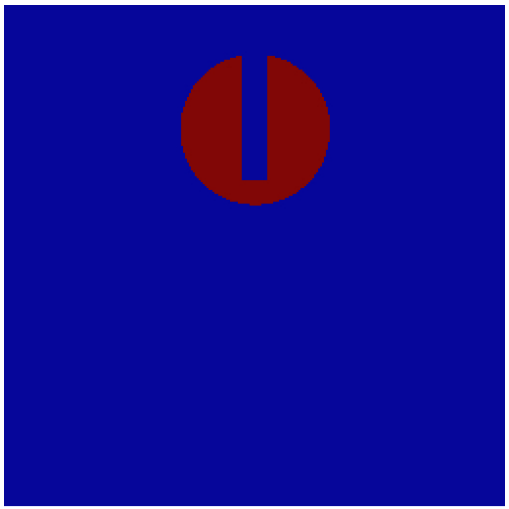
Test of Zalesak [Z79]

We consider a solid body rotation. We choose

$$u = -\Omega(y - y_0), \qquad v = \Omega(x - x_0),$$

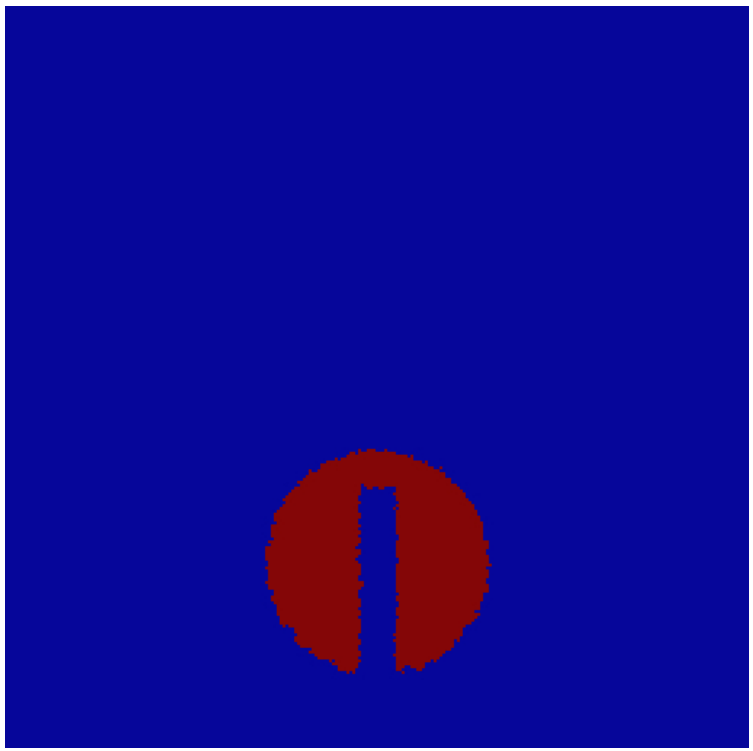
where $\Omega = \frac{2\pi}{628}$ is the angular velocity and (x_0, y_0) is the axis of rotation.

- Initial time:

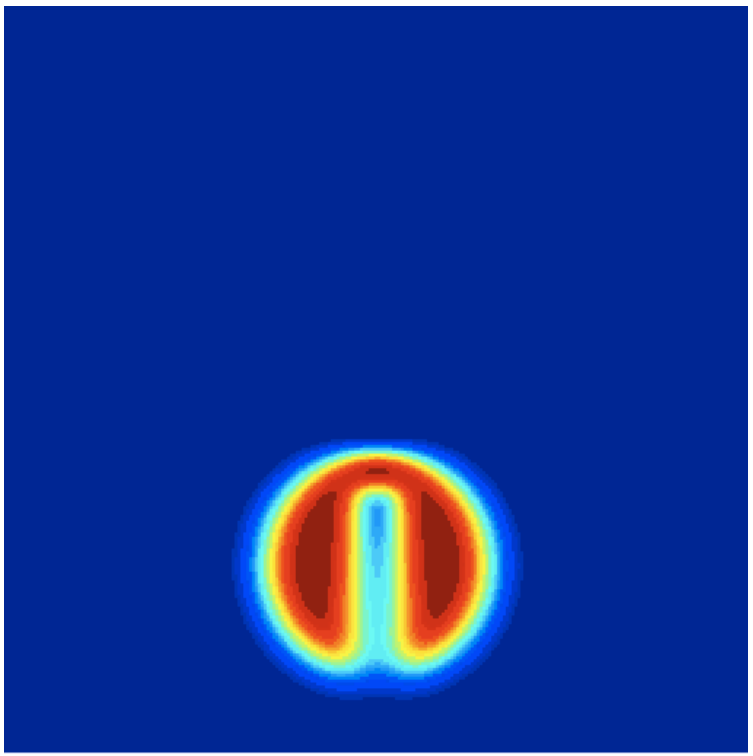


Test of Zalesak [Z79]: $\frac{1}{2}$ revolution

Random Scheme:

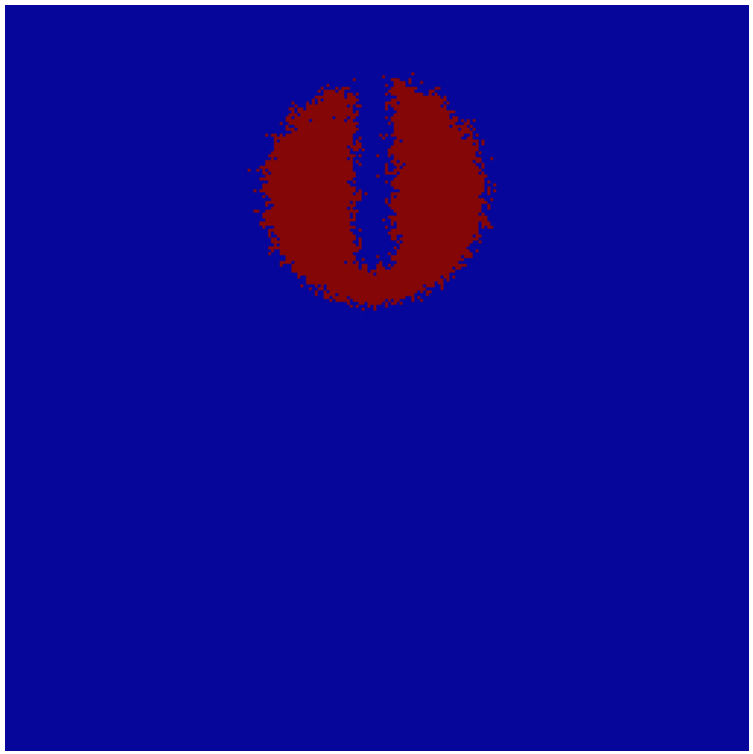


Conservative scheme:

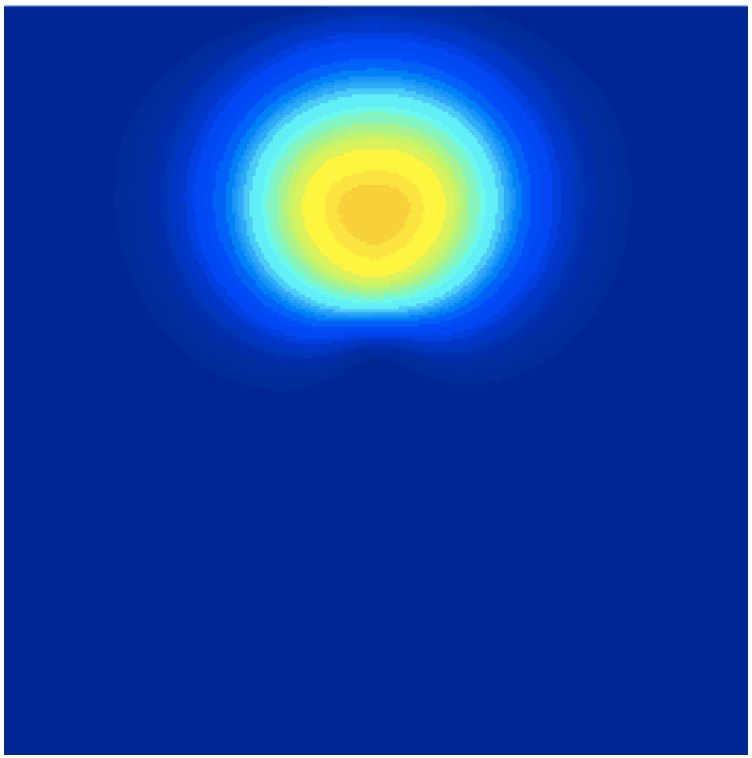


Test of Zalesak [Z79]: 5 revolutions

Random Scheme:

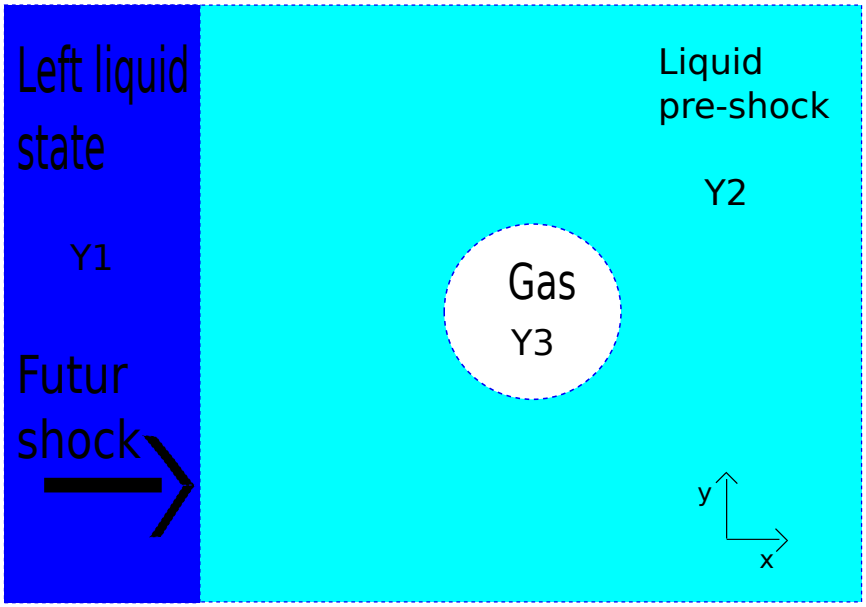


Conservative scheme:



Shock bubble interaction (water/gas)

We consider the shock-bubble interaction test of [KL2010]

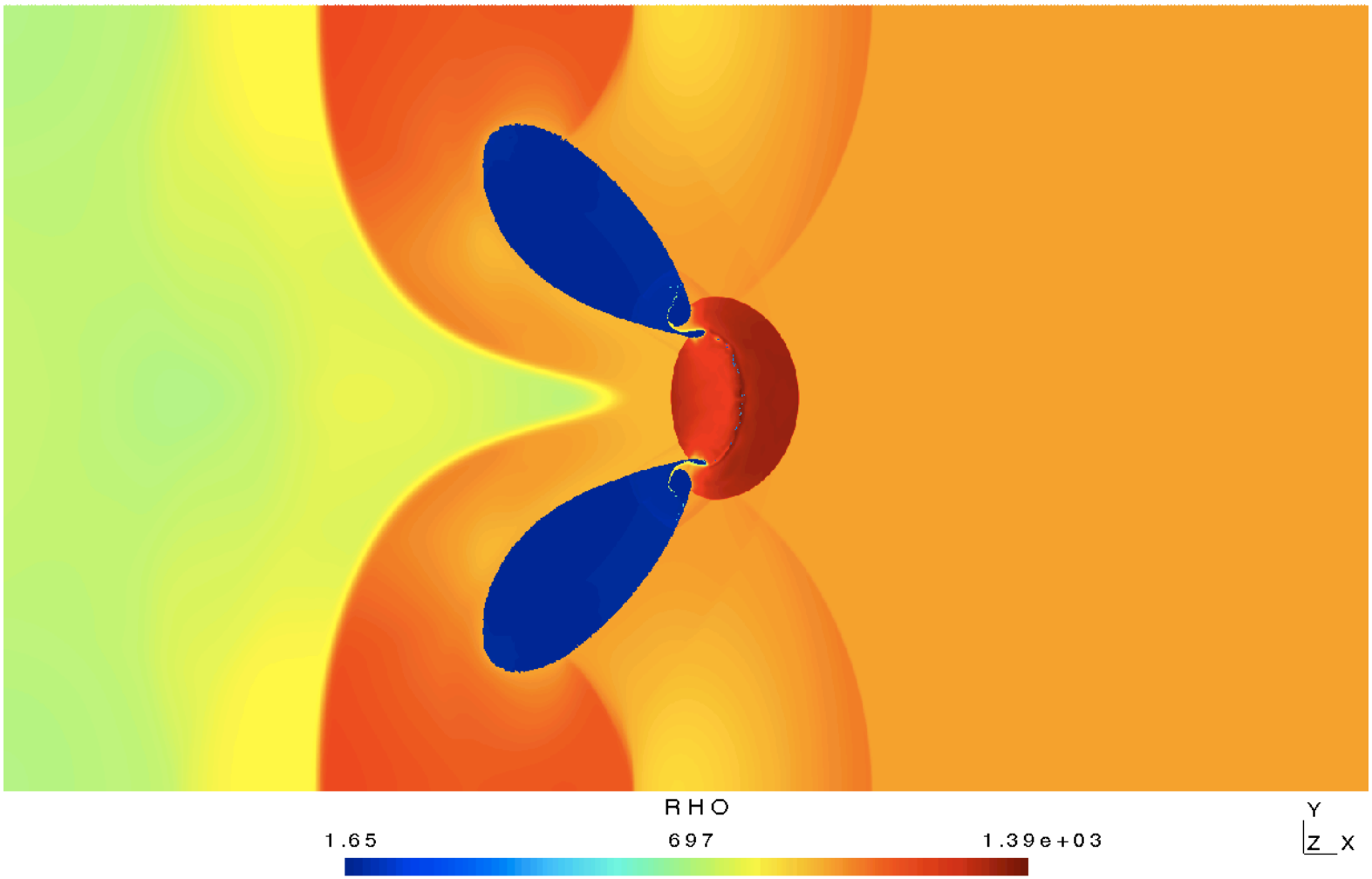


The initial data are:

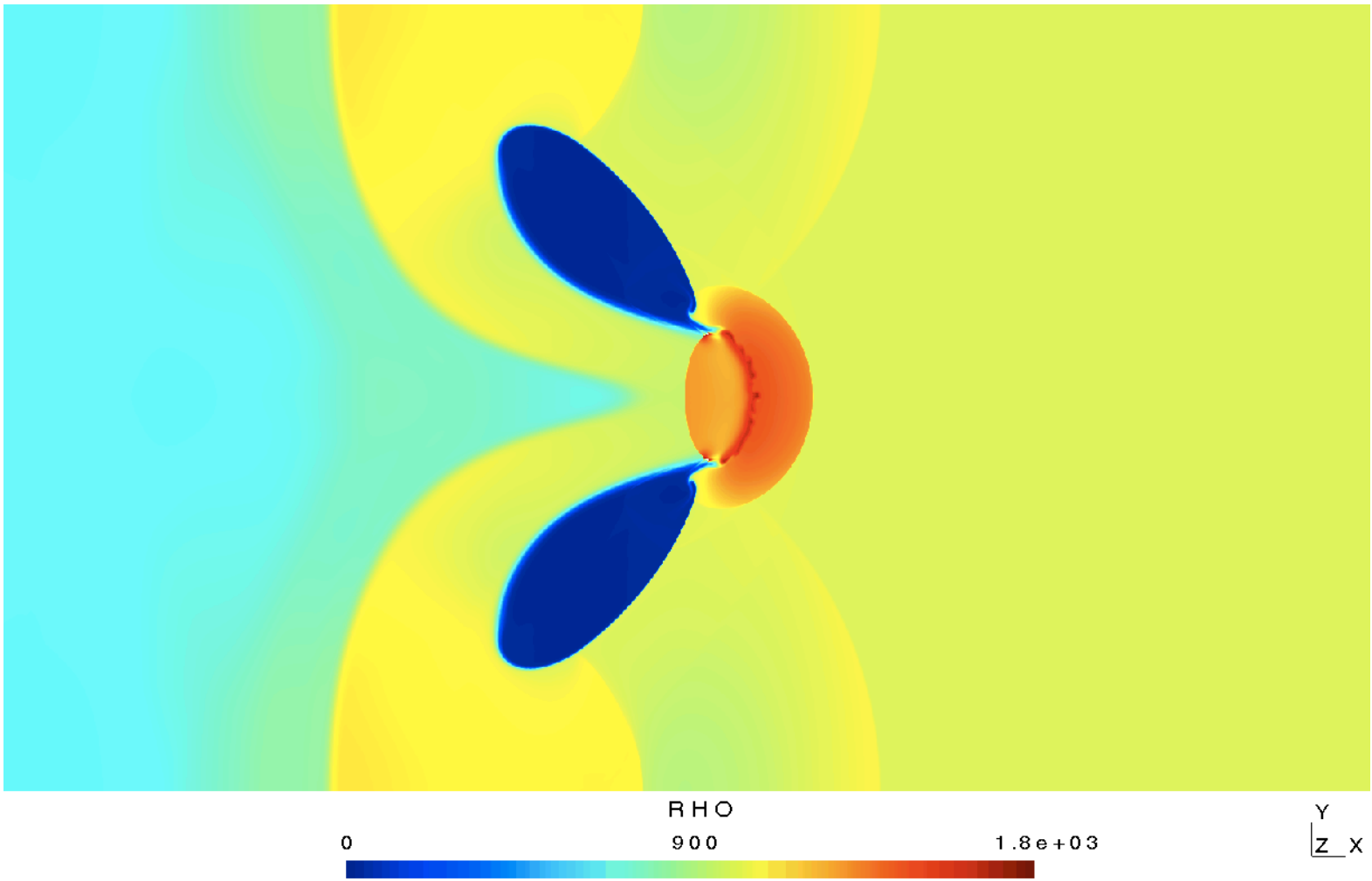
Quantities	Y1	Y2	Y3
$\rho(kg.m^{-3})$	1030.9	1000	1
$u(m.s^{-1})$	300	0	0
$v(m.s^{-1})$	0	0	0
$p(Pa)$	3.0e9	1.0e5	1.0e5
φ	0	0	1
γ	4.4	4.4	1.4
$p_\infty(Pa)$	6.8e8	6.8e8	0

Final time=$450\mu s$.
Grid: $14\,000 \times 7\,000$.
Time of computation on GPU (AMD Radeon HD7970): $\approx 1h$.

Density ρ at $450\mu s$: Random Scheme

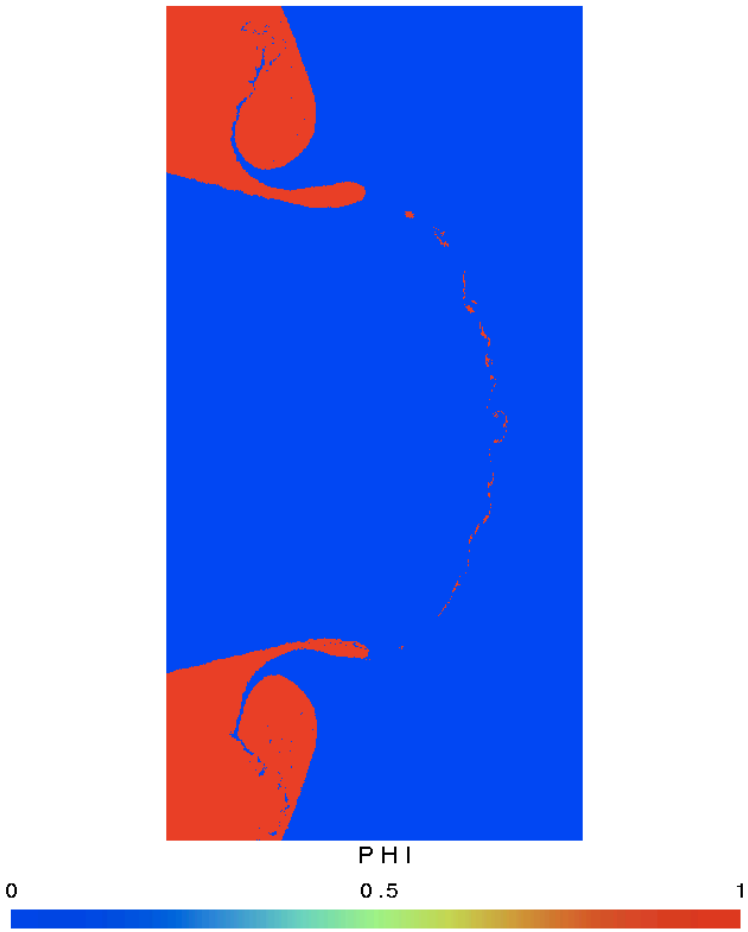


Density ρ at $450\mu s$: NonConsPhi scheme

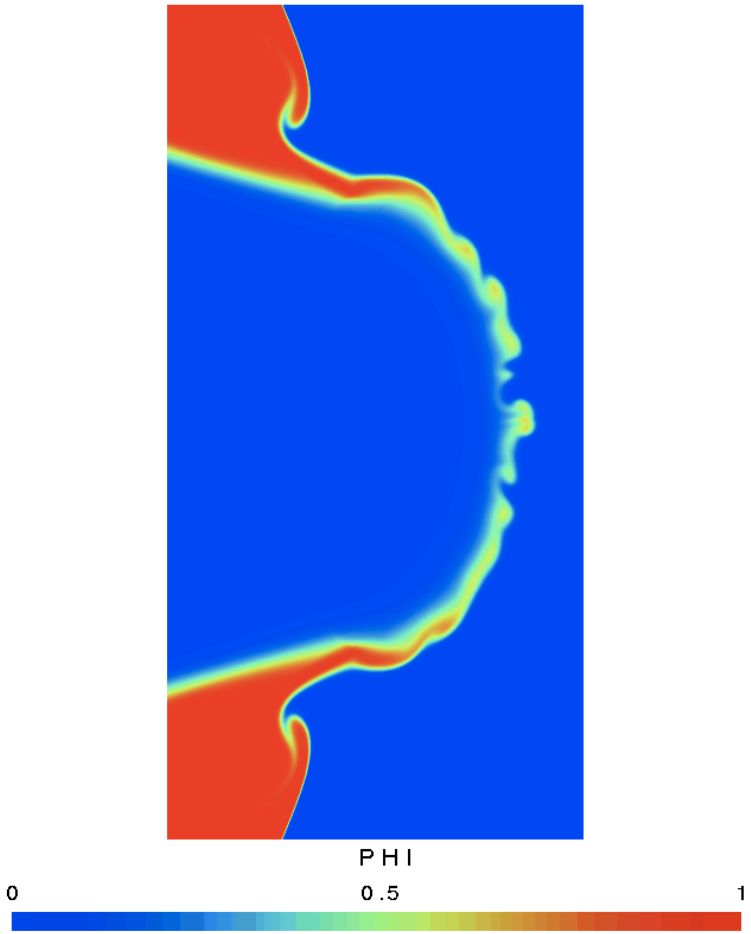


Zoom at $450\mu s$

Random Scheme:



NonConsPhi scheme:



$\rightarrow$ Different shapes with the two algorithms.

GPU and MPI implementation

What is a GPU?

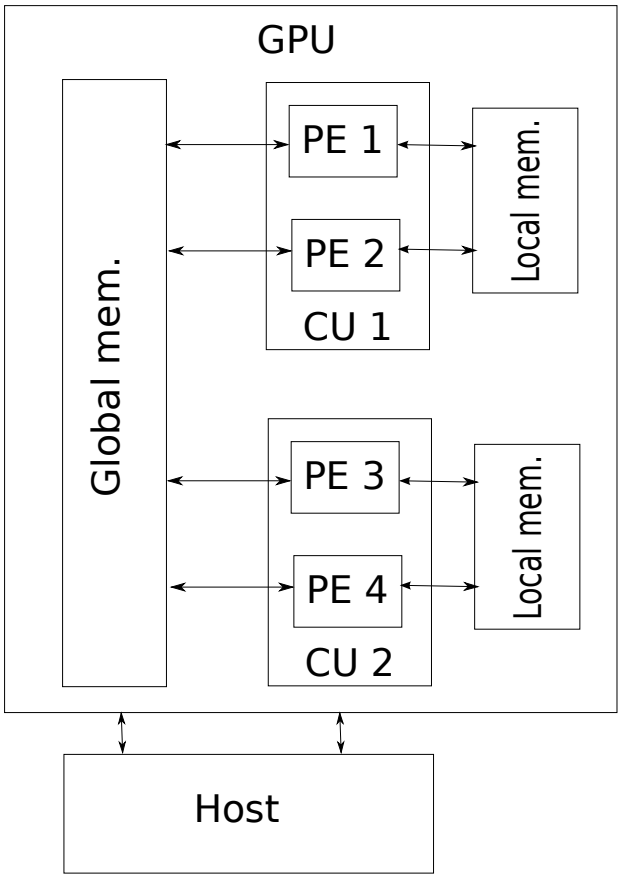
A modern Graphics Processing Unit (GPU) is made of:

- Global memory ($\approx$ 1 GB).
- Compute units ($\approx$ 27).

Each compute unit is made of:

- Processing elements ($\approx$ 8).
- Local memory ($\approx$16 kB).

A GPU with 2 Compute Units and 4 Processing Elements.



Programming rules on GPU

The same program can be executed on all the processing elements at the same time.

- All the processing elements have access to the global memory.
- The processing elements have only access to the local cache memory of their compute unit.
- The access to the global memory is slow while the access to the local memory is fast.
- The access to global memory is much faster if two neighboring processing elements read (or write) into two neighboring memory locations.
  $\rightarrow$ this is called "coalescent memory access".

OpenCL

- OpenCL means "Open Computing Language". It includes:
  - A library of C functions, called from the host, in order to drive the GPU.
  - A C-like language for writing the kernels that will be executed on the processing elements.
- Virtually, 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.
- Portable: the same program can run on a multicore CPU and a GPU. It is also possible to manage several devices in the same program.

Implementation in 2D

- Initialization: we initialize the data on the CPU and we send all the data to the global memory of the GPU.
- For each time step:
  - We associate to each row of the grid a work-group and to each cell of the row a virtual processor (work-item). We perform the flux computations and projections in the x direction for each work-group.
  - We "transpose": we exchange the ρu and ρv components and we reorganize the data such that the x and y coordinates are exchanged.
  - We perform the flux computations and projections in the y direction for each work group. Thanks to the transposition, the memory access is coalescent.
  - We transpose to have the correct value in the correct place for the next time step.
- We send all the data to the CPU for post-processing.

Relaxation solver

- $F(W_L, W_R, \xi) = F(R(W_L, W_R, \xi)) - \xi R(W_L, W_R, \xi)$
  $\rightarrow$ to compute the numerical flux, we need to solve the exact Riemann solver.
  $\rightarrow$ It is not efficient on GPU.
- We construct a relaxation solveur $\tilde{R}(W_L, W_R, \xi)$. We extend the Bouchut relaxation solver [B04] to
  - the two-fluid flow,
  - the ALE approach,
  - the projection.
- With the relaxation solver $\tilde{R}$, the Random Scheme keeps the same properties:
  - it preserve the hyperbolic set $\Omega_{\varphi=0} \cup \Omega_{\varphi=1}$,
  - it is statistically conservative and satisfies a discrete entropy inequality,
  - no spurious oscillations,
  - it handles vacuum.

Speedup

Implementation on GPU : The computation corresponds to 300 iterations on a grid 1024×512.

		Computation time (s)	Speedup
AMD A8 3850 (1 coeur)	CPU	527	1
AMD A8 3850 (4 coeurs)	CPU	205	2.6
NVIDIA GeForce 320M	GPU	56	9.4
AMD Radeon HD5850	GPU	3	175
AMD Radeon HD7970	GPU	2	260

There are two points to obtain this speedup:

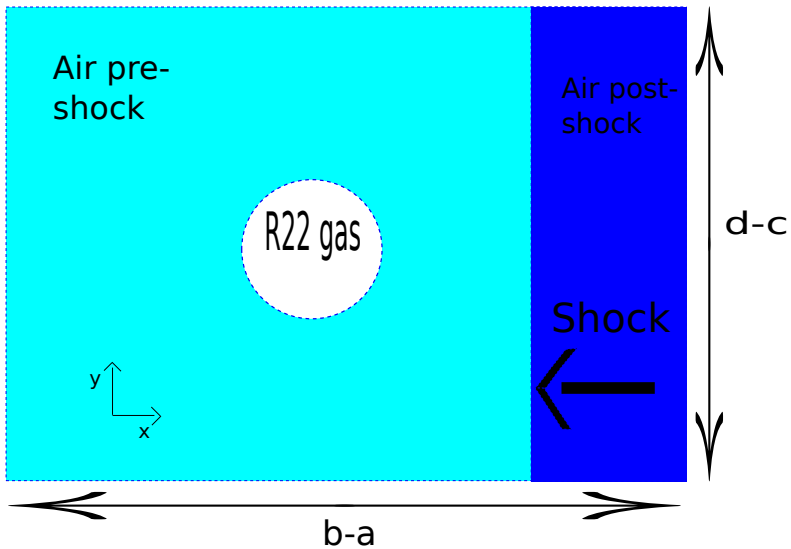
- we used an **optimized transposition** to have coalescent access in x and y directions.
  $\rightarrow$ 10 times faster with the transposition.
- the **relaxation solver**.
  $\rightarrow$ 50 times faster on GPU than exact solver.

OpenGL: video

- The code is very efficient on GPU.
  → The limiting factor is the memory of the GPU (1GB).

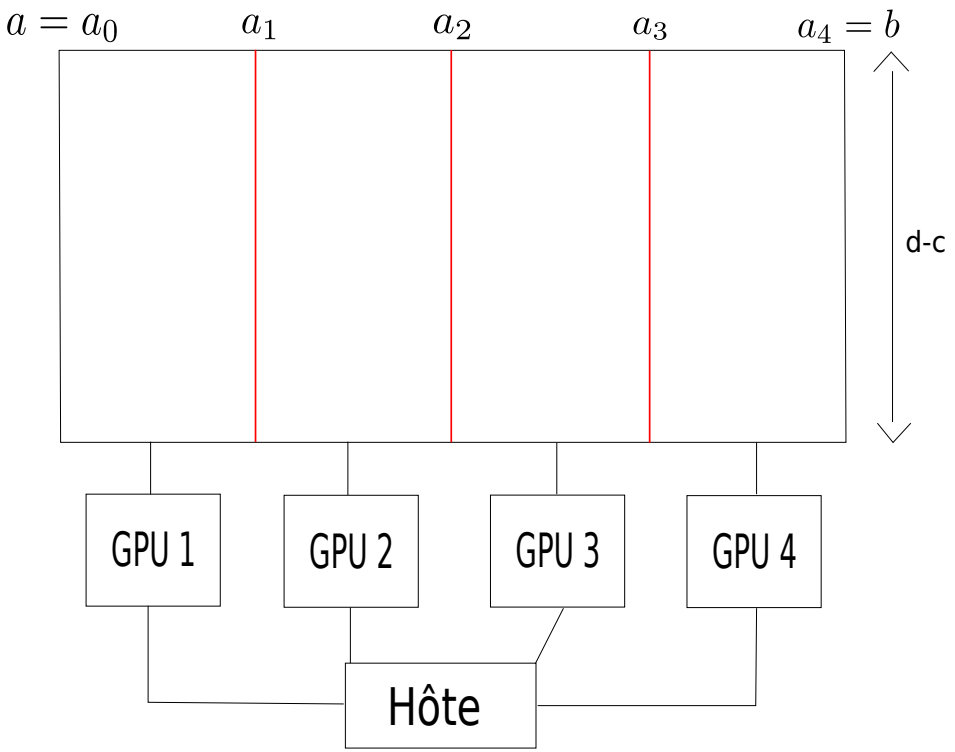
- We introduce Message Passing Interface (MPI) to consider finer mesh.

We present the 4 GPUs MPI implementation for the following test case



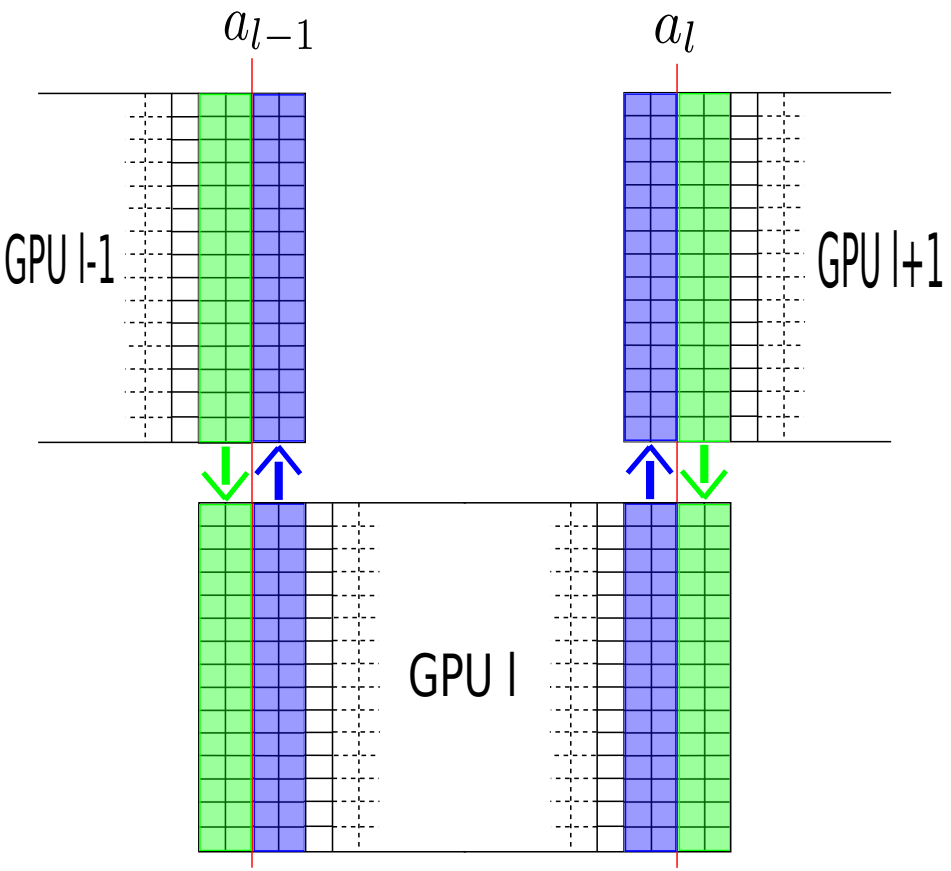
Splitting of the domain

- We split the domain in the x direction into 4 subdomains with a small overlap.
- Each GPU is associated to a subdomain.



MPI communications

Before each iteration, we have to exchange several layers of cells between neighboring GPUs.



Overlap =

- 1 for classical finite volume scheme,
- 2 for ALE-projection scheme,
- 5 for ALE-projection scheme with second order MUSCL reconstruction.

Speedup

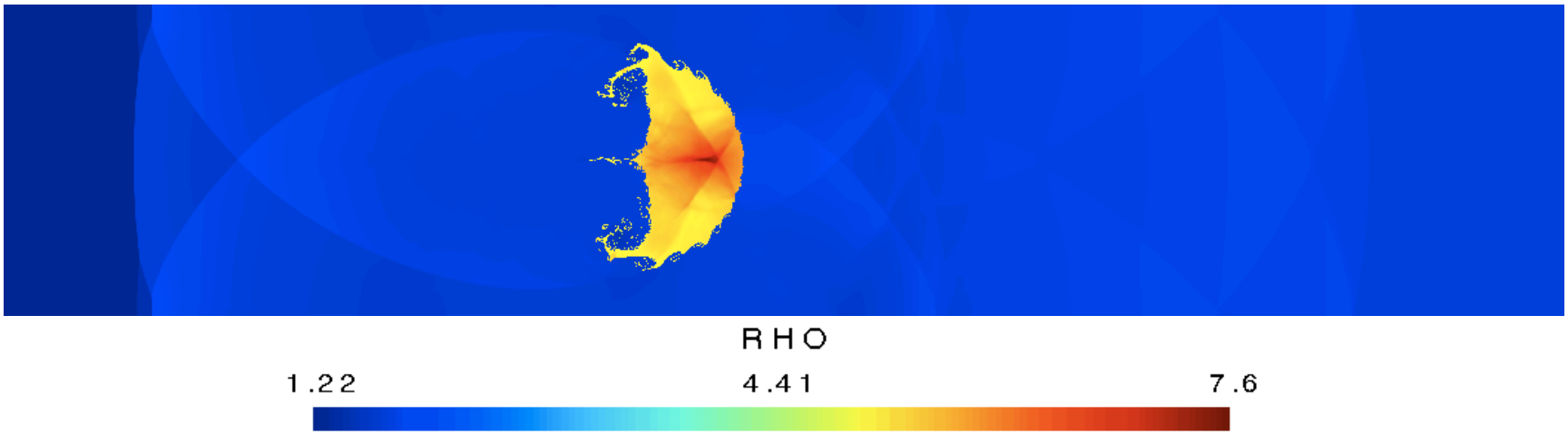
Implementation on MPI : Computation on a cluster of 4 cards AMD Radeon HD7970

Grille	1 GPU	4 GPUs	Speedup
2048×2048	14 s	14 s	1
4096×2048	22 s	16 s	1.4
4096×4096	77 s	60 s	1.3
8192×4096	150 s ?	61 s	2.5
16384×4096	600 s ?	230 s	2.6

$\rightarrow$ We can consider finer meshes. The MPI speedup is > 1 but not optimal. It could be improved by computations and communications overlap.

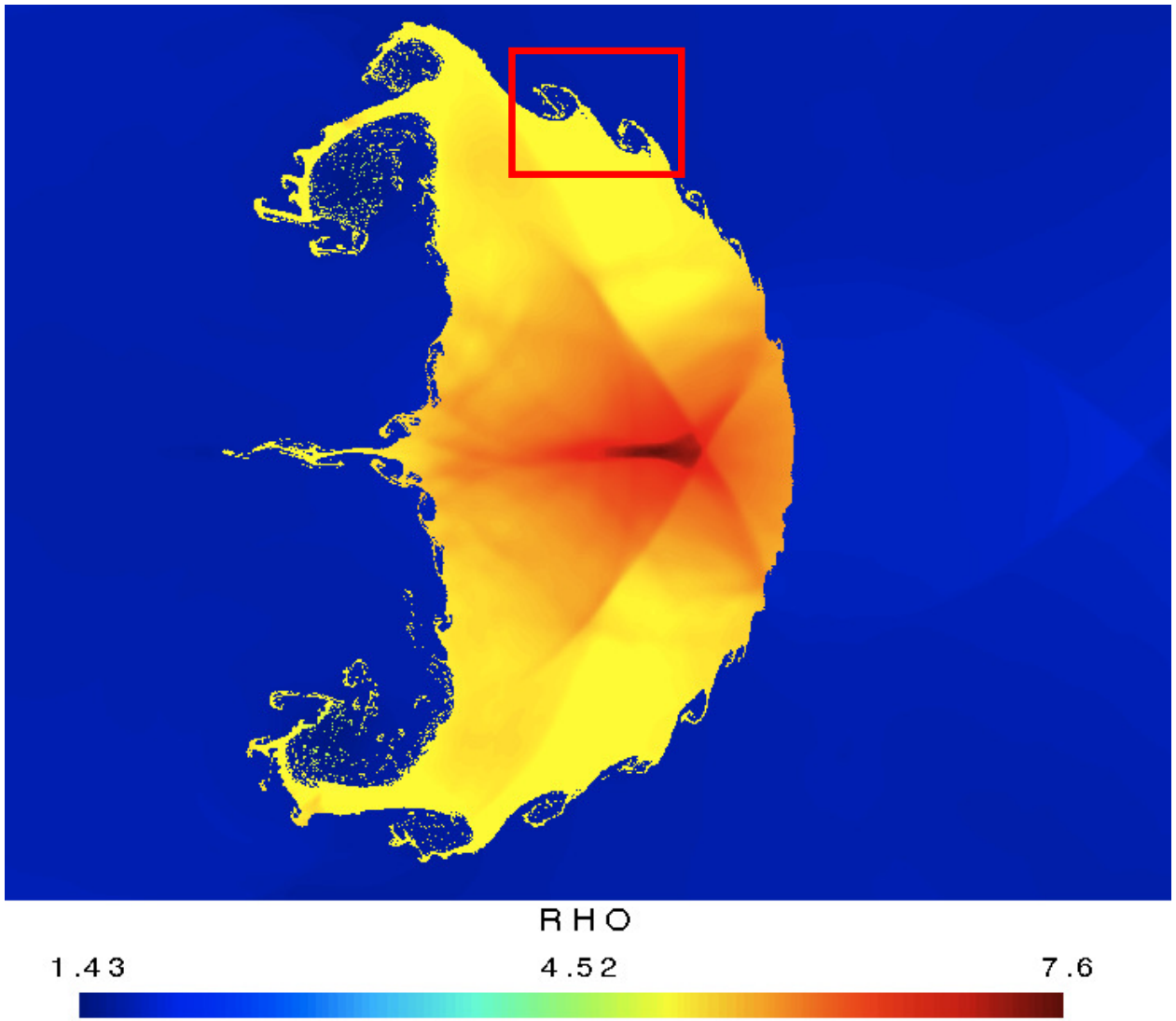
Schock R22/Air interaction: ρ at $600\mu s$

We consider a mesh of $20\,000 \times 5\,000$ cells.

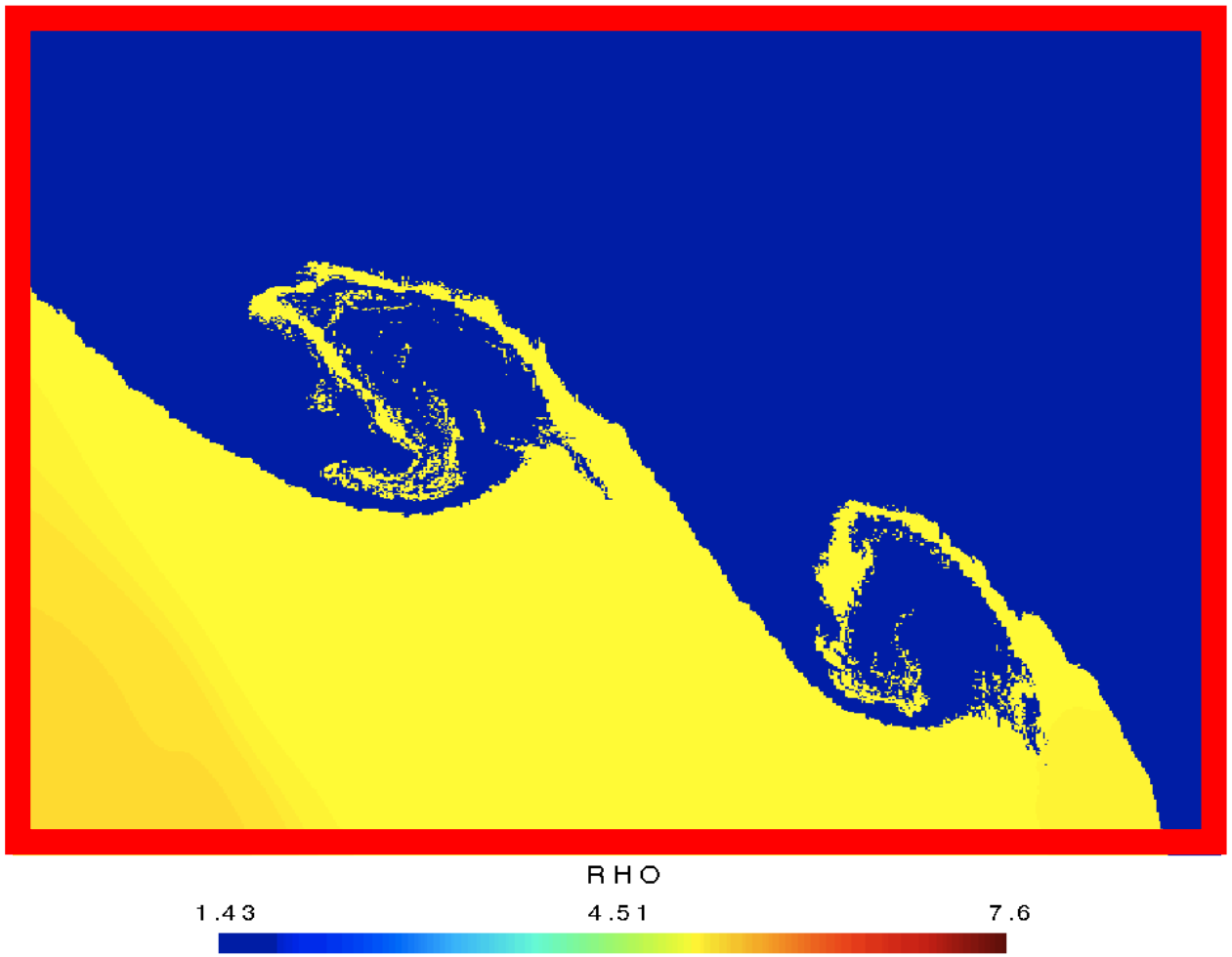


$\rightarrow$ We zoom on the bubble.

Schock R22/Air interaction: zoom on the bubble



Schock R22/Air interaction: zoom on the Rayleigh-Taylor instabilities



Conclusion and perspective

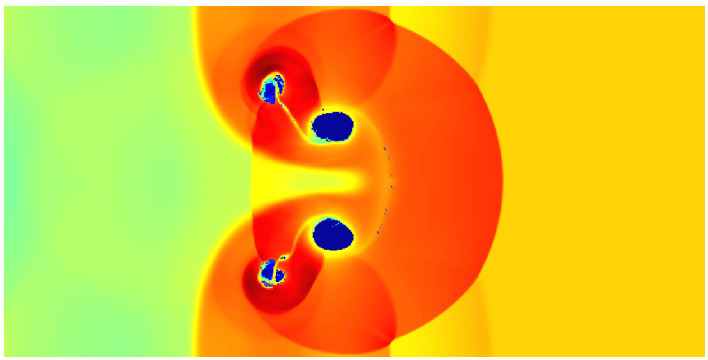
Conclusion:

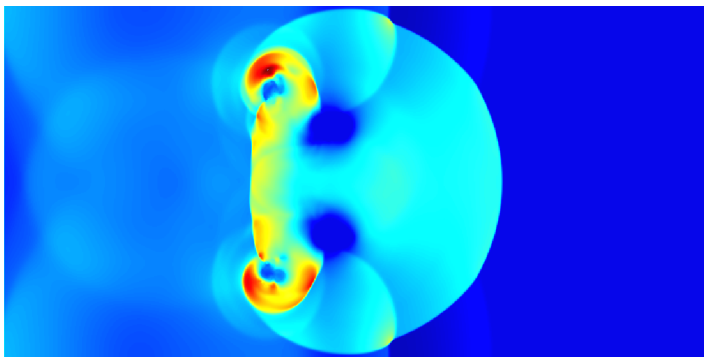
- We developed a robust algorithm to treat complex two-fluid compressible flows. The RS has the following properties:
  - it preserves the hyperbolic set without diffusion $\Omega_0 \cup \Omega_1$,
  - it is statistically conservative and satisfies a discrete entropy inequality.
- The code is very efficient on GPU+MPI, we need only few minutes to compute a complex flow on a mesh with millions of cells.

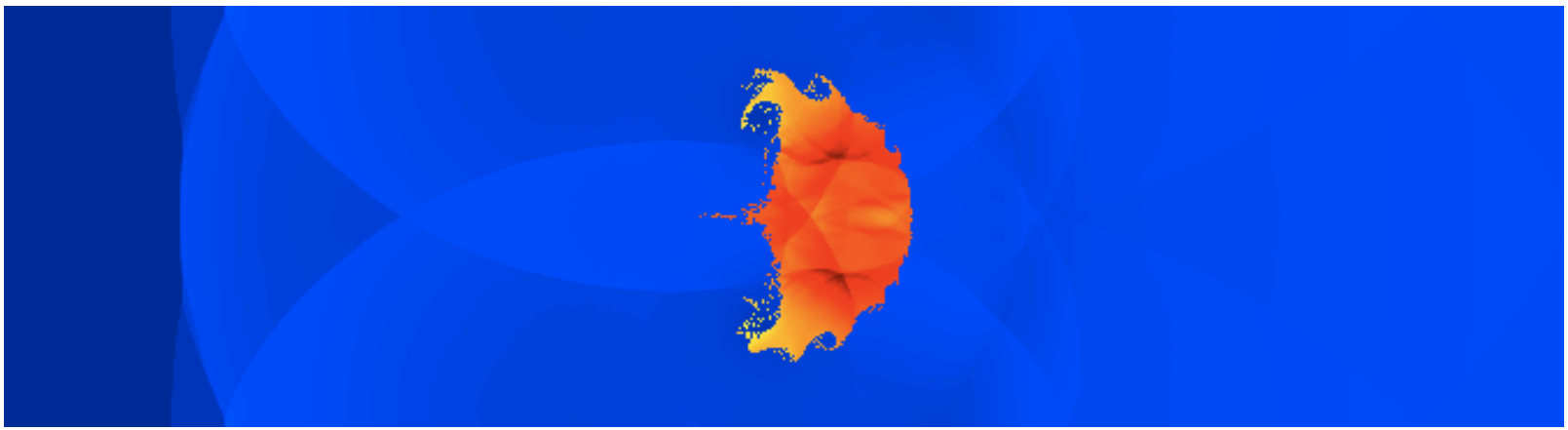
Perspectives:

- Extend the code in 3D.
- Test the Random Scheme with the "hyperbolic convex" pressure law $\tilde{p}$.
- Extend the approach to low Mach number flows.

Thank you for your attention!








R. Abgrall. Generalisation of the Roe scheme for the computation of mixture of perfect gases. Recherche aérospatiale, 6, 31-43, 1988.


R. Abgrall, S. Karni. Computations of compressible multifluids. Journal of Computational Physics, 169(2): 594-623, 2001.


T. Barberon, P. Helluy, S. Rouy. Practical computation of axisymmetrical multifluid flows, http://hal.archives-ouvertes.fr/hal-00139598/fr/.


F. Bouchut. Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-Balanced Schemes for Sources. Frontiers in Mathematics series, Birkhäuser, 2004.

 H.B. Callen. Thermodynamics and an introduction to thermostatistics, second edition. Wiley and Sons, 1985.

 C. Chalons, F. Coquel. Computing material fronts with Lagrange-Projection approach. HYP2010 Proc. http://hal.archives-ouvertes.fr/hal-00548938/fr/.

 C. Chalons, P. Goatin. Godunov scheme and sampling technique for computing phase transitions in traffic flow modeling. Interfaces and Free Boundaries, 10(2), 197-221, 2008.

 G. Chanteperdrix, P. Villedieu and J.-P. Vila. A compressible model for separated two-phase flows computations. In ASME Fluids Engineering Division Summer.


P. Colella. An Analysis of the Effect of Operator Splitting and of the Sampling Procedure on the Accuracy of Glimm's Method. PhD thesis, University of California, USA, 1978.


J.-P. Croisille. Contribution à l'étude théorique et à l'approximation par éléments finis du système hyperbolique de la dynamique des gaz multidimensionnelle et multiespèces. PhD thesis, University of Paris VI, 1991.


E. Godlewski, P.-A. Raviart. Hyperbolic systems of conservation laws. Société de mathematiques appliquées et industrielles Marketing, 1991.


B. Després, F. Lagoutière. Numerical resolution of a two-component compressible fluid model with interfaces. Prog. Comp. Fluid Dyn. 7 (6) (2007) 295-310.


R. Fedkiw, T. Aslam, B. Merriman, S. Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). Journal of Computational Physics, 152, 457-492, 1999.


J. Glimm. Solutions in the large for nonlinear hyperbolic systems of equations. Comm. Pure Appl. Math., 18, 697-715, 1965.


S.K. Godunov. Finite difference methods for numerical computation of discontinuous solutions of the equations of fluid dynamics. Mat. Sbornik 47, pp. 271-295, 1959.


J.F. Hass, B. Sturtevant. Interaction of weak shock waves with cylindrical ansperical gas inhomogeneities. J. Fluid Mechanics, 181, 41-71,1997.

S. Karni. Multicomponent flow calculations by a consistent primitive algorithm. J. Comput. Phys, 112, 31-43,1994.

S. Kokh, F. Lagoutière. An anti-diffuse numerical scheme for the simulation of interfaces between compressible fluids by means of a five-equation model. Journal of Comp. Phy. 229 (2010) 2773-2809.

S. Müller, P. Helluy, J. Ballmann. Numerical simulation of a single bubble by compressible two-phase fluids. J. Numer. Methods Fluids 62, no. 6, 591–631, 2010.

R. Saurel, R. Abgrall, A simple method for compressible multifluid flows. SIAM J. Sci. Comput. 21 (1999), no. 3, 1115–1145.

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


T. Steven, Zalesak. Fully Multidimensional Flux-Corrected Transports Algorithms for Fluids. Journal of Comp. Phy. 31 (1979) 335-362.