Block-Based Adaptive Mesh Refinement scheme based on numerical density of entropy production for conservation laws and applications.

Mehmet Ersoy¹, Frédéric Golay, Lyudmyla Yushchenko, Université de Toulon, IMATH, and Damien Sous, Aix-Marseille Université, CNRS/INSU, IRD, MIO

International meeting AMS/EMS/SPM
Partial Differential Equations: Ambitious Mathematics for Real-life Applications

2015, 10-13 June, Porto, Portugal

¹. Mehmet.Ersoy@univ-tln.fr
Motivations

- **Physical motivations**: to be able to simulate applications in real-life fluid mechanics in dimension 2 and 3
  - wave-breaking,
  - wave-impacting,
  - tsunami . . .
Motivations

- **Physical motivations**: to be able to simulate applications in real-life fluid mechanics in dimension 2 and 3
  - wave-breaking,
  - wave-impacting,
  - tsunami . . .

- **Numerical motivations**: to be able to design a model and a numerical code for such applications
  - fast and accurate,
  - limiting the numerical diffusion,
  - adaptive and a suitable meshing machinery,
  - optimized numerical code,
  - . . .

- **Mathematical motivations**: introducing new tools
  - a suitable mesh refinement tool and its mathematical properties
  - consistency at interface of two cells of different level
  - . . .
Motivations

- **Physical motivations**: to be able to simulate applications in real-life fluid mechanics in dimension 2 and 3
  - wave-breaking,
  - wave-impacting,
  - tsunami . . .

- **Numerical motivations**: to be able to design a model and a numerical code for such applications
  - fast and accurate,
  - limiting the numerical diffusion,
  - adaptive and a suitable meshing machinery,
  - optimized numerical code,
  - . . .

- **Mathematical motivations**: introducing new tools
  - a suitable mesh refinement tool and its mathematical properties
  - consistency at interface of two cells of different level,
  - . . .
1 Principle of the method
- Generality
- 1d examples and local time stepping
- Data structure: BB-AMR

2 Applications
- The two phase low Mach model
- A two-dimensional dam-break problem
- A three-dimensional dam-break problem

3 Conclusions
1 Principle of the method
   - Generality
   - 1d examples and local time stepping
   - Data structure : BB-AMR

2 Applications
   - The two phase low Mach model
   - A two-dimensional dam-break problem
   - A three-dimensional dam-break problem

3 Conclusions
1 **Principle of the method**
   - Generality
     - 1d examples and local time stepping
     - Data structure: BB-AMR

2 **Applications**
   - The two phase low Mach model
   - A two-dimensional dam-break problem
   - A three-dimensional dam-break problem

3 **Conclusions**
Hyperbolic equations and entropy condition

We focus on general non linear hyperbolic conservation laws

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

\( w \in \mathbb{R}^d \) : vector state,
\( f \) : flux governing the physical description of the flow.
We focus on general non linear hyperbolic conservation laws

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

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:

\[
(\nabla \psi(w))^T = (\nabla s(w))^T D_w f(w)
\]
We focus on general non linear hyperbolic conservation laws

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

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:

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

Entropy inequality \(\simeq \text{“smoothness indicator”}\)

Hyperbolic equations and entropy condition

We focus on general non linear hyperbolic conservation laws

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

Weak solutions satisfy

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

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

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

Entropy inequality \(\simeq \text{"smoothness indicator"}\)

Finite volume approximation

\[ C_k, \quad |C_k| = h_k \]

\[ x_{k-1/2} \quad x_k \quad x_{k+1/2} \]

**Figure:** a cell \( C_k \)

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

\[ w_{k}^{n} \approx \frac{1}{h_k} \int_{C_k} w(t_n, x) \, dx \quad \text{and} \quad F_{k+1/2}^{n} \approx \frac{1}{\delta t} \int_{C_k} f(t, w(t, x_{k+1/2})) \, dx \]
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

\[ w_{k}^{n} \simeq \frac{1}{h_{k}} \int_{C_{k}} w(t_{n}, x) \, dx \] and \[ F_{k+1/2}^{n} \simeq \frac{1}{\delta t} \int_{C_{k}} f(t, w(t, x_{k+1/2})) \, dx \]

The numerical density of entropy production:

\[ S_{k}^{n} = \frac{s_{k}^{n+1} - s_{k}^{n}}{\delta t_{n}} + \frac{\psi_{k+1/2}^{n} - \psi_{k-1/2}^{n}}{h_{k}} \approx 0 \]
Finite volume approximation:

\[
\begin{align*}
\mathbf{w}_k^{n+1} &= \mathbf{w}_k^n - \frac{\delta t_n}{h_k} \left( \sum_a F(\mathbf{w}_k^n, \mathbf{w}_a^n; n_{k/a}) \right), \\
h_k &= \frac{|C_k|}{\sum_a |\partial C_{k/a}|}
\end{align*}
\]

with

\[
\begin{align*}
\mathbf{w}_k^n &\approx \frac{1}{h_k} \int_{C_k} \mathbf{w}(t_n, x) \, dx, \quad \text{and} \quad F(\mathbf{w}_k^n, \mathbf{w}_a^n; n_{k/a}) \approx \frac{1}{\delta t} \int_{\partial C_k} \mathbf{f}(t, \mathbf{w}) \cdot n_{k/a} \, ds
\end{align*}
\]
Finite volume approximation:

\[ w_{k}^{n+1} = w_{k}^{n} - \frac{\delta t_{n}}{h_{k}} \left( \sum_{a} F(w_{k}^{n}, w_{a}^{n}; n_{k}/a) \right), \quad h_{k} = \frac{|C_{k}|}{\sum_{a} |\partial C_{k}/a|} \]

with

\[ w_{k}^{n} \approx \frac{1}{h_{k}} \int_{C_{k}} w(t_{n}, x) \, dx, \quad \text{and} \quad F(w_{k}^{n}, w_{a}^{n}; n_{k}/a) \approx \frac{1}{\delta t} \int_{\partial C_{k}} f(t, w) \cdot n_{k}/a \, ds \]

The numerical density of entropy production:

\[ S_{k}^{n} = \frac{s_{k}^{n+1} - s_{k}^{n}}{\delta t_{n}} + \sum_{a} \frac{\psi(w_{k}^{n}, w_{a}^{n}; n_{k}/a)}{h_{k}} \approx 0 \]
Mesh refinement indicator: principle & illustration

- Given $w_k^n \rightarrow \text{compute } w_k^{n+1}$
Mesh refinement indicator: principle & illustration

- Given $w_k^n \rightarrow$ compute $w_k^{n+1}$
- Compute $S_k^n : S_k^n \neq 0 \implies$ the cell is refined or coarsened
Mesh refinement indicator: principle & illustration

- Given $w^n_k \rightarrow$ compute $w^{n+1}_k$
- 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 $\overline{S} = \frac{1}{|\Omega|} \int_{\Omega} S^n_k$
Mesh refinement indicator: principle & illustration

- Given $w_k^n \rightarrow$ compute $w_k^{n+1}$
- Compute $S_k^n : S_k^n \neq 0 \implies$ the cell is refined or coarsened
- More precisely:
  - $S_k^n \leq \alpha_{\min} \bar{S} \implies$ the cell is refined with $\bar{S} = \frac{1}{|\Omega|} \int_{\Omega} S_k^n$
  - $S_k^n \geq \alpha_{\max} \bar{S} \implies$ the cell is coarsened

M. Ersoy (IMATH)
Mesh refinement indicator: principle & illustration

- Given $w^n_k \rightarrow$ compute $w^{n+1}_k$
- 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
- Dynamic mesh refinement:
  - Dynamic mesh refinement:
    - Dyadic tree (1D)
    - Hierarchical numbering: basis 2
Mesh refinement indicator: principle & illustration

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

\[\begin{array}{ccc}
0 & 10 & 11 \\
& 120 & 121 \\
& 122 & 123 & 13 \\
2 &  & 3
\end{array}\]
Mesh refinement indicator : principle & illustration

- Given \( w_k^n \rightarrow \text{compute} \ w_k^{n+1} \)
- Compute \( S_k^n : S_k^n \neq 0 \rightarrow \text{the cell is refined or coarsened} \)
- More precisely :
  - \( S_k^n \leq \alpha_{\min} \bar{S} \rightarrow \text{the cell is refined with} \ \bar{S} = \frac{1}{|\Omega|} \int_{\Omega} S_k^n \)
  - \( S_k^n \geq \alpha_{\max} \bar{S} \rightarrow \text{the cell is coarsened} \)
  - 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

- Given $w^n_k \rightarrow$ compute $w^{n+1}_k$
- 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

- Given $w^n_k \rightarrow$ compute $w^{n+1}_k$
- 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

\[
\begin{align*}
F_{k_b-1/2} &= F_{k_b+1/2} \\
w^n_{k_b} &= w^n_{k_b} \\
w^n_{k_b-1/2} &= F_{k_b-1/2} = F_{k_b+1/2} \\
w^n_{k_b} &= w^n_{k_b} + w^n_{k_b} \\
F_{k_b+1/2} &= F_{k_b+1/2}
\end{align*}
\]
**Mesh refinement indicator: principle & illustration**

- Given $w^n_k \rightarrow$ compute $w^{n+1}_k$
- Compute $S^n_k : S^n_k \neq 0 \implies$ the cell is refined or coarsened
- More precisely:
  - $S^n_k \leq \alpha_{\min} \bar{S} \implies$ the cell is refined with $\bar{S} = \frac{1}{|\Omega|} \int_{\Omega} S^n_k$
  - $S^n_k \geq \alpha_{\max} \bar{S} \implies$ the cell is coarsened
  - Simple approach but the scheme is locally non consistent \([SO88, TW05]\)

---


Mesh refinement indicator: principle & illustration

- Given $w_k^n \rightarrow$ compute $w_k^{n+1}$
- Compute $S_k^n : S_k^n \neq 0 \implies$ the cell is refined or coarsened
- More precisely:
  - $S_k^n \leq \alpha \min S \implies$ the cell is refined with $S = \frac{1}{|\Omega|} \int_{\Omega} S_k^n$
  - $S_k^n \geq \alpha \max S \implies$ the cell is coarsened
- Simple approach but the scheme is locally non consistent [SO88, TW05]
- Limit the mesh level of adjacent cells by 2

---


Mesh refinement indicator: principle & illustration

- Given $w^n_k \rightarrow$ compute $w^{n+1}_k$
- 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
  - Simple approach but the scheme is locally non-consistent [SO88, TW05]
  - Limit the mesh level of adjacent cells by 2
  - A correction can be obtained (work in progress) [AE15]


1 Principle of the method
   - Generality
   - 1d examples and local time stepping
   - Data structure: BB-AMR

2 Applications
   - The two phase low Mach model
   - A two-dimensional dam-break problem
   - A three-dimensional dam-break problem

3 Conclusions
**An example: the 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

- \(\rho(t, x)\): density
- \(u(t, x)\): velocity
- \(p(t, x)\): pressure
- \(\gamma := 1.4\): ratio of the specific heats
- \(E(\varepsilon, u)\): total energy
- \(\varepsilon\): internal specific energy
- \(E = \varepsilon + \frac{u^2}{2}\)
An example: the one-dimensional gas dynamics equations for ideal gas

\[ \begin{align*}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} &= 0 \\
\frac{\partial \rho u}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} &= 0 \\
\frac{\partial \rho E}{\partial t} + \frac{\partial (\rho E + p) u}{\partial x} &= 0
\end{align*} \]

where

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

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

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

\[ \varepsilon \]: internal specific energy

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

- **Conservative variables**

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

- **entropy**

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

Mesh refinement parameter $\alpha_{\text{max}}$ : 0.01

Mesh coarsening parameter $\alpha_{\text{min}}$ : 0.001

Mesh refinement parameter $\bar{S}$ : \( \frac{1}{|\Omega|} \sum_{kb} S_{kb} \)

CFL : 0.25

Simulation time (s) : 0.4

Initial number of cells : 200

Maximum level of mesh refinement : $L_{\text{max}}$
**Accuracy**

(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.
Explicit adaptive schemes: time consuming due to the restriction

\[ \|w\| \frac{\delta t}{h} \leq 1, \quad h = \min_k h_k \]
**Time restriction, local time stepping approach**

- Explicit adaptive schemes: time consuming due to the restriction

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

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

---

Explicit adaptive schemes: time consuming due to the restriction

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

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

Explicit adaptive schemes: time consuming due to the restriction

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

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


1 Principle of the method
   - Generality
   - 1d examples and local time stepping
   - Data structure : BB-AMR

2 Applications
   - The two phase low Mach model
   - A two-dimensional dam-break problem
   - A three-dimensional dam-break problem

3 Conclusions
Main difficulty: mesh and data structure. For fast computation, the following are required:

- parallel treatment
- hierarchical grids
Main difficulty: mesh and data structure. Some interesting issues:

- 2D quad-tree [ZW11],
- 3D octree [LGF04],
- 2D/3D anisotropic AMR [HFCC13].


Main difficulty: mesh and data structure.

The strategy adopted:
Main difficulty: mesh and data structure.

The strategy adopted:

1. **1 fixed domain= 1 fixed block=1 cpu**: “failure” → synchronization depends on the finest domain
2. **Dynamic domain=n × static blocks = 1 cpu**: “good compromise” → each domain has almost the same number of cells → “better” synchronization

It certainly exists better strategy...
Main difficulty: mesh and data structure.

The strategy adopted:

1. 1 fixed domain = 1 fixed block = 1 cpu: “failure” → synchronization depends on the finest domain

2. 1 dynamic domain = n × static blocks = 1 cpu: “good compromise” → each domain has almost the same number of cells → “better” synchronization = Block-Based Adaptive Mesh Refinement (BB-AMR)
Main difficulty: mesh and data structure.

The strategy adopted:

1. 1 fixed domain = 1 fixed block = 1 cpu: "failure" → synchronization depends on the finest domain
2. 1 dynamic domain = n × static blocks = 1cpu: "good compromise" → each domain has almost the same number of cells → "better" synchronization = Block-Based Adaptive Mesh Refinement (BB-AMR)
3. It certainly exists better strategy...
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

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
How it works?

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

![Grid with numbered cells](image)
How it works?

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

![Diagram](image)
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...
How it works?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering
- more sophisticated numbering exists . . .
- re-numbering and re-meshing being expensive
  - the mesh should be kept constant on a time interval
How it works?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering
- more sophisticated numbering exists . . .
- re-numbering and re-meshing being expensive
  - the mesh should be kept constant on a time interval
  - AMR time-step computed through the smallest block and not the smallest cell

\[ T_{n+1} - T_n = \Delta T_{\text{AMR}} \]

\[ \Delta T_{\text{AMR}} \leq \beta \frac{\min_k h_{\text{block}k}}{\max_k \| u_{\text{block}k} \|}, \quad 0 < \beta \leq 1. \]

\[
\begin{array}{c|c|c}
T_0 & \delta t & T_1 \\
\hline
\text{AMR} & \text{AMR} & \\
\end{array}
\]
How it works?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering
- more sophisticated numbering exists . . .
- re-numbering and re-meshing being expensive
  - the mesh should be kept constant on a time interval
  - AMR time-step computed through the smallest block and not the smallest cell
  - Gain is important and numerical stability is conserved!

Examples:

- A two-dimensional example of BB-AMR with 3 domains and 9 blocks.
Examples:

- A two dimensional example of BB-AMR with 3 domains and 9 blocks.

- A three dimensional example of BB-AMR with 3 domains and 27 blocks.
1. **PRINCIPLE OF THE METHOD**
   - Generality
   - 1d examples and local time stepping
   - Data structure : BB-AMR

2. **APPLICATIONS**
   - The two phase low Mach model
   - A two-dimensional dam-break problem
   - A three-dimensional dam-break problem

3. **CONCLUSIONS**
Understanding of wave hydrodynamics is of primary interest for ocean and naval engineering applications:

- dynamics of ships and floating structures,
- stability of offshore structures,
- coastal erosion and submersion processes, . . . .
Understanding of wave hydrodynamics is of primary interest for ocean and naval engineering applications:

It’s difficult to describe accurately wave dynamics and still a fairly open research field.

breaking or impacting waves on rigid structures = violent transformations
Understanding of wave hydrodynamics is of primary interest for ocean and naval engineering applications:

- It’s difficult to describe accurately wave dynamics and still a fairly open research field.
- Involved physical processes, such as splash-ups or gas pockets entrapment, are quite complex and can hardly be characterized by field or laboratory experiments or analytical approaches: several models!
Understanding of wave hydrodynamics is of primary interest for ocean and naval engineering applications:

It’s difficult to describe accurately wave dynamics and still a fairly open research field.

Involved physical processes, such as splash-ups or gas pockets entrapment, are quite complex and can hardly be characterized by field or laboratory experiments or analytical approaches: several models!

Therefore, numerical simulation of breaking and impacting waves is both
  ▶ an attractive research topic
  ▶ a challenging task for coastal and environmental engineering
1 Principle of the method
- Generality
- 1d examples and local time stepping
- Data structure: BB-AMR

2 Applications
- The two phase low Mach model
- A two-dimensional dam-break problem
- A three-dimensional dam-break problem

3 Conclusions
The governing equations

- Assumptions: physics of impacting/breaking waves can be simplified
  - mainly governed by pressure forces and overturning forces
  - Mach number $< 0.3 \rightarrow$ fluid is slightly compressible
The governing equations

Assumptions: physics of impacting/breaking waves can be simplified

- mainly governed by pressure forces and overturning forces
- Mach number $< 0.3 \rightarrow$ fluid is slightly compressible
- small-scale friction and dissipation process are neglected

Moreover, hyperbolic system, entropy available, automatic mesh refinement, local time stepping
The governing equations

- Assumptions: physics of impacting/breaking waves can be simplified
  - mainly governed by pressure forces and overturning forces
  - Mach number \( < 0.3 \rightarrow \) fluid is slightly compressible
  - small-scale friction and dissipation process are neglected
  - two-phase flow Compressible Euler equations can be considered

- Model (2D and 3D): low mach two phase

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

where

- \( \rho(t, x) \) : density
- \( u(t, x) \) : velocity
- \( p(t, x) \) : pressure
- \( \varphi \) : fluid’s fraction
The governing equations

- **Assumptions**: physics of impacting/breaking waves can be simplified
  - mainly governed by pressure forces and overturning forces
  - Mach number \(< 0.3 \rightarrow\) fluid is slightly compressible
  - small-scale friction and dissipation process are neglected
  - two-phase flow Compressible Euler equations can be considered
  - An artificial linearized pressure law is used to compute low Mach flows [C67]

- **Model (2D and 3D)**: low mach two phase

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

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

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

where
- \(\rho(t, x)\) : density
- \(u(t, x)\) : velocity
- \(p(t, x)\) : pressure
- \(\varphi\) : fluid’s fraction

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

The governing equations

- Assumptions
- Model (2D and 3D) : low mach two phase

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

where \( \rho(t, x) \) : density, \( u(t, x) \) : velocity, \( p(t, x) \) : pressure, \( \phi \) : fluid’s fraction

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

- Equation of state with artificial sound speed \( \rightarrow \) CFL less restrictive
The governing equations

- Assumptions
- Model (2D and 3D) : low mach two phase

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

where
- \(\rho(t,x)\) : density
- \(u(t,x)\) : velocity
- \(p(t,x)\) : pressure
- \(\varphi\) : fluid’s fraction

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

Equation of state with artificial sound speed \(\rightarrow\) CFL less restrictive

Explicit scheme \(\rightarrow\) easy parallel implementation (MPI)
The governing equations

- Assumptions
- Model (2D and 3D) : low mach two phase

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

where
- \( \rho(t, x) \) : density
- \( u(t, x) \) : velocity
- \( p(t, x) \) : pressure
- \( \varphi \) : fluid’s fraction

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

- Equation of state with artificial sound speed \( \rightarrow \) CFL less restrictive
- Explicit scheme \( \rightarrow \) easy parallel implementation (MPI)
  - hyperbolic system

Moreover,
The governing equations

- Assumptions
- Model (2D and 3D) : low mach two phase

$$\rho(t, x) : \text{density}$$
$$u(t, x) : \text{velocity}$$
$$p(t, x) : \text{pressure}$$
$$\varphi : \text{fluid’s fraction}$$

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

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

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

with $$p = p_0 + c_0 (\rho - (\varphi \rho_w + (1 - \varphi) \rho_\alpha))$$

- Equation of state with artificial sound speed $\rightarrow$ CFL less restrictive
- Explicit scheme $\rightarrow$ easy parallel implementation (MPI)
  - hyperbolic system
  - entropy available
- Moreover,
The governing equations

- Assumptions
- Model (2D and 3D): low mach two phase

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

where
- \( \rho(t, x) \): density
- \( u(t, x) \): velocity
- \( p(t, x) \): pressure
- \( \varphi \): fluid’s fraction

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

- Equation of state with artificial sound speed → CFL less restrictive
- Explicit scheme → easy parallel implementation (MPI)
- Moreover, hyperbolic system
- entropy available
- automatic mesh refinement
The governing equations

- Assumptions
- Model (2D and 3D): low mach two phase

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

where
- \(\rho(t, x)\): density
- \(u(t, x)\): velocity
- \(p(t, x)\): pressure
- \(\varphi\): fluid’s fraction

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

- Equation of state with artificial sound speed \(\rightarrow\) CFL less restrictive
- Explicit scheme \(\rightarrow\) easy parallel implementation (MPI)
- Moreover,
  - hyperbolic system
  - entropy available
  - automatic mesh refinement
  - local time stepping
1 Principle of the method
- Generality
- 1d examples and local time stepping
- Data structure: BB-AMR

2 Applications
- The two phase low Mach model
- A two-dimensional dam-break problem
- A three-dimensional dam-break problem

3 Conclusions
A two-dimensional dam-break problem [KTO95]

- capture the complex structure of the air-water interface after wave impact

A two-dimensional dam-break problem

- capture the complex structure of the air-water interface after wave impact
- Experimental configuration
A two-dimensional dam-break problem

- capture the complex structure of the air-water interface after wave impact
- **Numerical parameters**:

  - Mesh refinement parameter $\alpha_{\text{max}}$: 0.2
  - Mesh coarsening parameter $\alpha_{\text{min}}$: 0.02
  - Number of domain: 321
  - Number of blocks: 321
  - Number of processors: 120
  - Maximum level of mesh refinement: $L_{\text{max}} = 5$
  - CFL: $CFL = 0.8$
  - Simulation time: $T = 1.5$
  - AMR time: $AMR = 300$
A two-dimensional dam-break problem

- capture the complex structure of the air-water interface after wave impact
- Confrontation with experiments: $T = 0$

**Figure:** mesh (left), density with blue and red corresponding to air and water, respectively (center), mesh refinement level (1 to 5) per block (right)
A two-dimensional dam-break problem

capture the complex structure of the air-water interface after wave impact

Confrontation with experiments: $T = 0.2$

Figure: (a) Mesh; (b) Density (air-blue, water-red); (c) Density of numerical entropy production (green-zero, blue-negative values); (d) Mesh refinement level per block (1 to 5); (e) Experiment; (f) Mesh refinement criterion per block.
A two-dimensional dam-break problem

- capture the complex structure of the air-water interface after wave impact
- Confrontation with experiments: $T = 0.4$

**Figure:** (a) Mesh; (b) Density (air-blue, water-red); (c) Density of numerical entropy production (green-zero, blue-negative values); (d) Mesh refinement level per block (1 to 5); (e) Experiment; (f) Mesh refinement criterion per block.
A two-dimensional dam-break problem

capture the complex structure of the air-water interface after wave impact

Remarks:
- number of cells varies from 70,000 and 100,000
- elapsed computing time about 5 hours
- 1 domain = 1 block → better results with BB-AMR.
1 Principle of the method
   - Generality
   - 1d examples and local time stepping
   - Data structure: BB-AMR

2 Applications
   - The two phase low Mach model
   - A two-dimensional dam-break problem
   - A three-dimensional dam-break problem

3 Conclusions
A three-dimensional dam-break problem [K05]

- capture the complex structure of the air-water interface after wave impact

A three-dimensional dam-break problem

- capture the complex structure of the air-water interface after wave impact
- Experimental configuration

**Figure:** domain geometry and sensors points from http://www.math.rug.nl/~veldman/comflow/dambreak.html
A three-dimensional dam-break problem

capture the complex structure of the air-water interface after wave impact

Numerical parameters:

- Mesh refinement parameter $\alpha_{\text{max}}$ : 0.2
- Mesh coarsening parameter $\alpha_{\text{min}}$ : 0.02
- Number of domain : 48
- Number of blocks : 3628
- Number of processors : 48
- Maximum level of mesh refinement $L_{\text{max}}$ : 4
- CFL : $CFL = 0.8$
- Simulation time : $T = 4.8$
- AMR time : $AMR = 240$
A three-dimensional dam-break problem

- capture the complex structure of the air-water interface after wave impact
- Confrontation with experiments:

**Figure:** Free surface computed by Kleefsman (left), the experimentation (center) and our (right) at $t = 0.4, 0.6, 1, 1.8, 2, 4.8$s
A three-dimensional dam-break problem

- capture the complex structure of the air-water interface after wave impact
- Confrontation with experiments:


A three-dimensional dam-break problem

capture the complex structure of the air-water interface after wave impact

Confrontation with experiments:

Figure: Domains due to the BB-AMR scheme (left) and air-water interface (right) at time 0.4s, 0.6s, 1.0s, 2s.
A three-dimensional dam-break problem

- capture the complex structure of the air-water interface after wave impact

**Remarks:**
- number of cells varies from 800,000 cells up to about 1,500,000 cells
- elapsed computing time about 10 hours (instead of 24h [GH07])


A three-dimensional dam-break problem [AEGDSL15]

- A “block” dam break problem with a confrontation of RK2 and AB2

A three-dimensional dam-break problem

- A “block” dam break problem with a confrontation of RK2 and AB2
- Initial configuration

**Figure:** Unit cube \( \left( \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \right) \)
A three-dimensional dam-break problem

- A “block” dam break problem with a confrontation of RK2 and AB2
- Numerical parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh refinement parameter $\alpha_{\text{max}}$</td>
<td>0.2</td>
</tr>
<tr>
<td>Mesh coarsening parameter $\alpha_{\text{min}}$</td>
<td>0.02</td>
</tr>
<tr>
<td>Number of domain</td>
<td>1, 2, 4, 8, 32,</td>
</tr>
<tr>
<td>Number of blocks</td>
<td>3375</td>
</tr>
<tr>
<td>Number of processors</td>
<td>40</td>
</tr>
<tr>
<td>Maximum level of mesh refinement</td>
<td>$L_{\text{max}} = 4$</td>
</tr>
<tr>
<td>Simulation time</td>
<td>$T = 2.5$</td>
</tr>
<tr>
<td>AMR time</td>
<td>$AMR = 100$</td>
</tr>
</tbody>
</table>
A three-dimensional dam-break problem

- A “block” dam break problem with a confrontation of RK2 and AB2
- Confrontation with experiments:

Figure: AB2 vs RK2

(a) Speed up vs proc number

(b) cpu time vs proc number
A three-dimensional dam-break problem

- A “block” dam break problem with a confrontation of RK2 and AB2
- **Remarks:**
  - number of cells varies from 172215 cells up to about 587763 cells
  - The efficiency, i.e. \( \frac{\text{speed up}}{\text{number of processors}} \), of the computation is roughly 85% for 8 domains and 60% for 32 domains.
  - performance decrease after 20 processors \( \Rightarrow \) optimization is required to get more efficiency.
1 PRINCIPLE OF THE METHOD
   - Generality
   - 1d examples and local time stepping
   - Data structure : BB-AMR

2 APPLICATIONS
   - The two phase low Mach model
   - A two-dimensional dam-break problem
   - A three-dimensional dam-break problem

3 CONCLUSIONS
Conclusions

Several numerical validation on Euler equations
Conclusions

- Several numerical validation on Euler equations
- Several numerical validation (in progress) for shallow water equations

**Figure:** (left) $L$ and (right) Kleefsman test case (B. Cleirec)
Conclusions & Perspectives

- Several numerical validation on Euler equations
- Several numerical validation (in progress) for shallow water equations
- Local consistency error between two adjacent cells of different levels
Conclusions & Perspectives

- Several numerical validation on Euler equations
- Several numerical validation (in progress) for shallow water equations
- Local consistency error between two adjacent cells of different levels
- Capture accurately rarefactions and contact discontinuities
Conclusions & Perspectives

- Several numerical validation on Euler equations
- Several numerical validation (in progress) for shallow water equations
- Local consistency error between two adjacent cells of different levels
- Capture accurately rarefactions and contact discontinuities
- Develop a 'returning' wave model (as an intermediate one between the two-phase flow model and the shallow water equations)
Thank you for your attention.