UNIVERSITÉ DE TOULON

SeaTech ÉCOLE D'INGÉNIEURS

On the numerical entropy production as a useful mesh refinement parameter: application to wave-breaking.

Mehmet Ersoy[1], Frédéric Golay[2] and Lyudmyla Yushchenko[3]

The Third BCAM Workshop on Computational Mathematics

Bilbao,
July 17-18, 2014

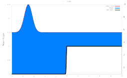
1. *Mehmet.Ersoy@univ-tln.fr*
2. *Frederic.Golay@univ-tln.fr*
3. *Lyudmyla.Yushchenko@univ-tln.fr*

Outline of the talk

OUTLINE

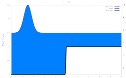
1. Physical modeling and numerical motivation
2. 2D and 3D applications
3. Concluding remarks& perspectives

- Shallow water equations : fast but unable to simulate wave breaking
  - Zaleski, Popinet, Diaz, Dutykh, . . .



(a) SW

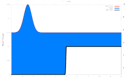
- Shallow water equations : fast but unable to simulate wave breaking
  - Zaleski, Popinet, Diaz, Dutykh, . . .
- Multi-phase Navier-Stokes equations :
  - FV, FE, VOF, level set, . . . → accurate but expensive
    - Nkonga, Lubin, Caltagirone . . .



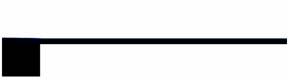
(d) SW

(e) Nkonga (FluidBox) (2009)

- Shallow water equations : fast but unable to simulate wave breaking
  - Zaleski, Popinet, Diaz, Dutykh, . . .
- Multi-phase Navier-Stokes equations :
  - FV, FE, VOF, level set, . . . → accurate but expensive
    - ⋆ Nkonga, Lubin, Caltagirone . . .
  - Lattice Boltzmann → accurate but expensive
    - ⋆ Janssen, Grilli, Krafczyk, . . .

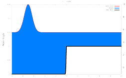


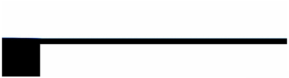
(g) SW



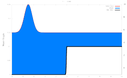
(h) Nkonga (FluidBox) (2009)

- Shallow water equations : fast but unable to simulate wave breaking
  - Zaleski, Popinet, Diaz, Dutykh, . . .
- Multi-phase Navier-Stokes equations :
  - FV, FE, VOF, level set, . . . → accurate but expensive
    - ⋆ Nkonga, Lubin, Caltagirone . . .
  - Lattice Boltzmann → accurate but expensive
    - ⋆ Janssen, Grilli, Krafczyk, . . .
  - SPH → difficult to implement and very expensive
    - ⋆ Monaghan, Lattanzio, De Padova, . . .


(j) SW


(k) Nkonga (FluidBox) (2009)

- Shallow water equations : fast but unable to simulate wave breaking
  - Zaleski, Popinet, Diaz, Dutykh, ...
- Multi-phase Navier-Stokes equations :
  - FV, FE, VOF, level set, ... → accurate but expensive
    - Nkonga, Lubin, Caltagirone ...
  - Lattice Boltzmann → accurate but expensive
    - Janssen, Grilli, Krafczyk, ...
  - SPH → difficult to implement and very expensive
    - Monaghan, Lattanzio, De Padova, ...
- Low-Mach models (Euler equations) : good compromise between physical modeling accuracy and cost



(m) SW

(n) Nkonga (FluidBox) (2009)

(o) Golay & Helluy (2005)

We focus on general non linear hyperbolic conservation laws

$$\left\{ \begin{array}{l} \dfrac{\partial \boldsymbol{w}}{\partial t} + \dfrac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial x} = 0,\ (t,x) \in \mathbb{R}^+ \times \mathbb{R} \\ \boldsymbol{w}(0,x) = \boldsymbol{w}_0(x),\ x \in \mathbb{R} \end{array} \right.$$

$\boldsymbol{w} \in \mathbb{R}^d$	:	vector state,
$\boldsymbol{f}$	:	flux governing the physical description of the flow.

We focus on general non linear hyperbolic conservation laws

$$\left\{ \begin{array}{l} \dfrac{\partial \boldsymbol{w}}{\partial t} + \dfrac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial x} = 0, \ (t,x) \in \mathbb{R}^+ \times \mathbb{R} \\ \boldsymbol{w}(0,x) = \boldsymbol{w}_0(x), \ x \in \mathbb{R} \end{array} \right.$$

Weak solutions satisfy

$$S = \frac{\partial s(\boldsymbol{w})}{\partial t} + \frac{\partial \psi(\boldsymbol{w})}{\partial x} \left\{ \begin{array}{lll} = & 0 & \text{for smooth solution} \\ = & 0 & \text{across rarefaction} \\ < & 0 & \text{across shock} \end{array} \right.$$

where (s, ψ) stands for a convex entropy-entropy flux pair :

$$(\nabla \psi(\boldsymbol{w}))^T = (\nabla s(\boldsymbol{w}))^T \ D_{\boldsymbol{w}} \boldsymbol{f}(\boldsymbol{w})$$

We focus on general non linear hyperbolic conservation laws

$$\left\{ \begin{array}{l} \dfrac{\partial \boldsymbol{w}}{\partial t} + \dfrac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial x} = 0,\ (t,x) \in \mathbb{R}^+ \times \mathbb{R} \\ \boldsymbol{w}(0,x) = \boldsymbol{w}_0(x),\ x \in \mathbb{R} \end{array} \right.$$

Weak solutions satisfy

$$S = \frac{\partial s(\boldsymbol{w})}{\partial t} + \frac{\partial \psi(\boldsymbol{w})}{\partial x} \left\{ \begin{array}{lll} = & 0 & \text{for smooth solution} \\ = & 0 & \text{across rarefaction} \\ < & 0 & \text{across shock} \end{array} \right.$$

where (s, ψ) stands for a convex entropy-entropy flux pair :

$$\left(\nabla \psi(\boldsymbol{w})\right)^T = \left(\nabla s(\boldsymbol{w})\right)^T \ D_{\boldsymbol{w}} \boldsymbol{f}(\boldsymbol{w})$$

Entropy inequality $\simeq$ "smoothness indicator"

We focus on general non linear hyperbolic conservation laws

$$\left\{\begin{array}{l}\dfrac{\partial \boldsymbol{w}}{\partial t} + \mathsf{div}(\boldsymbol{f}(\boldsymbol{w})) = 0,\ (t,x) \in \mathbb{R}^+ \times \mathbb{R}^d \\ \boldsymbol{w}(0,x) = \boldsymbol{w}_0(x),\ x \in \mathbb{R}^d\end{array}\right.$$

Weak solutions satisfy

$$S = \frac{\partial s(\boldsymbol{w})}{\partial t} + \mathsf{div}(\psi(\boldsymbol{w})) \left\{\begin{array}{lll} = & 0 & \text{for smooth solution} \\ = & 0 & \text{across rarefaction} \\ < & 0 & \text{across shock}\end{array}\right.$$

where (s,ψ) stands for a convex entropy-entropy flux pair :

$$(\nabla \psi_i(\boldsymbol{w}))^T = (\nabla s(\boldsymbol{w}))^T\ D_{\boldsymbol{w}} \boldsymbol{f}_i(\boldsymbol{w}), \quad i = 1, \ldots, d$$

Entropy inequality $\simeq$ "smoothness indicator"

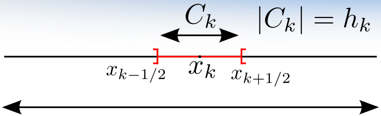


Figure : a cell C_k

Finite volume approximation :

$$\boldsymbol{w}_k^{n+1} = \boldsymbol{w}_k^n - \frac{\delta t_n}{h_k}\left(\boldsymbol{F}_{k+1/2}^n - \boldsymbol{F}_{k-1/2}^n\right)$$

with

$$\boldsymbol{w}_k^n \simeq \frac{1}{h_k}\int_{C_k} \boldsymbol{w}\left(t_n, x\right)\, dx$$

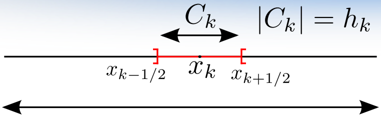


FIGURE : a cell C_k

Finite volume approximation :

$$\boldsymbol{w}_k^{n+1} = \boldsymbol{w}_k^n - \frac{\delta t_n}{h_k}\left(\boldsymbol{F}_{k+1/2}^n - \boldsymbol{F}_{k-1/2}^n\right)$$

with

$$\boldsymbol{w}_k^n \simeq \frac{1}{h_k}\int_{C_k} \boldsymbol{w}\left(t_n, x\right)\, dx$$

The numerical density of entropy production :

$$S_k^n = \frac{s_k^{n+1} - s_k^n}{\delta t_n} + \frac{\psi_{k+1/2}^n - \psi_{k-1/2}^n}{h_k} \lessapprox 0$$

- Compute w_k^n

Mesh refinement indicator : principle & illustration

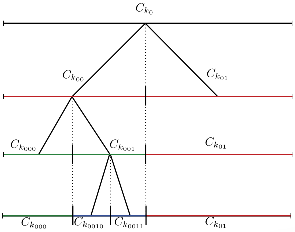
- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened

- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely :
  - $S_k^n \leqslant \alpha_{\mathsf{min}} \overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$

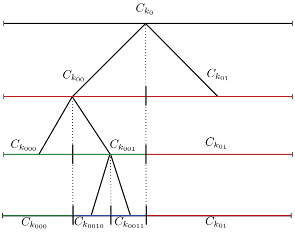
Mesh refinement indicator : principle & illustration

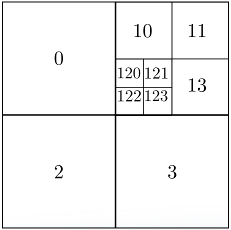
- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely :
  - $S_k^n \leqslant \alpha_{\text{min}}\overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \dfrac{1}{|\Omega|}\displaystyle\int_\Omega S_k^n$
  - $S_k^n \geqslant \alpha_{\text{max}}\overline{S} \Longrightarrow$ the cell is coarsened

- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely :
  - $S_k^n \leqslant \alpha_{\text{min}}\overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \dfrac{1}{|\Omega|}\displaystyle\int_\Omega S_k^n$
  - $S_k^n \geqslant \alpha_{\text{max}}\overline{S} \Longrightarrow$ the cell is coarsened
  - Dynamic mesh refinement :
    - ⋆ Dyadic tree (1D)
    - ⋆ hierarchical numbering : basis 2

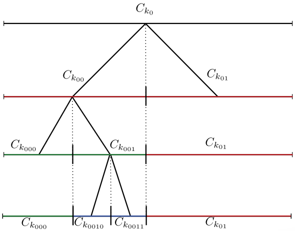


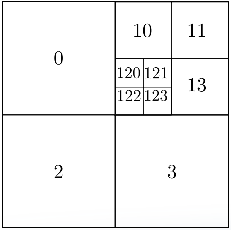
- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely :
  - $S_k^n \leqslant \alpha_{\text{min}}\overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \dfrac{1}{|\Omega|}\displaystyle\int_\Omega S_k^n$
  - $S_k^n \geqslant \alpha_{\text{max}}\overline{S} \Longrightarrow$ the cell is coarsened
  - Dynamic mesh refinement :
    - Non-structured grid : macro-cell
    - Dyadic tree (1D), Quadtree (2D)
    - hierarchical numbering : basis 2,4



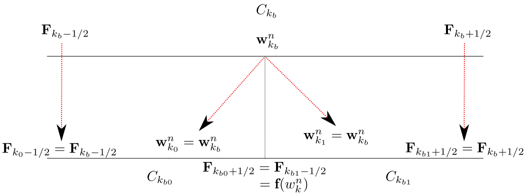


- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely :
  - $S_k^n \leqslant \alpha_{\text{min}} \overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n \geqslant \alpha_{\text{max}} \overline{S} \Longrightarrow$ the cell is coarsened
  - Dynamic mesh refinement :
    - ⋆ Non-structured grid : macro-cell
    - ⋆ Dyadic tree (1D), Quadtree (2D), Octree (3D)
    - ⋆ hierarchical numbering : basis 2,4,8

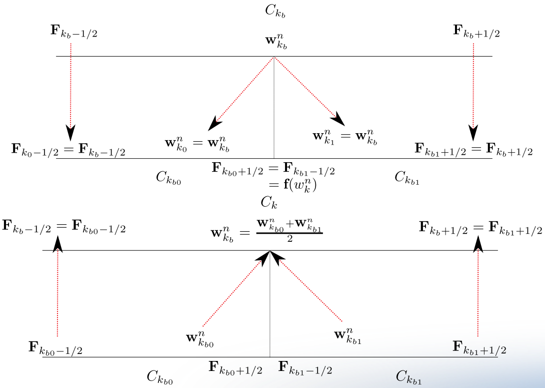




- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely :
  - $S_k^n \leqslant \alpha_{\mathsf{min}} \overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n \geqslant \alpha_{\mathsf{max}} \overline{S} \Longrightarrow$ the cell is coarsened



- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \implies$ the cell is refined or coarsened
- More precisely :
  - $S_k^n \leqslant \alpha_{\text{min}}\overline{S} \implies$ the cell is refined with $\overline{S} = \frac{1}{|\Omega|}\int_\Omega S_k^n$
  - $S_k^n \geqslant \alpha_{\text{max}}\overline{S} \implies$ the cell is coarsened



$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} &= 0 \\
\frac{\partial \rho u}{\partial t} + \frac{\partial \left(\rho u^2 + p\right)}{\partial x} &= 0 \\
\frac{\partial \rho E}{\partial t} + \frac{\partial \left(\rho E + p\right) u}{\partial x} &= 0 \\
p &= (\gamma - 1)\rho\varepsilon
\end{aligned}$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$\gamma := 1.4$	:	ratio of the specific heats
$E(\varepsilon,u)$	:	total energy
ε	:	internal specific energy
E	$=$	$\varepsilon + \frac{u^2}{2}$

Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} &= 0 \\
\frac{\partial \rho u}{\partial t} + \frac{\partial \left(\rho u^2 + p\right)}{\partial x} &= 0 \\
\frac{\partial \rho E}{\partial t} + \frac{\partial \left(\rho E + p\right) u}{\partial x} &= 0 \\
p &= (\gamma - 1)\rho\varepsilon
\end{aligned}$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$\gamma := 1.4$	:	ratio of the specific heats
$E(\varepsilon, u)$	:	total energy
ε	:	internal specific energy
E	$=$	$\varepsilon + \frac{u^2}{2}$

- Conservative variables

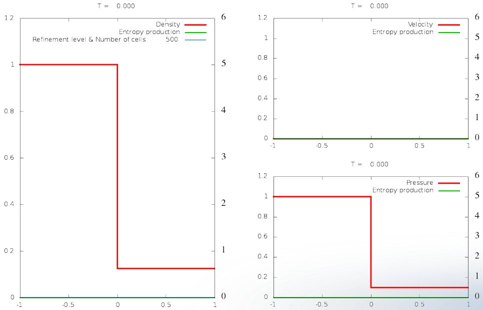
$$\boldsymbol{w} = (\rho, \rho u, \rho E)^t$$

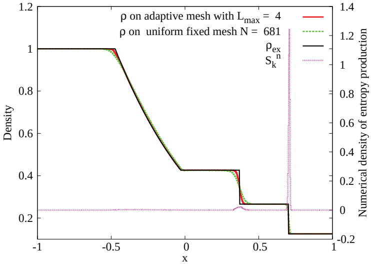
- convex continuous entropy

$$s(\boldsymbol{w}) = -\rho \ln\left(\frac{p}{\rho^\gamma}\right) \text{ of flux } \psi(\boldsymbol{w}) = u\, s(\boldsymbol{w}) \;.$$

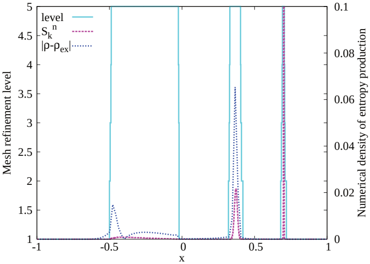
Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz

Mesh refinement parameter $\alpha_{\max}$	:	0.01 ,
Mesh coarsening parameter $\alpha_{\min}$	:	0.001 ,
Mesh refinement parameter $\bar{S}$	:	$\frac{1}{\lvert\Omega\rvert}\sum_{k_b} S^n_{k_b}$
CFL	:	0.25,
Simulation time (s)	:	0.4,
Initial number of cells	:	200,
Maximum level of mesh refinement	:	$L_{\max}$.





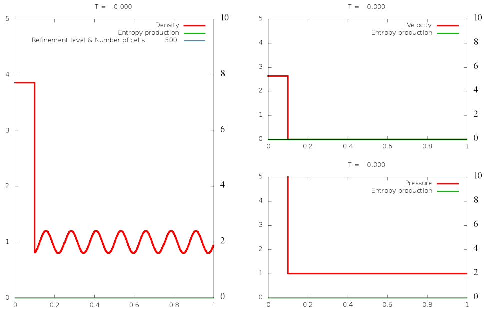
(a) Density and numerical density of entropy production.

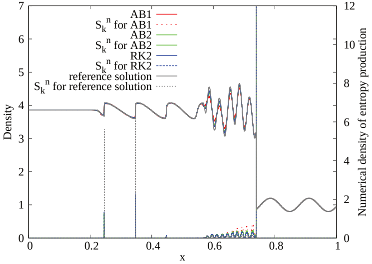


(b) Mesh refinement level, numerical density of entropy production and local error.

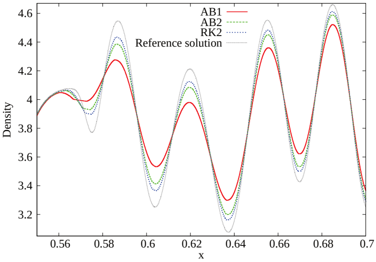
FIGURE : Sod's shock tube problem : solution at time $t = 0.4$ s using the AB1M scheme on a dynamic grid with $L_{\max} = 5$ and the AB1 scheme on a uniform fixed grid of 681 cells.

CFL	:	$0.219,$
Simulation time (s)	:	$0.18,$
Initial number of cells	:	$500,$
Maximum level of mesh refinement	:	$L_{\max} = 4.$





(a) Density and numerical density of entropy production.



(b) Zoom on oscillating region.

Figure : Shu and Osher test case.

- Explicit adaptive schemes : time consuming due to the restriction

$$\|w\|\frac{\delta}{h} \leqslant 1, \quad h = \min_k h_k$$

Müller S., Stiriba Y., *SIAM J. Sci. Comput.*, (07) ; Ersoy M., Golay F., Yushchenko L., *CEJM*, (13) ;

- Explicit adaptive schemes : time consuming due to the restriction

$$\|w\|\frac{\delta}{h} \leqslant 1, \quad h = \min_k h_k$$

- Local time stepping algorithm : save the cpu-time
  - Sort cells in groups w.r.t. to their level

Müller S., Stiriba Y., *SIAM J. Sci. Comput.*, (07) ; Ersoy M., Golay F., Yushchenko L., *CEJM*, (13) ;

- Explicit adaptive schemes : time consuming due to the restriction

$$\|w\|\frac{\delta}{h} \leqslant 1, \quad h = \min_k h_k$$

- Local time stepping algorithm : save the cpu-time
  - Sort cells in groups w.r.t. to their level
  - Update the cells following the local time stepping algorithm.

Müller S., Stiriba Y., *SIAM J. Sci. Comput.*, (07) ; Ersoy M., Golay F., Yushchenko L., *CEJM*, (13) ;

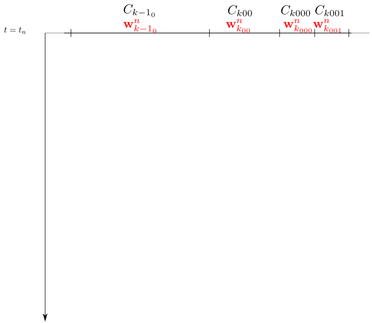
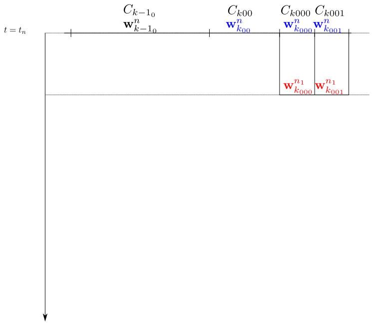


FIGURE : $t = t_n$

with

$$\delta F^n_{k-1,k,k+1} := \left(\boldsymbol{F}^n_{k+1/2}(\boldsymbol{w}_k, \boldsymbol{w}_{k+1}) - \boldsymbol{F}^n_{k-1/2}(\boldsymbol{w}_{k-1}, \boldsymbol{w}_k) \right)$$



$$\boldsymbol{w}^{n_1}_{k_{000}} = \boldsymbol{w}^n_{k_{000}} - \frac{\delta t_n}{h_{k_{000}}} \delta \boldsymbol{F}^n_{k_{00},k_{000},k_{001}}$$

$$\boldsymbol{w}^{n_1}_{k_{001}} = \boldsymbol{w}^n_{k_{001}} - \frac{\delta t_n}{h_{k_{001}}} \delta \boldsymbol{F}^n_{k_{000},k_{001},k+1_b}$$

Figure : $t_{n_1} = t_n + \delta t_n$

with

$$\delta F^n_{k-1,k,k+1} := \left(\boldsymbol{F}^n_{k+1/2}(\boldsymbol{w}_k, \boldsymbol{w}_{k+1}) - \boldsymbol{F}^n_{k-1/2}(\boldsymbol{w}_{k-1}, \boldsymbol{w}_k) \right)$$

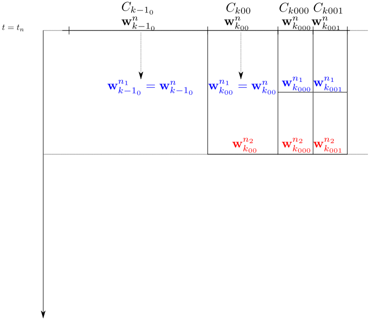


FIGURE : $t_{n_2} = t_n + 2\delta t_n$

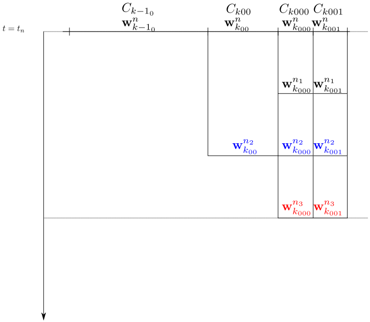
$$\boldsymbol{w}^{n_2}_{k_{00}} = \boldsymbol{w}^{n_1}_{k_{00}} - \frac{\delta t_n}{h_{k_{00}}} \delta \boldsymbol{F}^{n_1}_{k-1_0, k_{00}, k_{000}}$$

$$\boldsymbol{w}^{n_2}_{k_{000}} = \boldsymbol{w}^{n_1}_{k_{000}} - \frac{\delta t_n}{h_{k_{000}}} \delta \boldsymbol{F}^{n_1}_{k_{00}, k_{000}, k_{001}}$$

$$\boldsymbol{w}^{n_2}_{k_{001}} = \boldsymbol{w}^{n_1}_{k_{001}} - \frac{\delta t_n}{h_{k_{001}}} \delta \boldsymbol{F}^{n_1}_{k_{000}, k_{001}, k+1_b}$$

with

$$\delta F^n_{k-1,k,k+1} := \left(\boldsymbol{F}^n_{k+1/2}(\boldsymbol{w}_k, \boldsymbol{w}_{k+1}) - \boldsymbol{F}^n_{k-1/2}(\boldsymbol{w}_{k-1}, \boldsymbol{w}_k) \right)$$



$$\boldsymbol{w}^{n_3}_{k_{000}} = \boldsymbol{w}^{n_2}_{k_{000}} - \frac{\delta t_n}{h_{k_{000}}}\delta \boldsymbol{F}^{n_2}_{k_{00},k_{000},k_{001}}$$

$$\boldsymbol{w}^{n_3}_{k_{001}} = \boldsymbol{w}^{n_2}_{k_{001}} - \frac{\delta t_n}{h_{k_{001}}}\delta \boldsymbol{F}^{n_2}_{k_{000},k_{001},k+1_b}$$

FIGURE : $t_{n_3} = t_n + 3\delta t_n$

with

$$\delta F^n_{k-1,k,k+1} := \left(\boldsymbol{F}^n_{k+1/2}(\boldsymbol{w}_k, \boldsymbol{w}_{k+1}) - \boldsymbol{F}^n_{k-1/2}(\boldsymbol{w}_{k-1}, \boldsymbol{w}_k)\right)$$

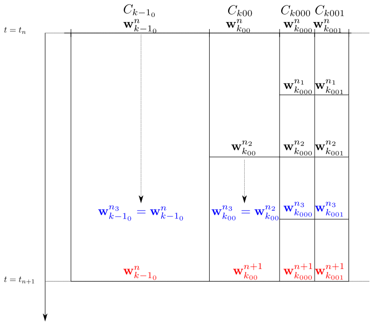


FIGURE : $t_{n+1} = t_n + 4\delta t_n$

$$\boldsymbol{w}^{n+1}_{k-1_0} = \boldsymbol{w}^{n_3}_{k-1_0} - \frac{\delta t_n}{h_{k-1_0}} \delta \boldsymbol{F}^{n_3}_{k-2_b,k-1_0,k_{00}}$$

$$\boldsymbol{w}^{n+1}_{k_{00}} = \boldsymbol{w}^{n_3}_{k_{00}} - \frac{\delta t_n}{h_{k_{00}}} \delta \boldsymbol{F}^{n_3}_{k-1_0,k_{00},k_{000}}$$

$$\boldsymbol{w}^{n+1}_{k_{000}} = \boldsymbol{w}^{n_3}_{k_{000}} - \frac{\delta t_n}{h_{k_{000}}} \delta \boldsymbol{F}^{n_3}_{k_{00},k_{000},k_{001}}$$

$$\boldsymbol{w}^{n+1}_{k_{001}} = \boldsymbol{w}^{n_3}_{k_{001}} - \frac{\delta t_n}{h_{k_{001}}} \delta \boldsymbol{F}^{n_3}_{k_{000},k_{001},k+1_b}$$

with

$$\delta F^n_{k-1,k,k+1} := \left(\boldsymbol{F}^n_{k+1/2}(\boldsymbol{w}_k, \boldsymbol{w}_{k+1}) - \boldsymbol{F}^n_{k-1/2}(\boldsymbol{w}_{k-1}, \boldsymbol{w}_k) \right)$$

foreach $i \in \{1, 2^N\}$ **do**

Let j be the biggest integer such that 2^j divides i

foreach *interface* $x_{k+1/2}$ *such that* $\mathcal{L}_{k+1/2} \geqslant N - j$ **do**

1. compute the integral of $\boldsymbol{F}_{k+1/2}(t)$ on the time interval $2^{N-\mathcal{L}_{k+1/2}}\delta t_n$,
2. distribute $\boldsymbol{F}_{k+1/2}(t_n)$ to the two adjacent cells,
3. update only the cells of level greater than $N - j$.

end

end

	$\mathcal{P}$	$\Vert\rho - \rho_{ref}\Vert_{l^1_x}$	cpu-time	$N_{L_{\max}}$	maximum number of cells
AB1	0.288	$4.74\,10^{-2}$	181	1574	2308

Table : Shu and Osher test case : comparison of numerical schemes of order 1

	$\mathcal{P}$	$\lVert\rho - \rho_{ref}\rVert_{l^1_x}$	cpu-time	$N_{L_{\max}}$	maximum number of cells
AB1	0.288	$4.74\,10^{-2}$	181	1574	2308
AB1M	0.288	$4.80\,10^{-2}$	120	1572	2314

Table : Shu and Osher test case : comparison of numerical schemes of order 1

	$\mathcal{P}$	$\|\rho - \rho_{ref}\|_{l_x^1}$	cpu-time	$N_{L_{\max}}$	maximum number of cells
AB1	0.288	$4.74\,10^{-2}$	181	1574	2308
AB1M	0.288	$4.80\,10^{-2}$	120	1572	2314
AB2	0.287	$2.75\,10^{-2}$	170	1391	2023

Table : Shu and Osher test case : comparison of numerical schemes of order 1 and 2

	$\mathcal{P}$	$\|\rho - \rho_{ref}\|_{l^1_x}$	cpu-time	$N_{L_{\max}}$	maximum number of cells
AB1	0.288	$4.74\,10^{-2}$	181	1574	2308
AB1M	0.288	$4.80\,10^{-2}$	120	1572	2314
AB2	0.287	$2.75\,10^{-2}$	170	1391	2023
AB2M	0.286	$2.74\,10^{-2}$	108	1357	1994

Table : Shu and Osher test case : comparison of numerical schemes of order 1 and 2

	$\mathcal{P}$	$\|\rho - \rho_{ref}\|_{l^1_x}$	cpu-time	$N_{L_{\max}}$	maximum number of cells
AB1	0.288	$4.74\,10^{-2}$	181	1574	2308
AB1M	0.288	$4.80\,10^{-2}$	120	1572	2314
AB2	0.287	$2.75\,10^{-2}$	170	1391	2023
AB2M	0.286	$2.74\,10^{-2}$	108	1357	1994
RK2	0.285	$2.08\,10^{-2}$	299	1375	2005

Table : Shu and Osher test case : comparison of numerical schemes of order 1 and 2

In particular, one has :

Theorem

Consider a p^{th} convergent scheme. Let S_k^n be the corresponding numerical density of entropy production and $\Delta t = \lambda h$ be a fixed time step where h stands for the meshsize.
Then

$$\lim_{n\to\infty} S_k^n = \begin{cases} O(\Delta t^p) & \text{if the solution is smooth,} \\ O\left(\dfrac{1}{\Delta t}\right) & \text{if the solution is discontinuous.} \end{cases}$$

In particular, one has :

Theorem

Consider a p^{th} convergent scheme. Let S_k^n be the corresponding numerical density of entropy production and $\Delta t = \lambda h$ be a fixed time step where h stands for the meshsize.
Then

$$\lim_{n\to\infty} S_k^n = \begin{cases} O(\Delta t^p) & \text{if the solution is smooth,} \\ O\left(\dfrac{1}{\Delta t}\right) & \text{if the solution is discontinuous.} \end{cases}$$

and the following property is satisfied :

Properties

Consider a monotone scheme. Then, for almost every k, every n,

$$S_k^n \leqslant 0.$$

In particular, one has :

Theorem

Consider a p^{th} convergent scheme. Let S_k^n be the corresponding numerical density of entropy production and $\Delta t = \lambda h$ be a fixed time step where h stands for the meshsize.
Then

$$\lim_{n\to\infty} S_k^n = \begin{cases} O(\Delta t^p) & \text{if the solution is smooth,} \\ O\left(\dfrac{1}{\Delta t}\right) & \text{if the solution is discontinuous.} \end{cases}$$

and the following property is satisfied :

Properties

Consider a monotone scheme. Then, for almost every k, every n,

$$S_k^n \leqslant 0.$$

Thus, even if locally S_k^n can take positive value, one has $S_k^n \leqslant C\Delta t^q, \quad q \geqslant p$.

Example

Let us consider the transport equation :

$$\left\{ \begin{array}{lcl} w_t + w_x & = & 0 \\ w(0,x) & = & w_0(x) \end{array} \right.$$

Example

Let us consider the transport equation :

$$\left\{ \begin{array}{rcl} w_t + w_x & = & 0 \\ w(0,x) & = & w_0(x) \end{array} \right.$$

and the Godunov scheme :

$$\left\{ \begin{array}{rcl} w_k^{n+1} & = & w_k^n - \dfrac{\delta t}{\delta x} \left(w_k^n - w_{k-1}^n \right) \\ & & \\ & & \end{array} \right.$$

Example

Let us consider the transport equation :

$$\begin{cases} w_t + w_x & = & 0 \\ w(0,x) & = & w_0(x) \end{cases}$$

and the Godunov scheme :

$$\begin{cases} w_k^{n+1} & = & w_k^n - \dfrac{\delta t}{\delta x}\left(w_k^n - w_{k-1}^n\right) \\ \\ S_k^{n+1} & = & \dfrac{s(w_k^{n+1}) - s(w_k^n)}{\delta t} + \dfrac{\psi(s(w_k^n)) - \psi(s(w_{k-1}^n))}{\delta x} \end{cases}$$

with $s(w) = w^2$ and $\psi(w) = w^2$.

Let us consider the transport equation :

$$\begin{cases} w_t + w_x & = & 0 \\ w(0,x) & = & w_0(x) \end{cases}$$

and the Godunov scheme :

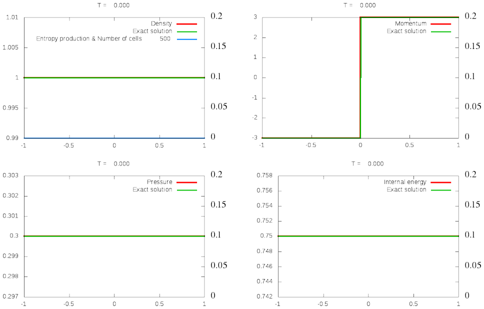
$$\begin{cases} w_k^{n+1} & = & w_k^n - \dfrac{\delta t}{\delta x}\left(w_k^n - w_{k-1}^n\right) \\ S_k^{n+1} & = & \dfrac{s(w_k^{n+1}) - s(w_k^n)}{\delta t} + \dfrac{\psi(s(w_k^n)) - \psi(s(w_{k-1}^n))}{\delta x} \end{cases}$$

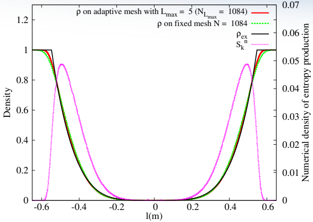
with $s(w) = w^2$ and $\psi(w) = w^2$.

Substituting w_k^{n+1} into S_k^{n+1}, we get

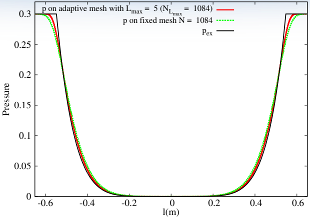
$$S_k^{n+1} = -\varepsilon \left(\frac{w_k^n - w_{k-1}^n}{\delta x}\right)^2 \leqslant 0 \text{ with } \varepsilon = \delta x \left(1 - \frac{\delta t}{\delta x}\right) > 0.$$

CFL	:	$0.25,$
Simulation time (s)	:	$0.15,$
Initial number of cells	:	$200,$
Maximum level of mesh refinement	:	$4.$

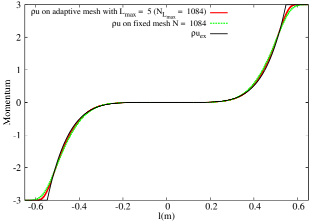




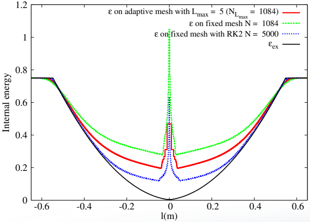
(a) Density and numerical density of entropy production.



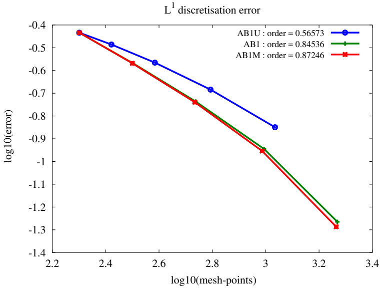
(b) Pressure.



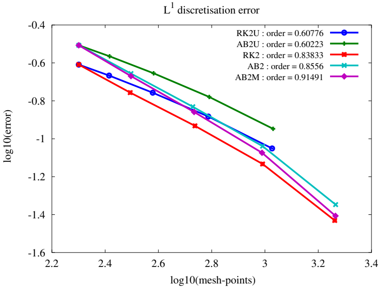
(c) Momentum.



(d) Internal energy.



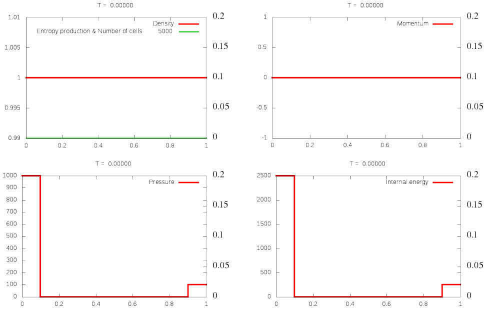
(e) First order scheme.

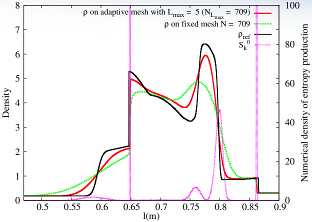


(f) Second order scheme.

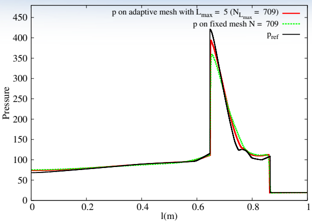
FIGURE : Test 2 : $\|\varepsilon - \varepsilon_{ex}\|_{l^1_x}$ with respect to the average number of cells at time $t = 0.15$.

CFL	:	$0.25,$
Simulation time (s)	:	$0.038,$
Initial number of cells	:	$200,$
Maximum level of mesh refinement	:	$L_{\max}.$

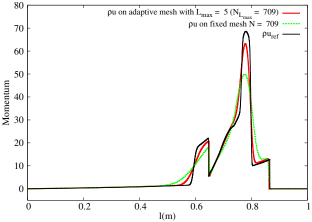




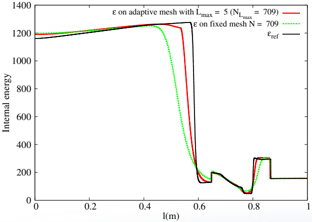
(a) Density and numerical density of entropy production.



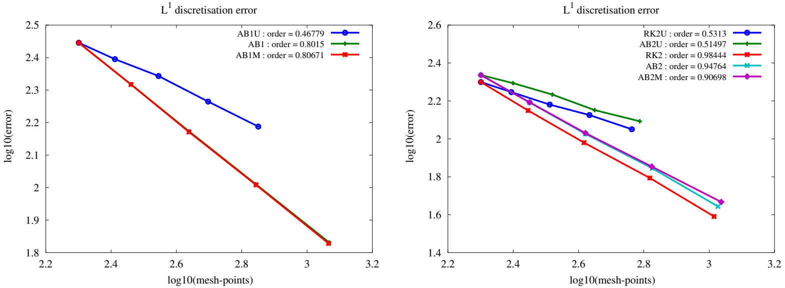
(b) Pressure.



(c) Momentum.



(d) Internal energy.



(e) First order scheme.

(f) Second order scheme.

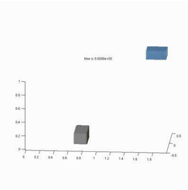
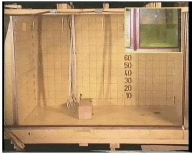
FIGURE : $\|\varepsilon - \varepsilon_{ex}\|_{l^1_x}$ with respect to the average number of cells at time $t = 0.038$.

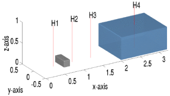
Outline

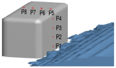
- Main task : wave propagation and wave breaking.

- Main task : wave propagation and wave breaking.
- Reproduce with accuracy saving the cpu-time, previous works by Golay & Helluy and co . . .

- Main task : wave propagation and wave breaking.
- Reproduce with accuracy saving the cpu-time, previous works by Golay & Helluy and co . . .

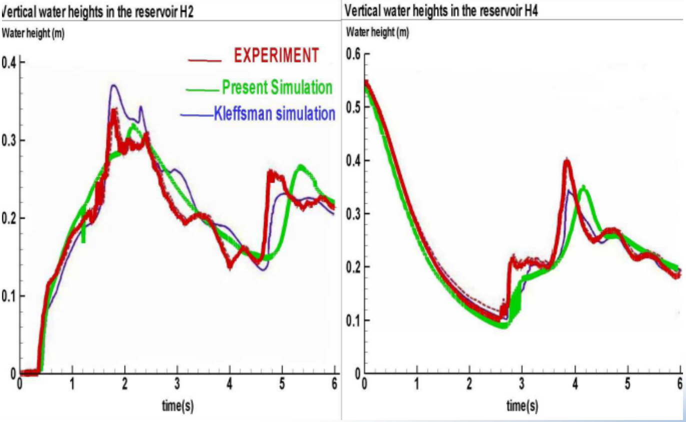







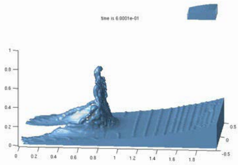
- Main task : wave propagation and wave breaking.
- Reproduce with accuracy saving the cpu-time, previous works by Golay & Helluy and co ...

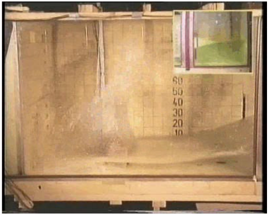
- Main task : wave propagation and wave breaking.
- Reproduce with accuracy saving the cpu-time, previous works by Golay & Helluy and co ...

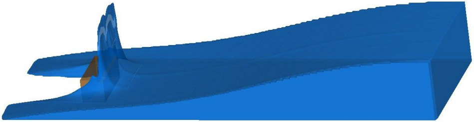


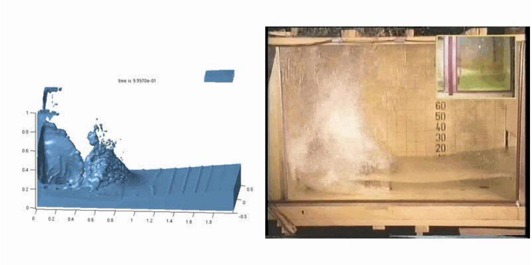
- Main task : wave propagation and wave breaking.
- Reproduce with accuracy saving the cpu-time, previous works by Golay & Helluy and co ...
-

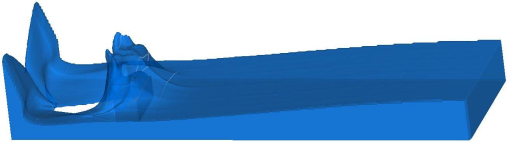
Kleefsmann (ComFlow)	Golay
1.2M cells	0.8M cells
NS+VOF+Surface tension	Bifluid Euler
MAC	FV
	2days CPU M=0.1
	1 day CPU M=0.2


time is 6.0001e-01

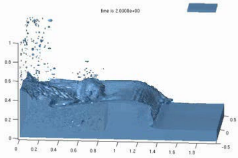

60
50
40
30
20

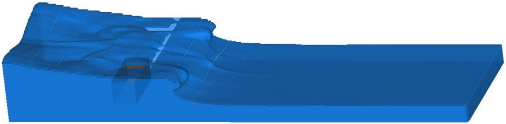



60
50
40
30
20



Application to wave breaking





- Model (2D and 3D) : low mach bi-fluid euler

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \text{div}(\rho u) &= 0\\
\frac{\partial \rho u}{\partial t} + \text{div}\left(\rho u^2 + pI\right) &= \rho g\\
\frac{\partial \rho E}{\partial t} + \text{div}\left((\rho E + p)\, u\right) &= 0\\
\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi &= 0
\end{aligned}
$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$E(\varepsilon,u)$	:	total energy
ε	:	internal specific energy
φ	:	fluid's fraction
E	$=$	$\varepsilon + \frac{u^2}{2}$

- Model (2D and 3D) : low mach bi-fluid euler

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho u) = 0$$
$$\frac{\partial \rho u}{\partial t} + \mathrm{div}\left(\rho u^2 + pI\right) = \rho g$$
$$\frac{\partial \rho E}{\partial t} + \mathrm{div}\left(\left(\rho E + p\right) u\right) = 0$$
$$\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi = 0$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$E(\varepsilon,u)$	:	total energy
ε	:	internal specific energy
φ	:	fluid's fraction
E	$=$	$\varepsilon + \frac{u^2}{2}$

- Mach number $< 0.3 \rightarrow$ fluid is slightly compressible

- Model (2D and 3D) : low mach bi-fluid euler

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho u) = 0$$
$$\frac{\partial \rho u}{\partial t} + \text{div}\left(\rho u^2 + pI\right) = \rho g$$
$$\frac{\partial \rho E}{\partial t} + \text{div}\left((\rho E + p)\, u\right) = 0$$
$$\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi = 0$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$E(\varepsilon,u)$	:	total energy
ε	:	internal specific energy
φ	:	fluid's fraction
E	$=$	$\varepsilon + \frac{u^2}{2}$

- Mach number $< 0.3 \to$ fluid is slightly compressible
- easy to solve

- Model (2D and 3D) : low mach bi-fluid euler

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \text{div}(\rho u) &= 0 \\
\frac{\partial \rho u}{\partial t} + \text{div}\left(\rho u^2 + pI\right) &= \rho g \\
\frac{\partial \rho E}{\partial t} + \text{div}\left((\rho E + p)\,u\right) &= 0 \\
\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi &= 0
\end{aligned}$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$E(\varepsilon,u)$	:	total energy
ε	:	internal specific energy
φ	:	fluid's fraction
E	$=$	$\varepsilon + \frac{u^2}{2}$

- Mach number $< 0.3 \rightarrow$ fluid is slightly compressible
- easy to solve
- Explicit scheme $\rightarrow$ easy parallel implementation (MPI)

- Model (2D and 3D) : low mach bi-fluid euler (isothermal non-cv)

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho u) = 0$$
$$\frac{\partial \rho u}{\partial t} + \text{div}\left(\rho u^2 + pI\right) = \rho g$$
$$\frac{\partial \rho E}{\partial t} + \text{div}\left((\rho E + p)\, u\right) = 0$$
$$\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi = 0$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$E(\varepsilon,u)$	:	total energy
ε	:	internal specific energy
φ	:	fluid's fraction
E	=	$\varepsilon + \frac{u^2}{2}$

with

$$p = p_0 + c_0\left(\rho - (\varphi\rho_w + (1-\varphi)\rho_a)\right)$$

- Mach number < 0.3 → fluid is slightly compressible
- easy to solve
- Explicit scheme → easy parallel implementation (MPI)
- Equation of state with artificial sound speed → CFL less restrictive

- Model (2D and 3D) : low mach bi-fluid euler (isothermal non-cv)

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho u) &= 0\\ \frac{\partial \rho u}{\partial t} + \mathrm{div}\left(\rho u^2 + pI\right) &= \rho g\\ \frac{\partial \rho E}{\partial t} + \mathrm{div}\left((\rho E + p)\, u\right) &= 0\\ \frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi &= 0\end{aligned}$$

where

$$\begin{array}{lcl} \rho(t,x) & : & \text{density}\\ u(t,x) & : & \text{velocity}\\ p(t,x) & : & \text{pressure}\\ E(\varepsilon,u) & : & \text{total energy}\\ \varepsilon & : & \text{internal specific energy}\\ \varphi & : & \text{fluid's fraction}\\ E & = & \varepsilon + \frac{u^2}{2}\end{array}$$

with

$$p = p_0 + c_0\left(\rho - (\varphi\rho_w + (1-\varphi)\rho_a)\right)$$

- Moreover,

- Model (2D and 3D) : low mach bi-fluid euler (isothermal non-cv)

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho u) = 0$$
$$\frac{\partial \rho u}{\partial t} + \text{div}\left(\rho u^2 + pI\right) = \rho g$$
$$\frac{\partial \rho E}{\partial t} + \text{div}\left(\left(\rho E + p\right) u\right) = 0$$
$$\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi = 0$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$E(\varepsilon,u)$	:	total energy
ε	:	internal specific energy
φ	:	fluid's fraction
E	$=$	$\varepsilon + \frac{u^2}{2}$

with

$$p = p_0 + c_0\left(\rho - (\varphi\rho_w + (1-\varphi)\rho_a)\right)$$

✅ hyperbolic system

- Moreover,

- Model (2D and 3D) : low mach bi-fluid euler (isothermal non-cv)

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho u) = 0$$
$$\frac{\partial \rho u}{\partial t} + \mathrm{div}\left(\rho u^2 + pI\right) = \rho g$$
$$\frac{\partial \rho E}{\partial t} + \mathrm{div}\left(\left(\rho E + p\right) u\right) = 0$$
$$\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi = 0$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$E(\varepsilon,u)$	:	total energy
ε	:	internal specific energy
φ	:	fluid's fraction
E	=	$\varepsilon + \frac{u^2}{2}$

with

$$p = p_0 + c_0\left(\rho - (\varphi\rho_w + (1-\varphi)\rho_a)\right)$$

- Moreover,
  - hyperbolic system
  - entropy available

- Model (2D and 3D) : low mach bi-fluid euler (isothermal non-cv)

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho u) = 0$$
$$\frac{\partial \rho u}{\partial t} + \mathrm{div}\left(\rho u^2 + pI\right) = \rho g$$
$$\frac{\partial \rho E}{\partial t} + \mathrm{div}\left(\left(\rho E + p\right) u\right) = 0$$
$$\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi = 0$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$E(\varepsilon,u)$	:	total energy
ε	:	internal specific energy
φ	:	fluid's fraction
E	=	$\varepsilon + \frac{u^2}{2}$

with

$$p = p_0 + c_0\left(\rho - (\varphi\rho_w + (1-\varphi)\rho_a)\right)$$

- hyperbolic system
- entropy available
- automatic mesh refinement

- Moreover,

- Model (2D and 3D) : low mach bi-fluid euler (isothermal non-cv)

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho u) = 0$$
$$\frac{\partial \rho u}{\partial t} + \mathrm{div}\left(\rho u^2 + pI\right) = \rho g$$
$$\frac{\partial \rho E}{\partial t} + \mathrm{div}\left((\rho E + p)\, u\right) = 0$$
$$\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi = 0$$

where

$\rho(t,x)$	:	density
$u(t,x)$	:	velocity
$p(t,x)$	:	pressure
$E(\varepsilon,u)$	:	total energy
ε	:	internal specific energy
φ	:	fluid's fraction
E	$=$	$\varepsilon + \frac{u^2}{2}$

with

$$p = p_0 + c_0\left(\rho - (\varphi \rho_w + (1-\varphi)\rho_a)\right)$$

- Moreover,
  - ☑ hyperbolic system
  - ☑ entropy available
  - ☑ automatic mesh refinement
  - ☑ local time stepping

- hard and main task to handle

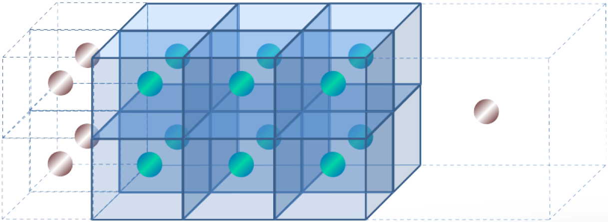
- hard and main task to handle
- strategy : domain, block, cpu ?

- hard and main task to handle
- strategy : domain, block, cpu ?
  1. domain=block=1 cpu : "failure" $\rightarrow$ synchronization depends on the finest domain

- hard and main task to handle
- strategy : domain, block, cpu ?
  1. domain=block=1 cpu : "failure" $\rightarrow$ synchronization depends on the finest domain
  2. domain= n $\times$ blocks = 1cpu : "good compromise" $\rightarrow$ each domain has almost the same number number of cells $\rightarrow$ "better" synchronization

- hard and main task to handle
- strategy : domain, block, cpu ?
  1. domain=block=1 cpu : "failure" $\rightarrow$ synchronization depends on the finest domain
  2. domain= n $\times$ blocks = 1cpu : "good compromise" $\rightarrow$ each domain has almost the same number number of cells $\rightarrow$ "better" synchronization
  3. It certainly exists better strategy . . .

- hard and main task to handle
- strategy : domain, block, cpu ?
  1. domain=block=1 cpu : "failure" → synchronization depends on the finest domain
  2. domain= n × blocks = 1cpu : "good compromise" → each domain has almost the same number number of cells → "better" synchronization
  3. It certainly exists better strategy . . .
- Management of domain's interfaces, projection step, . . .



How it works ?

- each domain has almost the same number of cells

How it works ?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering

DOMAIN= N × BLOCKS = 1CPU

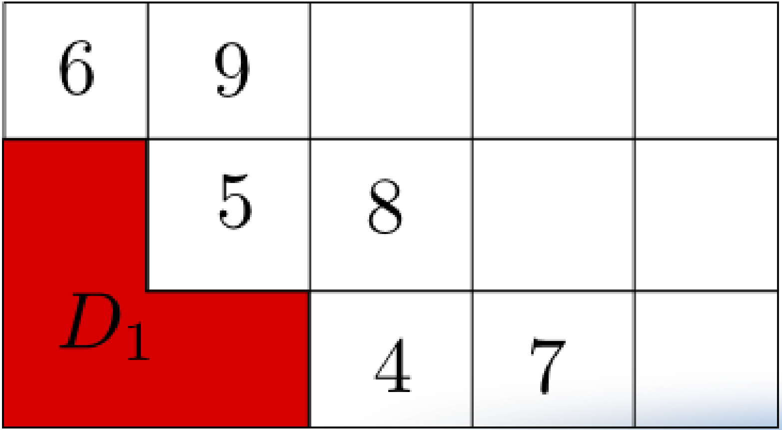
How it works ?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering

6	9			
3	5	8		
1	2	4	7	

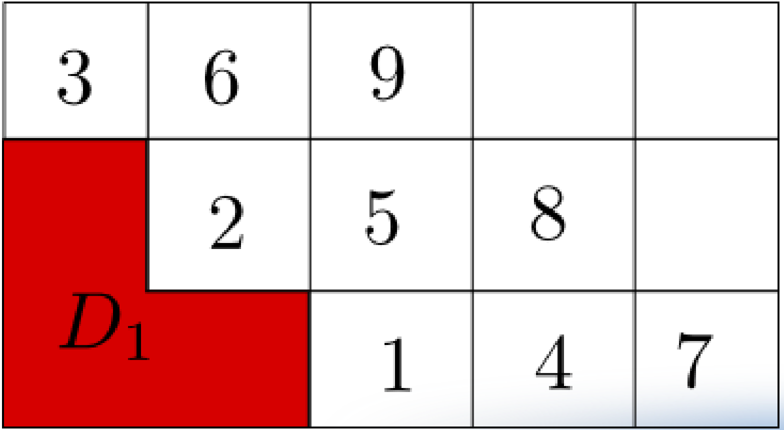
How it works ?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering



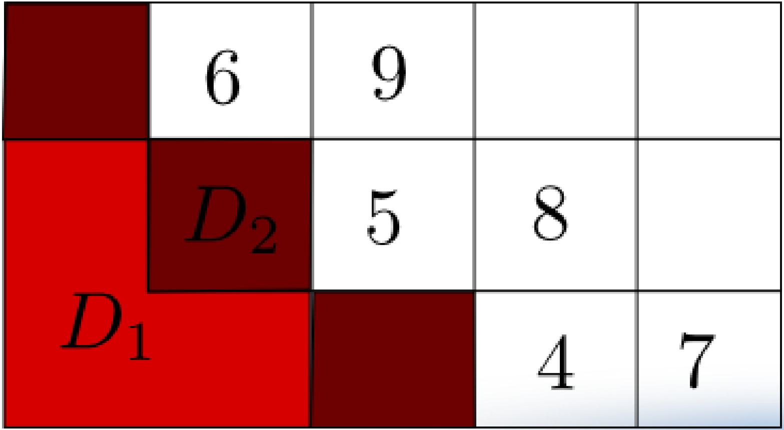
How it works ?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering



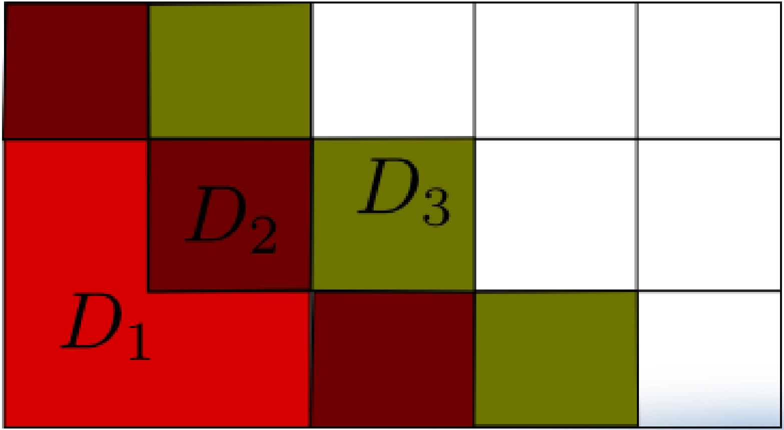
How it works ?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering



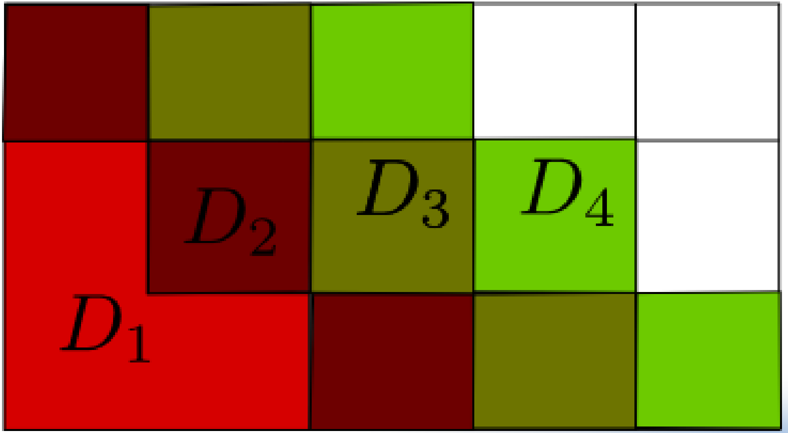
How it works ?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering



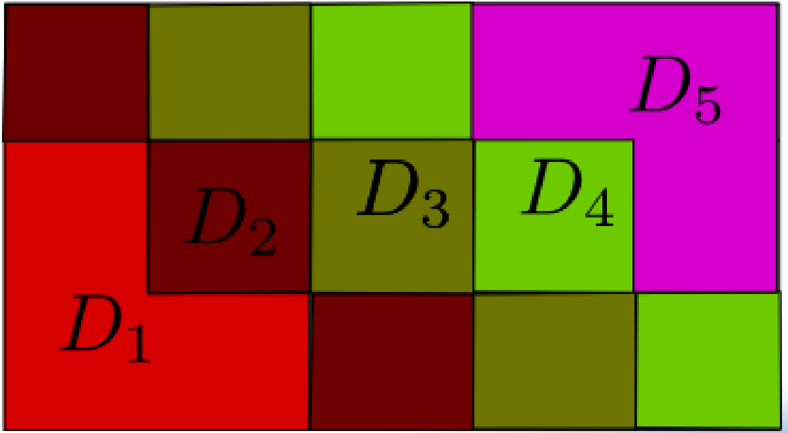
How it works ?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering



How it works ?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering



How it works ?

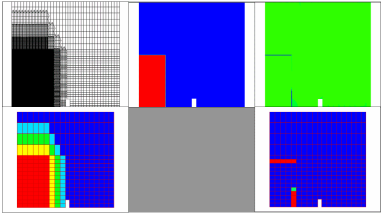
- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering
- more sophisticated numbering exists ...

DOMAIN= N × BLOCKS = 1CPU

How it works ?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering
- more sophisticated numbering exists ...
- main loop and parallelization (mpi)

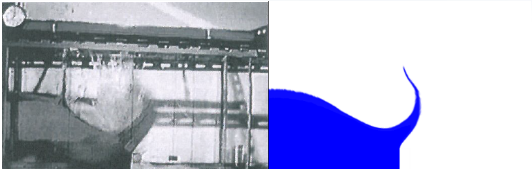
Mesh refinement parameter $\alpha_{\max}$	:	0.2 ,
Mesh coarsening parameter $\alpha_{\min}$	:	0.1 ,
Number of domain	:	321,
Number of processors	:	120,
efinement	:	$L_{\max} = 5$.



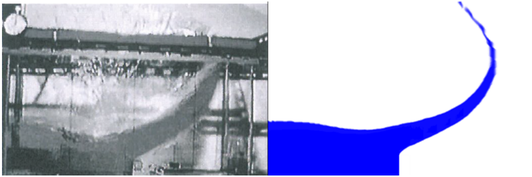
(top left : mesh, top middle : ρ, top right : S_k^n, bottom left : level, bottom right : $\dfrac{1}{|D|}\displaystyle\int_D S_k^n$)

Versus exprimental (Koshizuka, Tamako, Oka, 95)

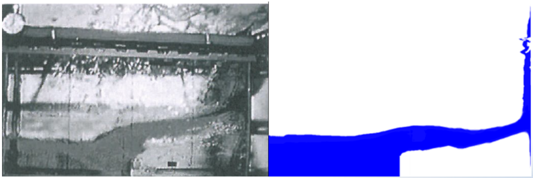
T = 0.2s



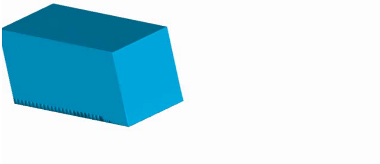
T = 0.3s

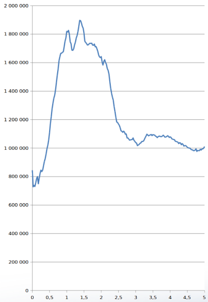


T = 0.4s

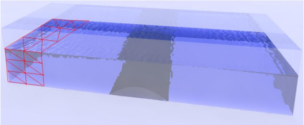
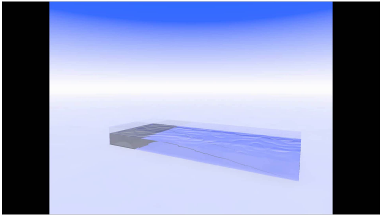


- 10h cpu (instead of 1 day)
- 48 cpus, 48 domains, 3628 blocks
- transfer and post-processing take more time !



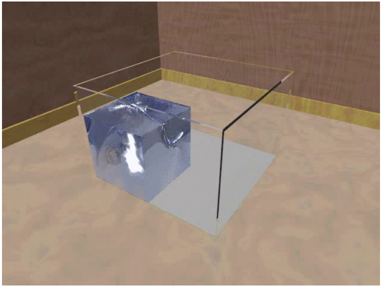


- povray = Persistence Of Vision RAYtracer : high quality and realistic picture
- Povray postprocess is expensive but the results are beautiful ! ! !
- first movie (Shallow water equations with a moving bed) :

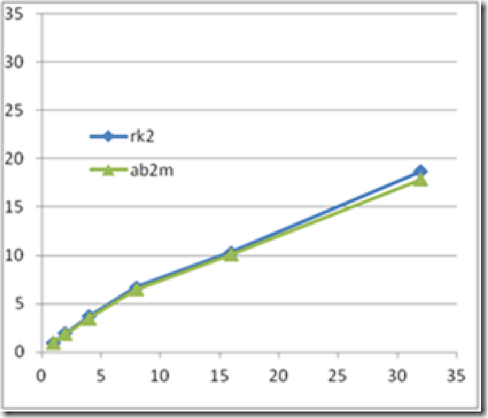
  - each picture ≈ 6Mo
  - time to generate 1 picture ≈ 10 min
  - here 500 picture . . .

- A second movie (bifluid Euler equations) :

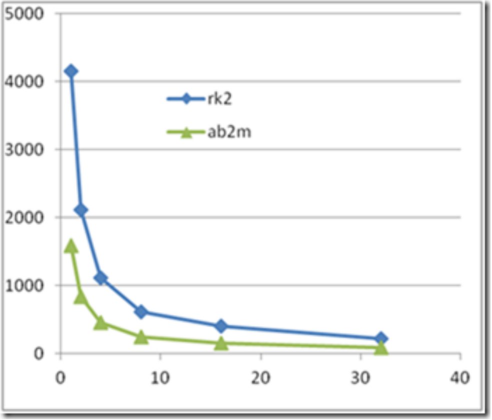


  - 4 level
  - 20 domains
  - 100 time step
  - $\alpha_{\min} = 0.02$, $\alpha_{\max} = 0.2$
  - 172 215 – 587763 cells
  - 7h computation

- speed-up vs proc number



- cpu time vs proc number



2D Euler Riemann problem : a computational challenge (Liska, Wendroff, 01)

- Riemann data :

$$(p, \rho, u, v)(0, x, y) = \begin{cases} (p_1, \rho_1, u_1, v_1), & \text{if} \quad x > 0.5 \quad \text{and} \quad y > 0.5 \\ (p_2, \rho_2, u_2, v_2), & \text{if} \quad x < 0.5 \quad \text{and} \quad y > 0.5 \\ (p_3, \rho_3, u_3, v_3), & \text{if} \quad x < 0.5 \quad \text{and} \quad y < 0.5 \\ (p_4, \rho_4, u_4, v_4), & \text{if} \quad x > 0.5 \quad \text{and} \quad y < 0.5 \end{cases}$$

- Riemann data :

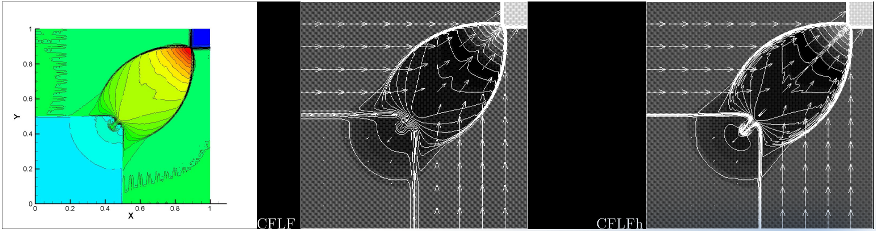
$$(p,\rho,u,v)(0,x,y) = \begin{cases} (p_1,\rho_1,u_1,v_1), & \text{if} \quad x > 0.5 \quad \text{and} \quad y > 0.5 \\ (p_2,\rho_2,u_2,v_2), & \text{if} \quad x < 0.5 \quad \text{and} \quad y > 0.5 \\ (p_3,\rho_3,u_3,v_3), & \text{if} \quad x < 0.5 \quad \text{and} \quad y < 0.5 \\ (p_4,\rho_4,u_4,v_4), & \text{if} \quad x > 0.5 \quad \text{and} \quad y < 0.5 \end{cases}$$

- 19 possible configuration : forward or backward 1 D waves (rarefaction, shock and contact discontinuity)

- Riemann data :

$$(p,\rho,u,v)(0,x,y) = \begin{cases} (0.4, 0.5313, 0, 0), & \text{if} \quad x > 0.5 \quad \text{and} \quad y > 0.5 \\ (1, 1, 0.7276, 0), & \text{if} \quad x < 0.5 \quad \text{and} \quad y > 0.5 \\ (1, 0.8, 0, 0), & \text{if} \quad x < 0.5 \quad \text{and} \quad y < 0.5 \\ (1, 1, 0, 0), & \text{if} \quad x > 0.5 \quad \text{and} \quad y < 0.5 \end{cases}$$

- Resolution of stationary contacts bordering the lower left quadrant

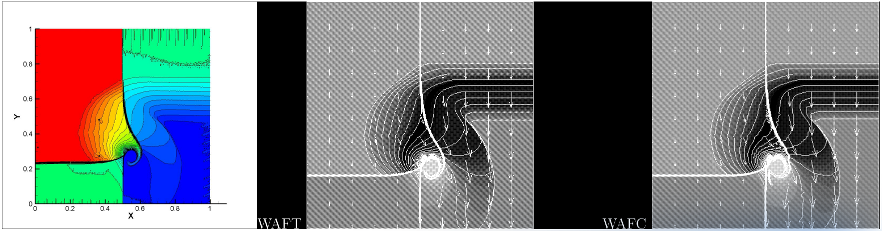


2D Euler Riemann problem : a computational challenge (Liska, Wendroff, 01)

- Riemann data :

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

- Two standing contacts on the line x=0.5



OUTLINE

- low mach bi-fluid model 1D, 2D and 3D

- low mach bi-fluid model 1D, 2D and 3D
- Bi-fluid Euler equations with other pressure law

- low mach bi-fluid model 1D, 2D and 3D
- Bi-fluid Euler equations with other pressure law
- For each model "Efficiency" = accuracy and save the cpu time

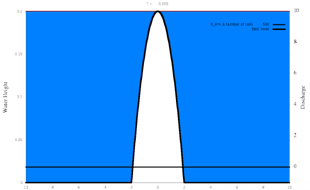
- low mach bi-fluid model 1D, 2D and 3D
- Bi-fluid Euler equations with other pressure law
- For each model "Efficiency" = accuracy and save the cpu time
- New 3D AMR meshing tool implemented

- low mach bi-fluid model 1D, 2D and 3D
- Bi-fluid Euler equations with other pressure law
- For each model "Efficiency" = accuracy and save the cpu time
- New 3D AMR meshing tool implemented
- Others models have been validated : interfacial erosion model with DDFV

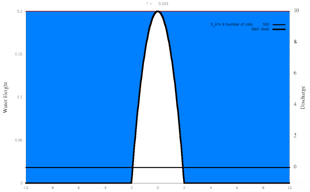
- low mach bi-fluid model 1D, 2D and 3D
- Bi-fluid Euler equations with other pressure law
- For each model "Efficiency" = accuracy and save the cpu time
- New 3D AMR meshing tool implemented
- Others models have been validated : interfacial erosion model with DDFV
- To do
  - optimization of the parallel processing

- low mach bi-fluid model 1D, 2D and 3D
- Bi-fluid Euler equations with other pressure law
- For each model "Efficiency" = accuracy and save the cpu time
- New 3D AMR meshing tool implemented
- Others models have been validated : interfacial erosion model with DDFV
- To do
  - optimization of the parallel processing
  - GCPU

- low mach bi-fluid model 1D, 2D and 3D
- Bi-fluid Euler equations with other pressure law
- For each model "Efficiency" = accuracy and save the cpu time
- New 3D AMR meshing tool implemented
- Others models have been validated : interfacial erosion model with DDFV
- To do
  - optimization of the parallel processing
  - GCPU
  - Application to Shallow water equations :



- low mach bi-fluid model 1D, 2D and 3D
- Bi-fluid Euler equations with other pressure law
- For each model "Efficiency" = accuracy and save the cpu time
- New 3D AMR meshing tool implemented
- Others models have been validated : interfacial erosion model with DDFV
- To do
  - optimization of the parallel processing
  - GCPU
  - Application to Shallow water equations :



  - ...

Thank you

for your

attention