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

Mehmet Ersoy\textsuperscript{1}, Frédéric Golay\textsuperscript{2} and Lyudmyla Yushchenko\textsuperscript{3}

The Third BCAM Workshop on Computational Mathematics

Bilbao,
July 17-18, 2014

\textsuperscript{1} Mehmet.Ersoy@univ-tln.fr
\textsuperscript{2} Frederic.Golay@univ-tln.fr
\textsuperscript{3} Lyudmyla.Yushchenko@univ-tln.fr
Outline of the talk

1. Physical modeling and numerical motivation

2. 2D and 3D applications

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

2 2D and 3D applications

3 Concluding remarks & perspectives
**Simulation of wave propagation and wave breaking**

- **Shallow water equations**: fast but unable to simulate wave breaking
  - Zaleski, Popinet, Diaz, Dutykh, ...
Simulation of wave propagation and wave breaking

- 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 ...

Low-Mach models (Euler equations): good compromise between physical modeling accuracy and cost

(d) SW  (e) Nkonga (FluidBox) (2009)
Simulation of wave propagation and wave breaking

- 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)
Simulation of wave propagation and wave breaking

- 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)
Simulation of wave propagation and wave breaking

- 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

\[\text{(m) SW} \quad \text{(n) Nkonga (FluidBox) (2009)} \quad \text{(o) Golay & Helluy (2005)}\]
We focus on general non linear hyperbolic conservation laws

\[
\begin{cases}
    \frac{\partial w}{\partial t} + \frac{\partial f(w)}{\partial x} = 0, \quad (t, x) \in \mathbb{R}^+ \times \mathbb{R} \\
    w(0, x) = w_0(x), \quad x \in \mathbb{R}
\end{cases}
\]

\( w \in \mathbb{R}^d \): vector state,
\( f \): flux governing the physical description of the flow.
**Hyperbolic equations and entropy condition**

We focus on general non linear hyperbolic conservation laws

\[
\begin{align*}
\frac{\partial w}{\partial t} + \frac{\partial f(w)}{\partial x} &= 0, \quad (t, x) \in \mathbb{R}^+ \times \mathbb{R} \\
w(0, x) &= w_0(x), \quad x \in \mathbb{R}
\end{align*}
\]

Weak solutions satisfy

\[
S = \frac{\partial s(w)}{\partial t} + \frac{\partial \psi(w)}{\partial x} \begin{cases} = 0 & \text{for smooth solution} \\
= 0 & \text{across rarefaction} \\
< 0 & \text{across shock}
\end{cases}
\]

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

\[
(\nabla \psi(w))^T = (\nabla s(w))^T \quad D_w f(w)
\]

Application to wave-breaking

Bilbao, July 17-18, 2014 5 / 38
We focus on general non linear hyperbolic conservation laws

\[
\begin{align*}
\frac{\partial w}{\partial t} + \frac{\partial f(w)}{\partial x} &= 0, \quad (t, x) \in \mathbb{R}^+ \times \mathbb{R} \\
w(0, x) &= w_0(x), \quad x \in \mathbb{R}
\end{align*}
\]

Weak solutions satisfy

\[
S = \frac{\partial s(w)}{\partial t} + \frac{\partial \psi(w)}{\partial x}
\]

\[
\begin{align*}
= 0 & \quad \text{for smooth solution} \\
= 0 & \quad \text{across rarefaction} \\
< 0 & \quad \text{across shock}
\end{align*}
\]

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

\[
(\nabla \psi(w))^T = (\nabla s(w))^T \ D_w f(w)
\]

Entropy inequality \(\simeq\) "smoothness indicator"
We focus on general non linear hyperbolic conservation laws

\[ \begin{cases} 
\frac{\partial w}{\partial t} + \text{div}(f(w)) = 0, & (t, x) \in \mathbb{R}^+ \times \mathbb{R}^d \\
w(0, x) = w_0(x), & x \in \mathbb{R}^d 
\end{cases} \]

Weak solutions satisfy

\[ S = \frac{\partial s(w)}{\partial t} + \text{div}(\psi(w)) \begin{cases} 
= 0 & \text{for smooth solution} \\
= 0 & \text{across rarefaction} \\
< 0 & \text{across shock}
\end{cases} \]

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

\[ (\nabla \psi_i(w))^T = (\nabla s(w))^T D_w f_i(w), \quad i = 1, \ldots, d \]

Entropy inequality \(\simeq\) "smoothness indicator"
Finite volume approximation:

\[ w_{k}^{n+1} = w_{k}^{n} - \frac{\delta t_{n}}{h_{k}} \left( F_{k+1/2}^{n} - F_{k-1/2}^{n} \right) \]

with

\[ w_{k}^{n} \approx \frac{1}{h_{k}} \int_{C_{k}} w(t_{n}, x) \, dx \]
Finite volume approximation:

\[ w^{n+1}_k = w^n_k - \frac{\delta t_n}{h_k} \left( F^n_{k+1/2} - F^n_{k-1/2} \right) \]

with

\[ w^n_k \approx \frac{1}{h_k} \int_{C_k} w(t_n, x) \, dx \]

The numerical density of entropy production:

\[ S^n_k = \frac{s^{n+1}_k - s^n_k}{\delta t_n} + \frac{\psi^n_{k+1/2} - \psi^n_{k-1/2}}{h_k} \approx 0 \]
Mesh refinement indicator: principle & illustration

- Compute $w_k^n$

\[ S_n^k \neq 0 \Rightarrow \text{the cell is refined or coarsened} \]

More precisely:

- $S_n^k \leq \alpha_{\text{min}} S_n = 1 | \Omega | \int_\Omega S_n^k \Rightarrow \text{the cell is refined with } S_n^k = 1$

- $S_n^k \geq \alpha_{\text{max}} S_n = 1 \Rightarrow \text{the cell is coarsened}$
Mesh refinement indicator: principle & illustration

- Compute $w_k^n$
- Compute $S_k^n$: $S_k^n \neq 0 \implies$ the cell is refined or coarsened
Mesh refinement indicator: principle & illustration

- Compute $w^n_k$
- Compute $S^n_k: S^n_k \neq 0 \implies$ the cell is refined or coarsened
- More precisely:
  - $S^n_k \leq \alpha_{\text{min}} \overline{S} \implies$ the cell is refined with $\overline{S} = \frac{1}{|\Omega|} \int_{\Omega} S^n_k$
  - $S^n_k \geq \alpha_{\text{max}} \overline{S} \implies$ the cell is coarsened
Mesh refinement indicator: principle & illustration

- Compute $w_k^n$
- Compute $S_k^n : S_k^n \neq 0 \implies$ the cell is refined or coarsened
- More precisely:
  - $S_k^n \leq \alpha_{\min} \overline{S} \implies$ the cell is refined with $\overline{S} = \frac{1}{|\Omega|} \int_{\Omega} S_k^n$
  - $S_k^n \geq \alpha_{\max} \overline{S} \implies$ the cell is coarsened
Mesh refinement indicator: principle & illustration

- Compute $w^n_k$
- Compute $S^n_k: S^n_k \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely:
  - $S^n_k \leq \alpha_{\min} \overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \frac{1}{|\Omega|} \int_{\Omega} S^n_k$
  - $S^n_k \geq \alpha_{\max} \overline{S} \Longrightarrow$ the cell is coarsened
  - Dynamic mesh refinement:

  ⋆ Dyadic tree (1D)
  ⋆ hierarchical numbering: basis 2
Mesh refinement indicator: principle & illustration

- Compute $w^k_n$
- Compute $S^n_k : S^n_k \neq 0 \Rightarrow$ the cell is refined or coarsened
- More precisely:
  - $S^n_k \leq \alpha_{\min} \bar{S} \Rightarrow$ the cell is refined with $\bar{S} = \frac{1}{|\Omega|} \int_{\Omega} S^n_k$
  - $S^n_k \geq \alpha_{\max} \bar{S} \Rightarrow$ the cell is coarsened
  - Dynamic mesh refinement:
    - Non-structured grid: macro-cell
    - Dyadic tree (1D), Quadtree (2D)
    - Hierarchical numbering: basis 2, 4

![Mesh refinement diagram]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>120</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td></td>
<td>122</td>
<td>123</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
</tr>
</thead>
</table>

M. Ersoy (IMATH)
Application to wave-breaking
Bilbao, July 17-18, 2014 7 / 38
Mesh refinement indicator: principle & illustration

- Compute $w^n_k$
- Compute $S^n_k : S^n_k \neq 0 \implies$ the cell is refined or coarsened
- More precisely:
  - $S^n_k \leq \alpha_{\min} \overline{S} \implies$ the cell is refined with \( \overline{S} = \frac{1}{|\Omega|} \int_{\Omega} S^n_k \)
  - $S^n_k \geq \alpha_{\max} \overline{S} \implies$ 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
Mesh refinement indicator: principle & illustration

- Compute $w^n_k$
- Compute $S^n_k : S^n_k \neq 0 \implies$ the cell is refined or coarsened
- More precisely:
  - $S^n_k \leq \alpha_{\min} \overline{S} \implies$ the cell is refined with $\overline{S} = \frac{1}{|\Omega|} \int_{\Omega} S^n_k$
  - $S^n_k \geq \alpha_{\max} \overline{S} \implies$ the cell is coarsened

\[
\begin{align*}
C_{k_0} & \quad F_{k_0+1/2} = F_{k_0-1/2} \quad w^n_{k_0} = w^n_{k_0} \quad w^n_{k_1} = w^n_{k_0} \quad F_{k_1+1/2} = F_{k_1+1/2} \\
C_{k_0} & \quad F_{k_0+1/2} = F_{k_0-1/2} = f(w^n_k) \quad C_{k_1}
\end{align*}
\]
Mesh refinement indicator: principle & illustration

- Compute $w^n_k$
- Compute $S^n_k : S^n_k \neq 0 \implies$ the cell is refined or coarsened
- More precisely:
  - $S^n_k \leq \alpha_{\text{min}} \overline{S} \implies$ the cell is refined with $\overline{S} = \frac{1}{|\Omega|} \int_\Omega S^n_k$
  - $S^n_k \geq \alpha_{\text{max}} \overline{S} \implies$ the cell is coarsened
ONE-DIMENSIONAL GAS DYNAMICS EQUATIONS FOR IDEAL GAS

\[
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0
\]

\[
\frac{\partial \rho u}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} = 0
\]

\[
\frac{\partial \rho E}{\partial t} + \frac{\partial (\rho E + p) u}{\partial x} = 0
\]

\[
p = (\gamma - 1) \rho \varepsilon
\]

where

\[
\rho(t, x) \quad : \quad \text{density}
\]

\[
u(t, x) \quad : \quad \text{velocity}
\]

\[
p(t, x) \quad : \quad \text{pressure}
\]

\[
\gamma := 1.4 \quad : \quad \text{ratio of the specific heats}
\]

\[
E(\varepsilon, u) \quad : \quad \text{total energy}
\]

\[
\varepsilon \quad : \quad \text{internal specific energy}
\]

\[
E = \varepsilon + \frac{u^2}{2}
\]
ONE-DIMENSIONAL GAS DYNAMICS EQUATIONS FOR IDEAL GAS

\[ \frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0 \]

\[ \frac{\partial \rho u}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} = 0 \]

\[ \frac{\partial \rho E}{\partial t} + \frac{\partial (\rho E + p) u}{\partial x} = 0 \]

where

\[ p = (\gamma - 1) \rho \varepsilon \]

\[ \rho(t, x) : \text{density} \]

\[ u(t, x) : \text{velocity} \]

\[ p(t, x) : \text{pressure} \]

\[ \gamma := 1.4 : \text{ratio of the specific heats} \]

\[ E(\varepsilon, u) : \text{total energy} \]

\[ \varepsilon : \text{internal specific energy} \]

\[ E = \varepsilon + \frac{u^2}{2} \]

- Conservative variables

\[ w = (\rho, \rho u, \rho E)^t \]

- Convex continuous entropy

\[ s(w) = -\rho \ln \left( \frac{p}{\rho \gamma} \right) \text{ of flux } \psi(w) = u s(w) \]
Sod’s shock tube problem

Mesh refinement parameter $\alpha_{\text{max}}$: 0.01,
Mesh coarsening parameter $\alpha_{\text{min}}$: 0.001,
Mesh refinement parameter $\bar{S}$: $\frac{1}{|\Omega|} \sum_{k_b} S_{k_b}$

CFL: 0.25,
Simulation time (s): 0.4,
Initial number of cells: 200,
Maximum level of mesh refinement: $L_{\text{max}}$. 
(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 $L_{\text{max}} = 5$ and the AB1 scheme on a uniform fixed grid of 681 cells.
**Shu and Osher test case**

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

---

**Diagram**

- Line graphs showing density, entropy production, and flow solutions at different times.

---

**M. Ersoy (IMATH)**

Application to wave-breaking

Bilbao, July 17-18, 2014
(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

\[ \|w\| \frac{\delta}{h} \leq 1, \quad h = \min_k h_k \]
Explicit adaptive schemes: time consuming due to the restriction

\[ \|w\| \frac{\delta}{h} \leq 1, \quad h = \min_k h_k \]

Local time stepping algorithm: save the cpu-time

- Sort cells in groups w.r.t. to their level

Explicit adaptive schemes: time consuming due to the restriction

\[ \| w \| \frac{\delta}{h} \leq 1, \quad h = \min_k h_k \]

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.

Figure: $t = t_n$

with

$$\delta F_{k-1,k,k+1}^n := \left( F_{k+1/2}^n(w_k, w_{k+1}) - F_{k-1/2}^n(w_{k-1}, w_k) \right)$$
Illustration

\[ t_{n_1} = t_n + \delta t_n \]

with

\[ \delta F_{k-1,k,k+1}^n := \left( F_{k+1/2}^n (w_k, w_{k+1}) - F_{k-1/2}^n (w_{k-1}, w_k) \right) \]

\[ w_{k000}^n = w_{k000}^{n-1} - \frac{\delta t_n}{h_{k000}} \delta F_{k000,k000,k001}^n \]

\[ w_{k001}^n = w_{k001}^{n-1} - \frac{\delta t_n}{h_{k001}} \delta F_{k000,k001,k+1b}^n \]
Illustration

Figure: $t_{n2} = t_n + 2\delta t_n$

with

$$\delta F_{k-1,k,k+1}^n := \left( F_{k+1/2}^n(w_k, w_{k+1}) - F_{k-1/2}^n(w_{k-1}, w_k) \right)$$
Illustration

Figure: $t_{n3} = t_n + 3\delta t_n$

with

$$\delta F_{k-1,k,k+1}^n := \left( F_{k+1/2}^n(w_k, w_{k+1}) - F_{k-1/2}^n(w_{k-1}, w_k) \right)$$
\[ w_{n+1}^{k-1} = w_{k-1}^n - \frac{\delta t_n}{h_{k-1}} \delta F_{n}^{k-2, k-1, k0} \]

\[ w_{n+1}^{k0} = w_{k0}^n - \frac{\delta t_n}{h_{k0}} \delta F_{n}^{k-1, k0, k0} \]

\[ w_{n+1}^{k00} = w_{k00}^n - \frac{\delta t_n}{h_{k00}} \delta F_{n}^{k0, k00, k01} \]

\[ w_{n+1}^{k001} = w_{k001}^n - \frac{\delta t_n}{h_{k001}} \delta F_{n}^{k0, k001, k+1} \]

with

\[ \delta F_{n}^{k-1, k, k+1} := \left( F_{n}^{k+1/2}(w_k, w_{k+1}) - F_{n}^{k-1/2}(w_{k-1}, w_k) \right) \]
**LOCAL TIME STEPPING ALGORITHM**

\[
\text{foreach } i \in \{1, 2^N\} \text{ do} \\
\quad \text{Let } j \text{ be the biggest integer such that } 2^j \text{ divides } i \\
\quad \text{foreach interface } x_{k+1/2} \text{ such that } L_{k+1/2} \geq N - j \text{ do} \\
\quad 
\begin{enumerate}
\item compute the integral of } F_{k+1/2}(t) \text{ on the time interval } 2^{N-L_{k+1/2}} \delta t_n, \\
\item distribute } F_{k+1/2}(t_n) \text{ to the two adjacent cells,} \\
\item update only the cells of level greater than } N - j. \\
\end{enumerate}
\text{end}
\text{end}
**Efficiency of the local time stepping method**

<table>
<thead>
<tr>
<th></th>
<th>$\mathcal{P}$</th>
<th>$|\rho - \rho_{ref}|_{L_1}^{1}$</th>
<th>cpu-time</th>
<th>$N_{L_{max}}$</th>
<th>maximum number of cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB1</td>
<td>0.288</td>
<td>4.74 $10^{-2}$</td>
<td>181</td>
<td>1574</td>
<td>2308</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table**: Shu and Osher test case: comparison of numerical schemes of order 1
Efficiency of the local time stepping method

<table>
<thead>
<tr>
<th></th>
<th>( P )</th>
<th>( | \rho - \rho_{ref} |_{l^1_x} )</th>
<th>cpu-time</th>
<th>( N_{L_{max}} )</th>
<th>maximum number of cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB1</td>
<td>0.288</td>
<td>( 4.74 \times 10^{-2} )</td>
<td>181</td>
<td>1574</td>
<td>2308</td>
</tr>
<tr>
<td>AB1M</td>
<td>0.288</td>
<td>( 4.80 \times 10^{-2} )</td>
<td>120</td>
<td>1572</td>
<td>2314</td>
</tr>
</tbody>
</table>

Table: Shu and Osher test case: comparison of numerical schemes of order 1
### Efficiency of the Local Time Stepping Method

<table>
<thead>
<tr>
<th></th>
<th>$P$</th>
<th>$| \rho - \rho_{ref} |_{L_x^1}$</th>
<th>cpu-time</th>
<th>$N_{L_{\text{max}}}$</th>
<th>maximum number of cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB1</td>
<td>0.288</td>
<td>$4.74 \times 10^{-2}$</td>
<td>181</td>
<td>1574</td>
<td>2308</td>
</tr>
<tr>
<td>AB1M</td>
<td>0.288</td>
<td>$4.80 \times 10^{-2}$</td>
<td>120</td>
<td>1572</td>
<td>2314</td>
</tr>
<tr>
<td>AB2</td>
<td>0.287</td>
<td>$2.75 \times 10^{-2}$</td>
<td>170</td>
<td>1391</td>
<td>2023</td>
</tr>
</tbody>
</table>

**Table**: Shu and Osher test case: comparison of numerical schemes of order 1 and 2
Efficiency of the local time stepping method

<table>
<thead>
<tr>
<th></th>
<th>$P$</th>
<th>$|\rho - \rho_{ref}|_{L_x}$</th>
<th>cpu-time</th>
<th>$N_{L_{\text{max}}}$</th>
<th>maximum number of cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB1</td>
<td>0.288</td>
<td>$4.74 \times 10^{-2}$</td>
<td>181</td>
<td>1574</td>
<td>2308</td>
</tr>
<tr>
<td>AB1M</td>
<td>0.288</td>
<td>$4.80 \times 10^{-2}$</td>
<td>120</td>
<td>1572</td>
<td>2314</td>
</tr>
<tr>
<td>AB2</td>
<td>0.287</td>
<td>$2.75 \times 10^{-2}$</td>
<td>170</td>
<td>1391</td>
<td>2023</td>
</tr>
<tr>
<td>AB2M</td>
<td>0.286</td>
<td>$2.74 \times 10^{-2}$</td>
<td>108</td>
<td>1357</td>
<td>1994</td>
</tr>
</tbody>
</table>

Table: Shu and Osher test case: comparison of numerical schemes of order 1 and 2
## Efficiency of the local time stepping method

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>$|\rho - \rho_{ref}|_{L_1}$</th>
<th>cpu-time</th>
<th>$N_{L_{\text{max}}}$</th>
<th>maximum number of cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB1</td>
<td>0.288</td>
<td>$4.74 \times 10^{-2}$</td>
<td>181</td>
<td>1574</td>
<td>2308</td>
</tr>
<tr>
<td>AB1M</td>
<td>0.288</td>
<td>$4.80 \times 10^{-2}$</td>
<td>120</td>
<td>1572</td>
<td>2314</td>
</tr>
<tr>
<td>AB2</td>
<td>0.287</td>
<td>$2.75 \times 10^{-2}$</td>
<td>170</td>
<td>1391</td>
<td>2023</td>
</tr>
<tr>
<td>AB2M</td>
<td>0.286</td>
<td>$2.74 \times 10^{-2}$</td>
<td>108</td>
<td>1357</td>
<td>1994</td>
</tr>
<tr>
<td>RK2</td>
<td>0.285</td>
<td>$2.08 \times 10^{-2}$</td>
<td>299</td>
<td>1375</td>
<td>2005</td>
</tr>
</tbody>
</table>

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

In particular, one has:

**Theorem**

Consider a $p^{\text{th}}$ 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

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

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

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

and the following property is satisfied:

Properties

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

\[ S_k^n \leq 0. \]
**Properties**

In particular, one has:

**Theorem**

Consider a $p^{th}$ convergent scheme. Let $S^n_k$ 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

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

and the following property is satisfied:

**Properties**

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

$$S^n_k \leq 0.$$

Thus, even if locally $S^n_k$ can take positive value, one has $S^n_k \leq C \Delta t^q, \quad q \geq p$. 

M. Ersoy (IMATH)
Example

Let us consider the transport equation:

\[
\begin{align*}
    w_t + w_x &= 0 \\
    w(0, x) &= w_0(x)
\end{align*}
\]
**Example**

Let us consider the transport equation:

\[
\begin{aligned}
&w_t + w_x = 0 \\
&w(0, x) = w_0(x)
\end{aligned}
\]

and the Godunov scheme:

\[
\begin{aligned}
&w_{n+1}^k = w_n^k - \frac{\delta t}{\delta x} (w_n^k - w_n^{k-1}) \\
&\text{with} \\
&s(w) = w^2 \\
&\psi(w) = w^2
\end{aligned}
\]

Substituting \(w_{n+1}^k\) into \(S_{n+1}^k\), we get:

\[
S_{n+1}^k = -\varepsilon (w_n^k - w_n^{k-1})^2 \leq 0
\]

with \(\varepsilon = \delta x (1 - \frac{\delta t}{\delta x}) > 0\).
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:

\[
\begin{cases}
  w_{k}^{n+1} &= w_k^n - \frac{\delta t}{\delta x} (w_k^n - w_{k-1}^n) \\
  S_{k}^{n+1} &= \frac{s(w_{k}^{n+1}) - s(w_k^n)}{\delta t} + \frac{\psi(s(w_{k}^{n+1})) - \psi(s(w_{k-1}^n))}{\delta x}
\end{cases}
\]

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

Let us consider the transport equation:

\[
\begin{align*}
    w_t + w_x &= 0 \\
    w(0, x) &= w_0(x)
\end{align*}
\]

and the Godunov scheme:

\[
\begin{align*}
    w_{k}^{n+1} &= w_k^n - \frac{\delta t}{\delta x} (w_k^n - w_{k-1}^n) \\
    S_{k}^{n+1} &= \frac{s(w_{k-1}^{n+1}) - s(w_k^n)}{\delta t} + \frac{\psi(s(w_k^n)) - \psi(s(w_{k-1}^n))}{\delta x}
\end{align*}
\]

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

Substituting \( w_{k}^{n+1} \) into \( S_{k}^{n+1} \), we get

\[
S_{k}^{n+1} = -\varepsilon \left( \frac{w_k^n - w_{k-1}^n}{\delta x} \right)^2 \leq 0 \quad \text{with} \quad \varepsilon = \delta x \left( 1 - \frac{\delta t}{\delta x} \right) > 0.
\]
CFL
Simulation time (s) : 0.25,
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.
Figure: Test 2: \( \| \varepsilon - \varepsilon_{ex} \|_{L^1_x} \) with respect to the average number of cells at time \( t = 0.15 \).
The blast wave problem

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

(a) Density and numerical density of entropy production.

(b) Pressure.

(c) Momentum.

(d) Internal energy.
The blast wave problem

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

(e) First order scheme.  
(f) Second order scheme.
1 Physical modeling and numerical motivation

2 2D and 3D applications

3 Concluding remarks & perspectives
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...
Application to wave breaking

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

Kleefsmann (ComFlow)
1.2M cells
NS+VOF+Surface tension
MAC

Golay
0.8M cells
Bifluid Euler
FV
2days CPU M=0.1
1 day CPU M=0.2
APPLICATION TO WAVE BREAKING
APPLICATION TO WAVE BREAKING
APPLICATION TO WAVE BREAKING
Application to wave breaking

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

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \text{div}(\rho u) &= 0 \\
\frac{\partial \rho u}{\partial t} + \text{div} (\rho u^2 + pI) &= \rho g \\
\frac{\partial \rho E}{\partial t} + \text{div} ((\rho E + p) u) &= 0 \\
\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi &= 0
\end{align*}
\]

where

\[
\begin{align*}
\rho(t, x) &: \text{ density} \\
u(t, x) &: \text{ velocity} \\
p(t, x) &: \text{ pressure} \\
E(\varepsilon, u) &: \text{ total energy} \\
\varepsilon &: \text{ internal specific energy} \\
\varphi &: \text{ fluid’s fraction} \\
E &= \varepsilon + \frac{u^2}{2}
\end{align*}
\]

Moreover, hyperbolic system

entropy available

automatic mesh refinement

local time stepping
Application to wave breaking

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

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

where
\[ \rho(t, x) \quad : \quad \text{density} \]
\[ u(t, x) \quad : \quad \text{velocity} \]
\[ p(t, x) \quad : \quad \text{pressure} \]
\[ E(\varepsilon, u) \quad : \quad \text{total energy} \]
\[ \varepsilon \quad : \quad \text{internal specific energy} \]
\[ \varphi \quad : \quad \text{fluid’s fraction} \]
\[ E = \varepsilon + \frac{u^2}{2} \]

Mach number < 0.3 \rightarrow \text{fluid is slightly compressible}
Model (2D and 3D) : low mach bi-fluid euler

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

where
\[ \rho(t, x) : \text{density} \]
\[ u(t, x) : \text{velocity} \]
\[ p(t, x) : \text{pressure} \]
\[ E(\varepsilon, u) : \text{total energy} \]
\[ \varepsilon : \text{internal specific energy} \]
\[ \varphi : \text{fluid’s fraction} \]
\[ E = \varepsilon + \frac{u^2}{2} \]

Mach number < 0.3 \rightarrow \text{fluid is slightly compressible}

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

\[
\frac{\partial \rho}{\partial t} + \text{div}(\rho u) = 0
\]

\[
\frac{\partial \rho u}{\partial t} + \text{div} (\rho u^2 + pI) = \rho g
\]

\[
\frac{\partial \rho E}{\partial t} + \text{div} ((\rho E + p)u) = 0
\]

\[
\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi = 0
\]

where

\[\rho(t, x) : \text{density}\]
\[u(t, x) : \text{velocity}\]
\[p(t, x) : \text{pressure}\]
\[E(\varepsilon, u) : \text{total energy}\]
\[\varepsilon : \text{internal specific energy}\]
\[\varphi : \text{fluid’s fraction}\]
\[E = \varepsilon + \frac{u^2}{2}\]

- Mach number < 0.3 → fluid is slightly compressible
- easy to solve
- **Explicit scheme** → easy parallel implementation (MPI)
Model (2D and 3D) : low mach bi-fluid euler (isothermal non-cv)

\[ \frac{\partial \rho}{\partial t} + \text{div}(\rho u) = 0 \]

\[ \frac{\partial \rho u}{\partial t} + \text{div}(\rho u^2 + pI) = \rho g \]

where

\[ \frac{\partial \rho E}{\partial t} + \text{div}((\rho E + p)u) = 0 \]

\[ \frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi = 0 \]

with

\[ p = p_0 + c_0 (\rho - (\varphi \rho_w + (1 - \varphi) \rho_a)) \]

- Mach number < 0.3 \( \rightarrow \) fluid is slightly compressible
- easy to solve
- Explicit scheme \( \rightarrow \) easy parallel implementation (MPI)
- Equation of state with artificial sound speed \( \rightarrow \) CFL less restrictive
Model (2D and 3D) : low mach bi-fluid euler (isothermal non-cv)

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{u}) &= 0 \\
\frac{\partial \rho \mathbf{u}}{\partial t} + \text{div} (\rho \mathbf{u}^2 + p \mathbf{I}) &= \rho \mathbf{g} \\
\frac{\partial \rho E}{\partial t} + \text{div} ((\rho E + p) \mathbf{u}) &= 0 \\
\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi &= 0
\end{align*}
\]

where

\[
\begin{align*}
\rho (t, x) : & \text{ density} \\
u (t, x) : & \text{ velocity} \\
p (t, x) : & \text{ pressure} \\
E (\varepsilon, u) : & \text{ total energy} \\
\varepsilon : & \text{ internal specific energy} \\
\varphi : & \text{ fluid’s fraction} \\
E &= \varepsilon + \frac{u^2}{2}
\end{align*}
\]

with

\[
p = p_0 + c_0 (\rho - (\varphi \rho_w + (1 - \varphi) \rho_a))
\]

Moreover,
Application to wave breaking

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

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \text{div}(\rho u) &= 0 \\
\frac{\partial \rho u}{\partial t} + \text{div}(\rho u^2 + pI) &= \rho g \\
\frac{\partial \rho E}{\partial t} + \text{div}((\rho E + p) u) &= 0 \\
\frac{\partial \phi}{\partial t} + u \cdot \nabla \phi &= 0
\end{align*}
\]

where

\[
\begin{align*}
\rho(t, x) &: \text{ density} \\
\rho(t, x) &: \text{ velocity} \\
p(t, x) &: \text{ pressure} \\
E(\varepsilon, u) &: \text{ total energy} \\
\varepsilon &: \text{ internal specific energy} \\
\phi &: \text{ fluid’s fraction} \\
E &= \varepsilon + \frac{u^2}{2}
\end{align*}
\]

with

\[
p = p_0 + c_0 (\rho - (\phi \rho_w + (1 - \phi) \rho_a))
\]

✓ hyperbolic system

- Moreover,
Application to wave breaking

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

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \text{div}(\rho u) &= 0 \\
\frac{\partial \rho u}{\partial t} + \text{div} (\rho u^2 + pI) &= \rho g \\
\frac{\partial \rho E}{\partial t} + \text{div} ((\rho E + p) u) &= 0 \\
\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi &= 0
\end{align*}
\]

where
\[
\begin{align*}
\rho(t, x) &\quad : \text{density} \\
u(t, x) &\quad : \text{velocity} \\
p(t, x) &\quad : \text{pressure} \\
E(\varepsilon, u) &\quad : \text{total energy} \\
\varepsilon &\quad : \text{internal specific energy} \\
\varphi &\quad : \text{fluid’s fraction} \\
E &\quad = \varepsilon + \frac{u^2}{2}
\end{align*}
\]

with
\[
p = p_0 + c_0 (\rho - (\varphi \rho_w + (1 - \varphi) \rho_a))
\]

- Hyperbolic system
- Entropy available

Moreover,
Application to wave breaking

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

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \text{div}(\rho u) &= 0 \\
\frac{\partial \rho u}{\partial t} + \text{div}(\rho u^2 + pI) &= \rho g \\
\frac{\partial \rho E}{\partial t} + \text{div}((\rho E + p) u) &= 0 \\
\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi &= 0
\end{align*}
\]

where

\[
\begin{align*}
\rho(t, x) & : \text{density} \\
u(t, x) & : \text{velocity} \\
p(t, x) & : \text{pressure} \\
E(\varepsilon, u) & : \text{total energy} \\
\varepsilon & : \text{internal specific energy} \\
\varphi & : \text{fluid’s fraction} \\
E &= \varepsilon + \frac{u^2}{2}
\end{align*}
\]

with

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

- hyperbolic system
- entropy available
- automatic mesh refinement

Moreover,
Application to wave breaking

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

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \text{div}(\rho u) &= 0 \\
\frac{\partial \rho u}{\partial t} + \text{div} (\rho u^2 + pI) &= \rho g \\
\frac{\partial \rho E}{\partial t} + \text{div} ((\rho E + p) u) &= 0 \\
\frac{\partial \varphi}{\partial t} + u \cdot \nabla \varphi &= 0
\end{align*}
\]

where:
- \( \rho(t, x) \) : density
- \( u(t, x) \) : velocity
- \( p(t, x) \) : pressure
- \( E(\varepsilon, u) \) : total energy
- \( \varepsilon \) : internal specific energy
- \( \varphi \) : fluid’s fraction
- \( E = \varepsilon + \frac{u^2}{2} \)

with:
- \( p = p_0 + c_0 (\rho - (\varphi \rho_w + (1 - \varphi) \rho_a)) \)

- hyperbolic system
- entropy available
- automatic mesh refinement
- local time stepping

Moreover,
Parallelization: mesh strategy?

- hard and main task to handle
Parallelization: mesh strategy?

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

- hard and main task to handle
- strategy: domain, block, cpu?
  - domain=block=1 cpu: “failure” → synchronization depends on the finest domain
Parallelization: mesh strategy?

- hard and main task to handle
- strategy: domain, block, cpu?
  - $\text{domain} = \text{block} = 1$ cpu: “failure” → synchronization depends on the finest domain
  - $\text{domain} = n \times \text{blocks} = 1$ cpu: “good compromise” → each domain has almost the same number of cells → “better” synchronization

It certainly exists better strategy...
Parallelization: mesh 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 = 1 cpu: “good compromise” → each domain has almost the same number of cells → “better” synchronization
  3. It certainly exists better strategy . . .
Parallelization: mesh strategy?

- 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 of cells $\rightarrow$ “better” synchronization
  3. It certainly exists better strategy . . .

- Management of domain’s interfaces, projection step, . . .
DOMAIN = N X BLOCKS = 1 CPU

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
How it works?

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

![Domain Grid]

\[
\begin{array}{cccc}
6 & 9 & & \\
3 & 5 & 8 & \\
1 & 2 & 4 & 7 \\
\end{array}
\]
DOMAIN = $n \times \text{ BLOCKS} = 1\text{CPU}$

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
Domain $= n \times \text{blocks} = 1\text{cpu}$

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

**Domain**: \( n \times \text{blocks} = 1 \text{cpu} \)
How it works?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering
- more sophisticated numbering exists...
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)
**2D-3D dambreak with an obstacle**

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

(top left : mesh, top middle : $\rho$, top right : $S^n_k$, bottom left : level, bottom right : $\frac{1}{|D|} \int_D S^n_k$)
VERSUS EXPERIMENTAL (KOSHIZUKA, TAMAKO, OKA, 95)

$T = 0.2s$

$T = 0.3s$

$T = 0.4s$
Kleeßmann test case

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

- 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 . . .
**Just for fun : visualisation tool**

- A second movie (bifluid Euler equations):

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

---

M. Ersoy (IMATH)  Application to wave-breaking  Bilbao, July 17-18, 2014  34 / 38
Just for fun: visualisation tool but not only

- speed-up vs proc number

![Graph showing speed-up vs proc number for different methods.](image-url)
Just for fun : visualisation tool but not only

- cpu time vs proc number
Riemann data:

\[(p, \rho, u, v)(0, x, y) = \begin{cases} 
(p_1, \rho_1, u_1, v_1), & \text{if } x > 0.5 \text{ and } y > 0.5 \\
(p_2, \rho_2, u_2, v_2), & \text{if } x < 0.5 \text{ and } y > 0.5 \\
(p_3, \rho_3, u_3, v_3), & \text{if } x < 0.5 \text{ and } y < 0.5 \\
(p_4, \rho_4, u_4, v_4), & \text{if } x > 0.5 \text{ and } y < 0.5 
\end{cases}\]
2D Euler Riemann problem: a computational challenge (Liska, Wendroff, 01)

- Riemann data:

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

- 19 possible configuration: forward or backward 1D waves (rarefaction, shock and contact discontinuity)
2D Euler Riemann problem: a computational challenge (Liska, Wendroff, 01)

- Riemann data:

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

- Resolution of stationary contacts bordering the lower left quadrant
2D Euler Riemann problem: a computational challenge (Liska, Wendroff, 01)

- Riemann data:

\[
(p, \rho, u, v)(0, x, y) = \begin{cases} 
(1, 1, 0, -0.4), & \text{if } x > 0.5 \text{ and } y > 0.5 \\
(1, 2, 0.0, -0.3), & \text{if } x < 0.5 \text{ and } y > 0.5 \\
(0.4, 1.0625, 0, 0.2145), & \text{if } x < 0.5 \text{ and } y < 0.5 \\
(0.4, 0.5197, 0, -1.1259), & \text{if } x > 0.5 \text{ and } y < 0.5 
\end{cases}
\]

- Two standing contacts on the line \(x=0.5\)
1 Physical modeling and numerical motivation

2 2D and 3D applications

3 Concluding remarks & perspectives
Achievements and perspectives in CM2

- low mach bi-fluid model 1D, 2D and 3D
Achievements and perspectives in CM2

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

To do
- ▶ optimization of the parallel processing
- ▶ GCPU
- ▶ Application to Shallow water equations:
- . . .
Achievements and perspectives in CM2

- 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
Achievements and perspectives in CM2

- 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:
  - ...

M. Ersoy (IMATH)
Achievements and perspectives in CM2

- 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
Achievements and perspectives in CM2

- 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
Achievements and perspectives in CM2

- 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
Achievements and perspectives in CM2

- 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:
Achievements and perspectives in CM2

- 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.