Adaptive mesh refinement method : numerical density of entropy production and automatic thresholding.

Ersoy, M.[1 a] Pons, K.[1,2 b]

[1] EA 2134, IMATH, Université de Toulon

[2] Principia S.A.S.,13705 La Ciotat cedex

2017, 17 February, LAMA – Chambéry

a. *ersoy@univ-tln.fr*
b. *pons@univ-tln.fr*

- numerical simulations of real-life large scale fluid flows in dimension $d = 1, 2, 3$

(a) Monster waves (Nazare)

(b) Wave-breaking (Nazare)

(c) Tsunami (Japan)

(d) tsunami (Brazil)

Motivations

- numerical simulations of real-life large scale fluid flows in dimension $d = 1, 2, 3$
- Simulating fluids in large-scale $\rightarrow$ high memory and computer requirements.
  - how to compute fast and accurate : $\rightarrow$ meshing or moving the mesh only in "desired" regions with a suitable mesh refinement criterion

Motivations

- numerical simulations of real-life large scale fluid flows in dimension $d = 1, 2, 3$
- Simulating fluids in large-scale $\rightarrow$ high memory and computer requirements.
- Mathematical motivations : introducing new tool for numerical purpose !

Outline of the talk

Outline

Outline

We focus on general non linear hyperbolic conservation laws

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

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

We focus on general non linear hyperbolic conservation laws

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

Weak solutions satisfy

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

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

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

Weak solutions satisfy

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

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

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

Entropy inequality $\simeq$ "smoothness indicator"

Croisille J.-P., *Contribution à l'Étude Théorique et à l'Approximation par Éléments Finis du Système Hyperbolique de la Dynamique des Gaz Multidimensionnelle et Multiespèces*, PhD thesis, Université de Paris VI, 1991

We focus on general non linear hyperbolic conservation laws

$$\begin{cases} \dfrac{\partial \boldsymbol{w}}{\partial t} + \mathrm{div}(\boldsymbol{f}(\boldsymbol{w})) = 0,\ (t,x) \in \mathbb{R}^+ \times \mathbb{R}^d \\ \boldsymbol{w}(0,x) = \boldsymbol{w}_0(x),\ x \in \mathbb{R}^d \end{cases}$$

Weak solutions satisfy

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

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"

Croisille J.-P., *Contribution à l'Étude Théorique et à l'Approximation par Éléments Finis du Système Hyperbolique de la Dynamique des Gaz Multidimensionnelle et Multiespèces*, PhD thesis, Université de Paris VI, 1991

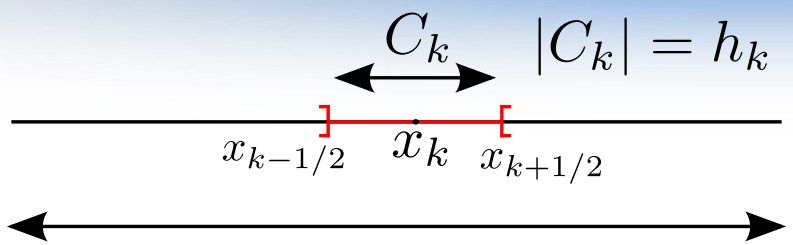


FIGURE: a cell C_k

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

$$\boldsymbol{w}_k^n \simeq \frac{1}{h_k}\int_{C_k} \boldsymbol{w}\left(t_n, x\right)\, dx \text{ and } \boldsymbol{F}_{k+1/2}^n \approx \frac{1}{\delta t}\int_{C_k} \boldsymbol{f}(t, w(t, x_{k+1/2}))\, dx$$

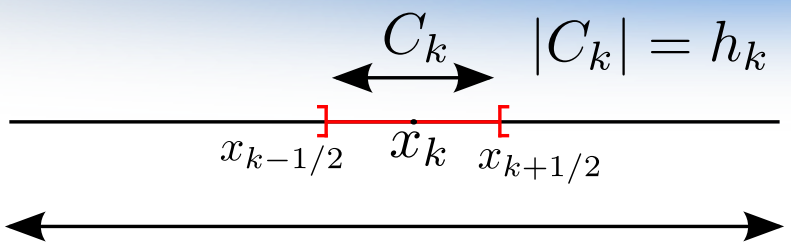


Figure: a cell C_k

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

$$\boldsymbol{w}_k^n \simeq \frac{1}{h_k}\int_{C_k} \boldsymbol{w}\left(t_n, x\right)\, dx \text{ and } \boldsymbol{F}_{k+1/2}^n \approx \frac{1}{\delta t}\int_{C_k} \boldsymbol{f}(t, w(t, x_{k+1/2}))\, dx$$

The numerical density of entropy production :

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

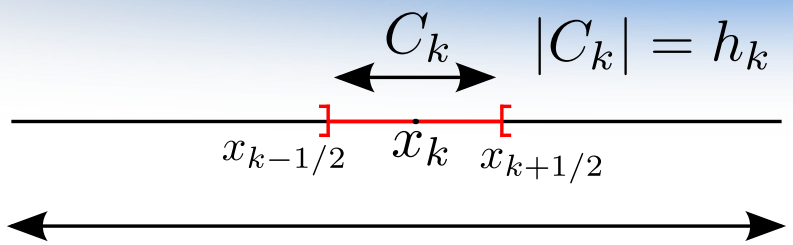


Figure: a cell C_k

Finite volume approximation :

$$\boldsymbol{w}_k^{n+1} = \boldsymbol{w}_k^n - \frac{\delta t_n}{h_k} \left(\sum_a \boldsymbol{F}(\boldsymbol{w}_k^n, \boldsymbol{w}_a^n; n_{k/a}) \right), \quad h_k = \frac{|C_k|}{\sum_a |\partial C_{k/a}|}$$

with

$$\boldsymbol{w}_k^n \simeq \frac{1}{h_k} \int_{C_k} \boldsymbol{w}\,(t_n, x)\; dx, \qquad \text{and } \boldsymbol{F}(\boldsymbol{w}_k^n, \boldsymbol{w}_a^n; n_{k/a}) \approx \frac{1}{\delta t} \int_{\partial C_k} \boldsymbol{f}(t, w) \cdot n_{k/a}\; ds$$

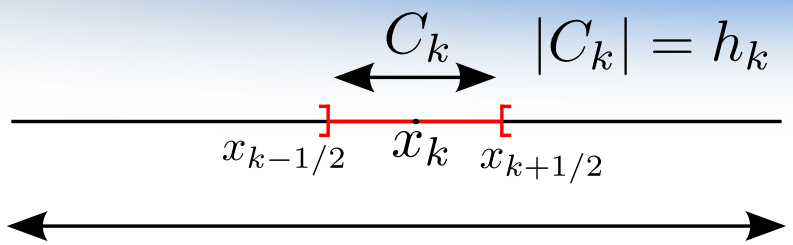


FIGURE: a cell C_k

Finite volume approximation :

$$\boldsymbol{w}_k^{n+1} = \boldsymbol{w}_k^n - \frac{\delta t_n}{h_k} \left(\sum_a \boldsymbol{F}(\boldsymbol{w}_k^n, \boldsymbol{w}_a^n; n_{k/a}) \right), \quad h_k = \frac{|C_k|}{\sum_a |\partial C_{k/a}|}$$

with

$$\boldsymbol{w}_k^n \simeq \frac{1}{h_k} \int_{C_k} \boldsymbol{w}\,(t_n, x)\; dx, \qquad \text{and } \boldsymbol{F}(\boldsymbol{w}_k^n, \boldsymbol{w}_a^n; n_{k/a}) \approx \frac{1}{\delta t} \int_{\partial C_k} \boldsymbol{f}(t, w) \cdot n_{k/a}\; ds$$

The numerical density of entropy production :

$$S_k^n = \frac{s_k^{n+1} - s_k^n}{\delta t_n} + \frac{\sum_a \psi(\boldsymbol{w}_k^n, \boldsymbol{w}_a^n; n_{k/a})}{h_k} \lessapprox 0$$

Mesh refinement indicator : principle & illustration

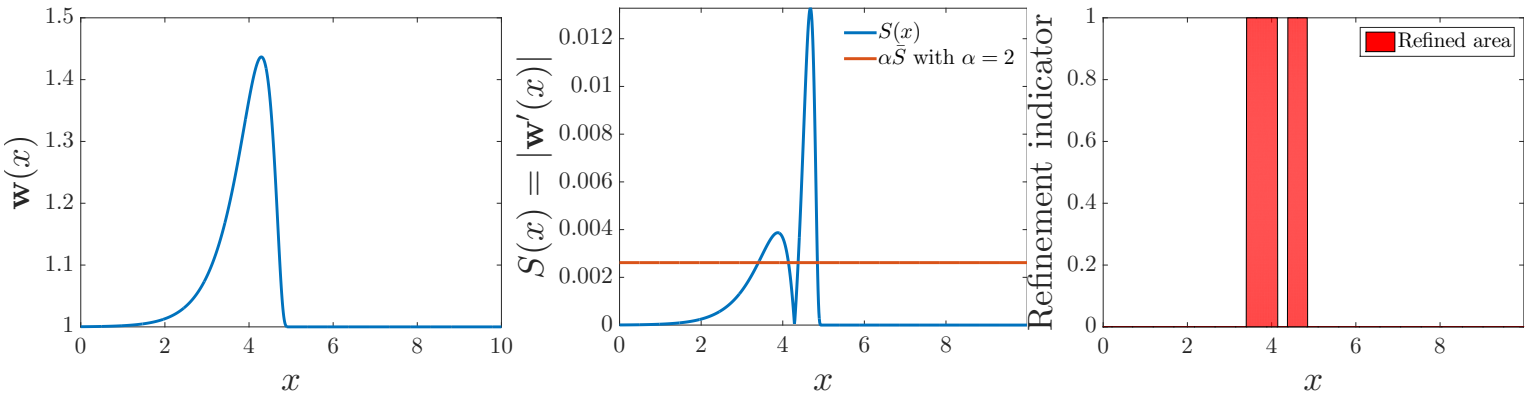
- Compute w_k^n

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

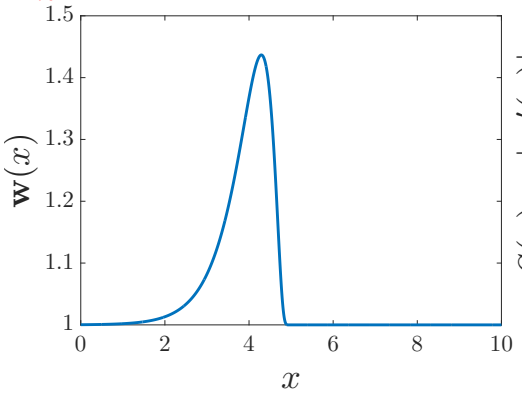
Mesh refinement indicator : principle & illustration

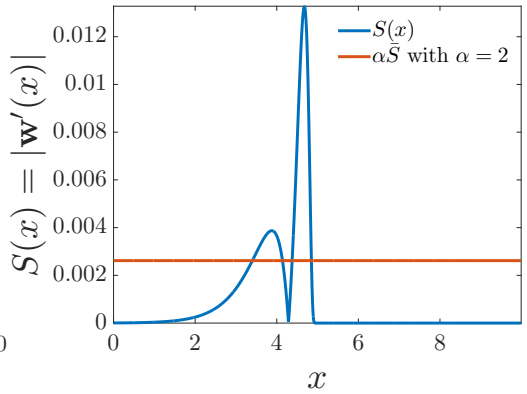
- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$

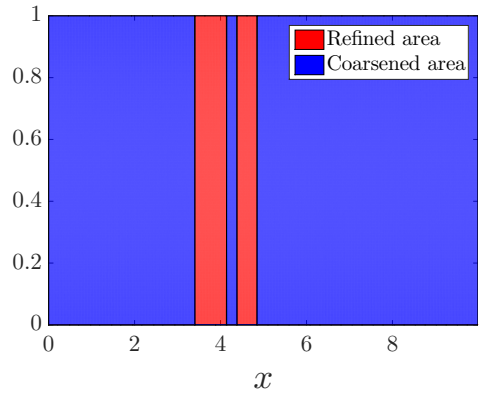


Mesh refinement indicator : principle & illustration

- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened

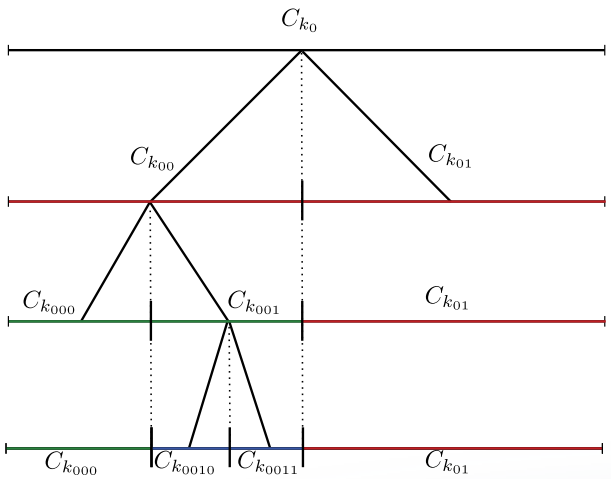






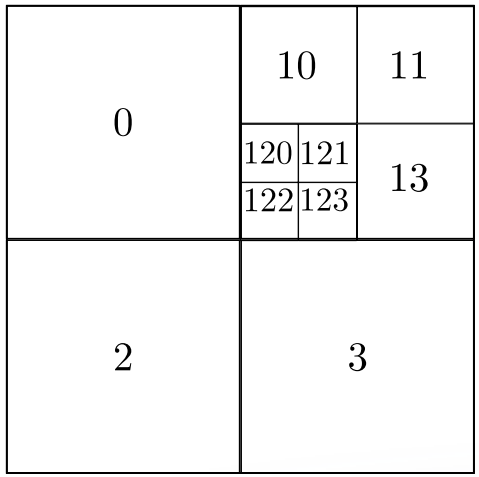
Mesh refinement indicator : principle & illustration

- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened
  - Dynamic mesh refinement :
    - ★ Dyadic tree (1D)
    - ★ hierarchical numbering : basis 2



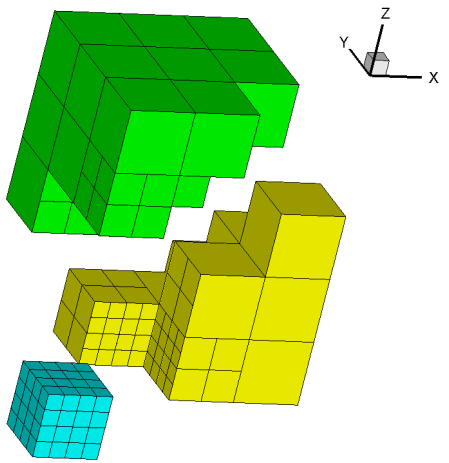
Mesh refinement indicator : principle & illustration

- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened
  - Dynamic mesh refinement :
    - ★ Non-structured grid : macro-cell
    - ★ Dyadic tree (1D), Quadtree (2D)
    - ★ hierarchical numbering : basis 2,4



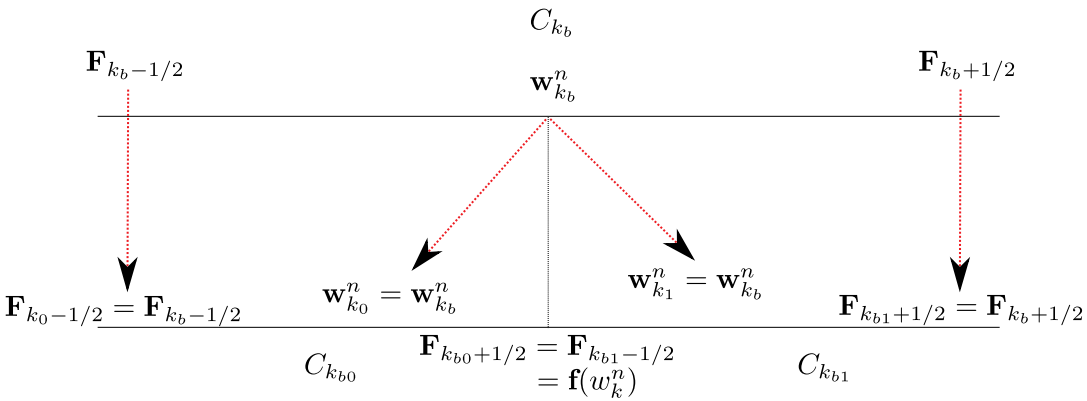
Mesh refinement indicator : principle & illustration

- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ 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



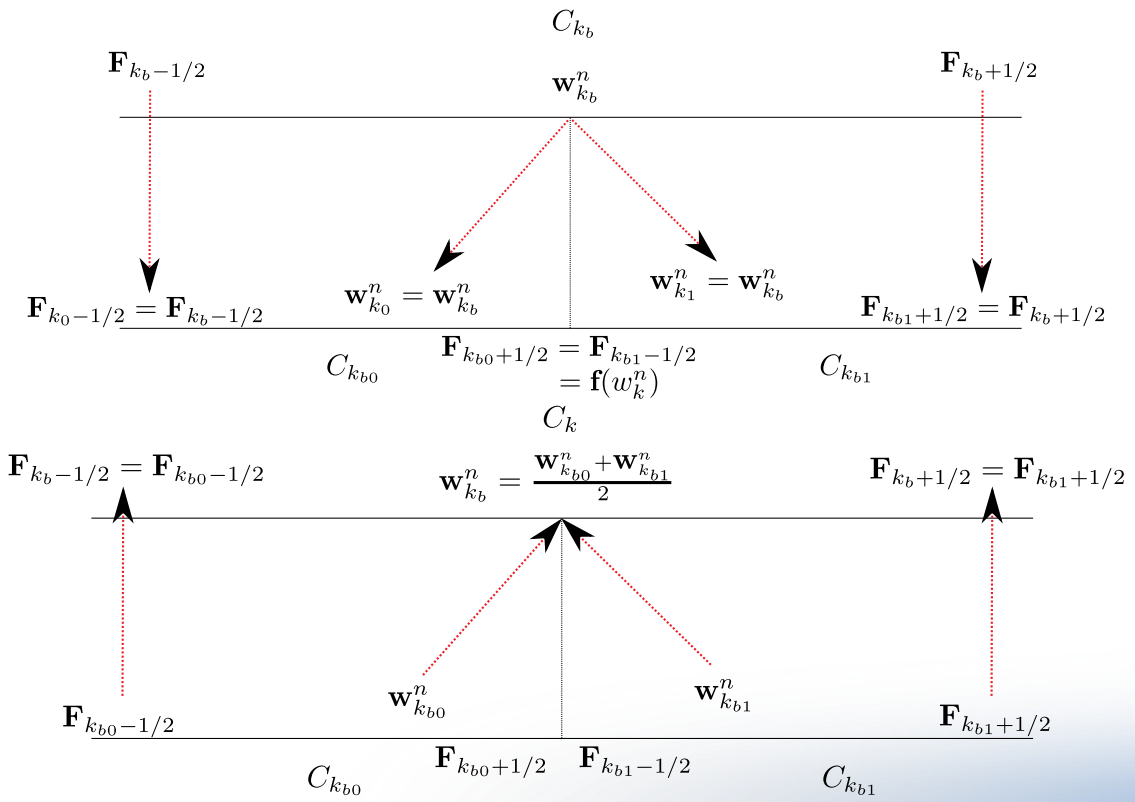
Mesh refinement indicator : principle & illustration

- Compute w_k^n
- Compute S_k^n : $S_k^n \neq 0 \Longrightarrow$ the cell is refined or coarsened
- More precisely : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened



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 : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened



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 : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened
  - A simple projection method but the scheme is locally non consistent [S088, TW05]

Shu C. W., Osher S., *Efficient implementation of essentially nonoscillatory shock-capturing schemes.* J. Comput. Phys., 77(2) :439–471, 1988.

Tang H., Warnecke G., *A class of high resolution difference schemes for nonlinear Hamilton-Jacobi equations with varying time and space grids.* SIAM J. Sci. Comput., 26(4) :1415–1431, 2005.

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 : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened
  - A simple projection method but the scheme is locally non consistent [S088, TW05]
    - ⋆ less time time consuming than other methods

Shu C. W., Osher S., *Efficient implementation of essentially nonoscillatory shock-capturing schemes.* J. Comput. Phys., 77(2) :439–471, 1988.

Tang H., Warnecke G., *A class of high resolution difference schemes for nonlinear Hamilton-Jacobi equations with varying time and space grids.* SIAM J. Sci. Comput., 26(4) :1415–1431, 2005.

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 : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened
  - A simple projection method but the scheme is locally non consistent [S088, TW05]
    - ★ less time time consuming than other methods
    - ★ non consistency is almost negligible if the level of adjacent cells is limited by 2


Shu C. W., Osher S., *Efficient implementation of essentially nonoscillatory shock-capturing schemes.* J. Comput. Phys., 77(2) :439–471, 1988.


Tang H., Warnecke G., *A class of high resolution difference schemes for nonlinear Hamilton-Jacobi equations with varying time and space grids.* SIAM J. Sci. Comput., 26(4) :1415–1431, 2005.


M. Ersoy, F. Golay, L. Yushchenko. *Adaptive multi-scale scheme based on numerical entropy production for conservation laws.* CEJM, Central European Journal of Mathematics, 11(8), pp 1392–1415, 2013.


F. Golay, M. Ersoy, L. Yuschenko, D. Sous. Block-based adaptive mesh refinement scheme using numerical density of entropy production for three-dimensional two-fluid flows. International Journal of Computational Fluid Dynamics, Taylor & Francis, 2015.

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 : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened
  - A simple projection method but the scheme is locally non consistent [S088, TW05]
    - ★ less time time consuming than other methods
    - ★ non consistency is almost negligible if the level of adjacent cells is limited by 2
  - Improvement (cpu-time) : local time stepping [EGY13] ▸ See more details


Shu C. W., Osher S., *Efficient implementation of essentially nonoscillatory shock-capturing schemes.* J. Comput. Phys., 77(2) :439–471, 1988.


Tang H., Warnecke G., *A class of high resolution difference schemes for nonlinear Hamilton-Jacobi equations with varying time and space grids.* SIAM J. Sci. Comput., 26(4) :1415–1431, 2005.


M. Ersoy, F. Golay, L. Yushchenko. *Adaptive multi-scale scheme based on numerical entropy production for conservation laws.* CEJM, Central European Journal of Mathematics, 11(8), pp 1392–1415, 2013.


F. Golay, M. Ersoy, L. Yuschenko, D. Sous. Block-based adaptive mesh refinement scheme using numerical density of entropy production for three-dimensional two-fluid flows. International Journal of Computational Fluid Dynamics, Taylor & Francis, 2015.

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 : for a given mesh refinement threshold α
  - $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with, for instance, $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
  - $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened
  - A simple projection method but the scheme is locally non consistent [S088, TW05]
    - ★ less time time consuming than other methods
    - ★ non consistency is almost negligible if the level of adjacent cells is limited by 2
  - Improvement (cpu-time) : local time stepping [EGY13] ▸ See more details
  - 2D-3D computer management : BB-AMR [GEYS] ▸ See more details

Shu C. W., Osher S., *Efficient implementation of essentially nonoscillatory shock-capturing schemes.* J. Comput. Phys., 77(2) :439–471, 1988.

Tang H., Warnecke G., *A class of high resolution difference schemes for nonlinear Hamilton-Jacobi equations with varying time and space grids.* SIAM J. Sci. Comput., 26(4) :1415–1431, 2005.

M. Ersoy, F. Golay, L. Yushchenko. *Adaptive multi-scale scheme based on numerical entropy production for conservation laws.* CEJM, Central European Journal of Mathematics, 11(8), pp 1392–1415, 2013.

F. Golay, M. Ersoy, L. Yuschenko, D. Sous. Block-based adaptive mesh refinement scheme using numerical density of entropy production for three-dimensional two-fluid flows. International Journal of Computational Fluid Dynamics, Taylor & Francis, 2015.

Outline

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

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

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

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

Sod's shock tube problem

Mesh coarsening parameter α : 0.001 ,

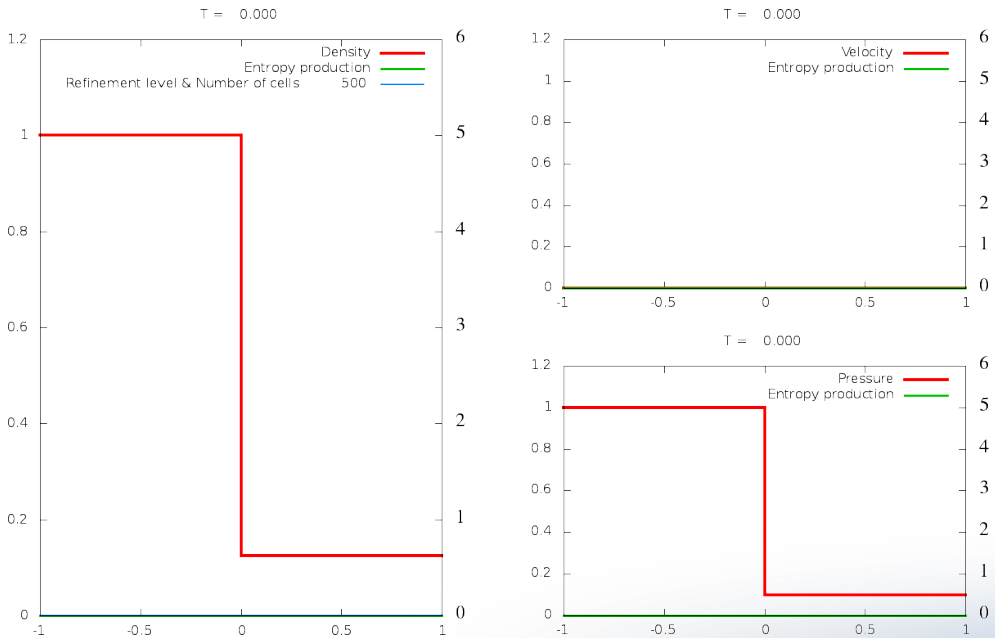
Mesh refinement parameter $\bar{S}$: $\dfrac{1}{|\Omega|}\sum_{k_b} S^n_{k_b}$

CFL : 0.25,

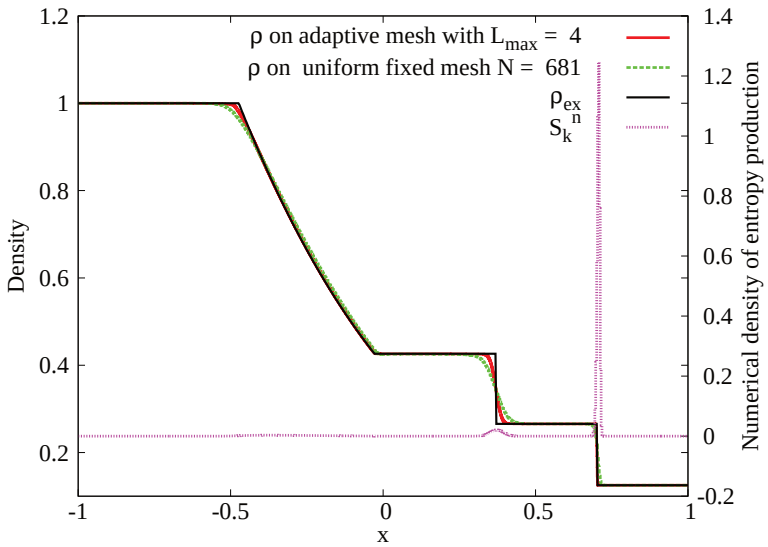
Simulation time (s) : 0.4,

Initial number of cells : 200,

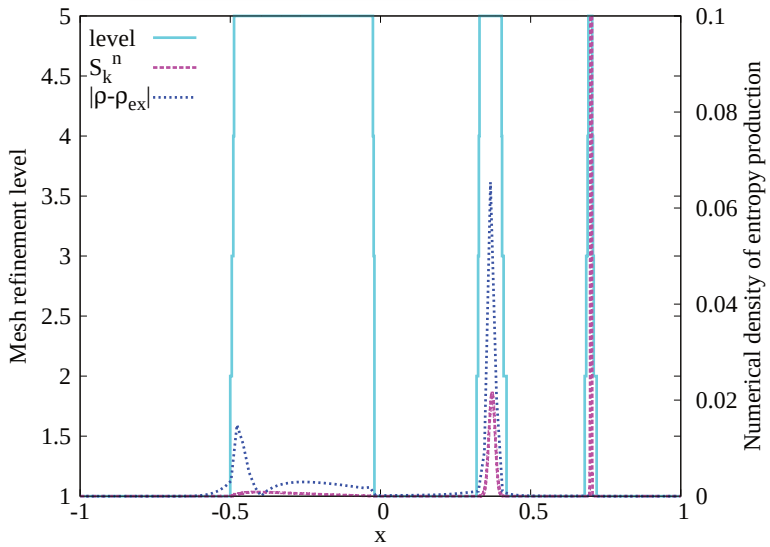
Maximum level of mesh refinement : $L_{\max}$.



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 $L_{\max} = 5$ and the AB1 scheme on a uniform fixed grid of 681 cells.

Theorem ([P04])

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

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

Puppo G., *Numerical entropy production for central schemes.* SIAM J. Sci. Comput., 25(4) :1382–1415, 2004.

Theorem ([P04])

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

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

Properties

Consider a monotone scheme. Then, for almost every k, every n, $S_k^n \leqslant 0$.

Theorem ([P04])

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

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

Properties

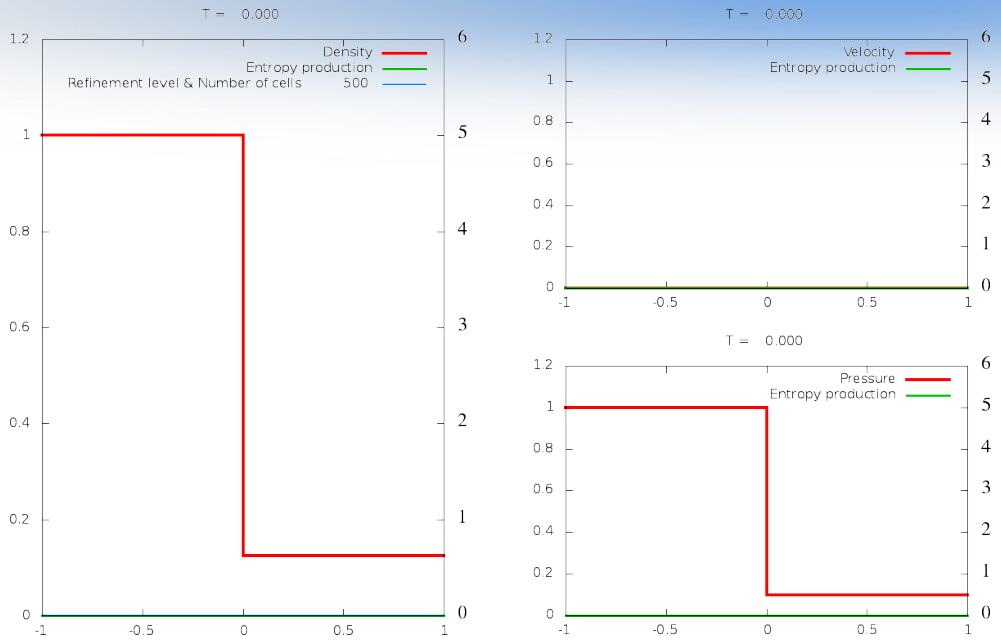
Consider a monotone scheme. Then, for almost every k, every n, $S_k^n \leqslant 0$.

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

▸ See example

Outline

Mesh refinement threshold



Mesh refinement criterion S_k^n and the mesh refinement threshold α

$\vdots$

- $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
- $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened

$\vdots$

Mesh refinement threshold

Mesh refinement criterion S_k^n and the mesh refinement threshold α

$\vdots$

- $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
- $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened

$\vdots$

- how to set α suitably ?

Mesh refinement threshold

Mesh refinement criterion S_k^n and the mesh refinement threshold α

$\vdots$

- $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
- $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened

$\vdots$

- how to set α suitably ? ideally
  - automatically
  - accuracy and computational time are "well-balanced"
    - ⋆ catch the smallest maxima but not the smallest

Mesh refinement threshold

Mesh refinement criterion S_k^n and the mesh refinement threshold α

$\vdots$

- $S_k^n \geqslant \alpha \overline{S} \Longrightarrow$ the cell is refined with $\overline{S} = \dfrac{1}{|\Omega|} \displaystyle\int_\Omega S_k^n$
- $S_k^n < \alpha \overline{S} \Longrightarrow$ the cell is coarsened

$\vdots$

- how to set α suitably ? ideally
  - automatically
  - accuracy and computational time are "well-balanced"
    - ⋆ catch the smallest maxima but not the smallest
    - ⋆ simple method ("low-cost")

- Mean method or deviation

$$S(x,t) > \alpha \frac{1}{|\Omega|} \int_{\Omega} S(x,t)\ dx$$

where α is a tunable dimensionless threshold parameter or deviation method

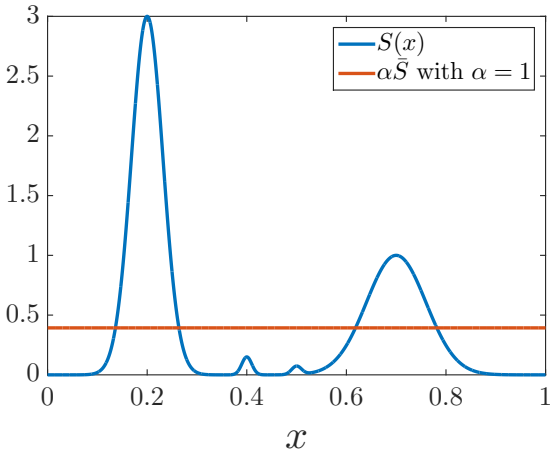
$$S(x,t) > \alpha \frac{1}{|\Omega|} \int_{\Omega} S(x,t)\ dx + \beta\sigma(x,t)$$

where σ is the standard deviation with β is a tunable dimensionless threshold parameter

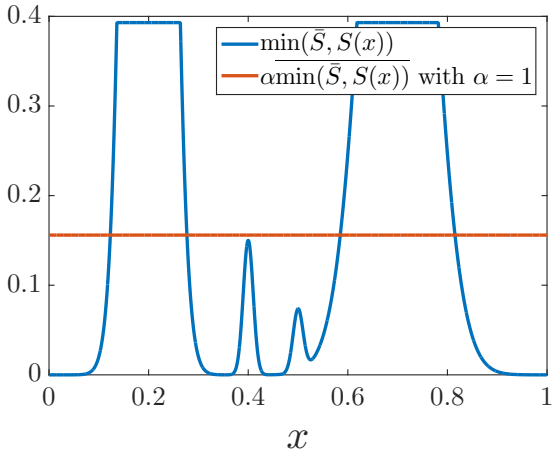

Kallinderis, Y.G., Baron, J.R. : Adaptation methods for a new navier-stokes algorithm. AIAA journal 27(1), 37–43 (1989)


Ersoy, M., Golay, F., Yushchenko, L. : Adaptive multiscale scheme based on numerical density of entropy production for conservation laws. Cent. Eur. J. Math. 11(8), 1392–1415 (2013).

- Mean method or deviation → pb : set α or (α, β)
- Filtering : two-steps



(a) Filtering (mean method) : first step

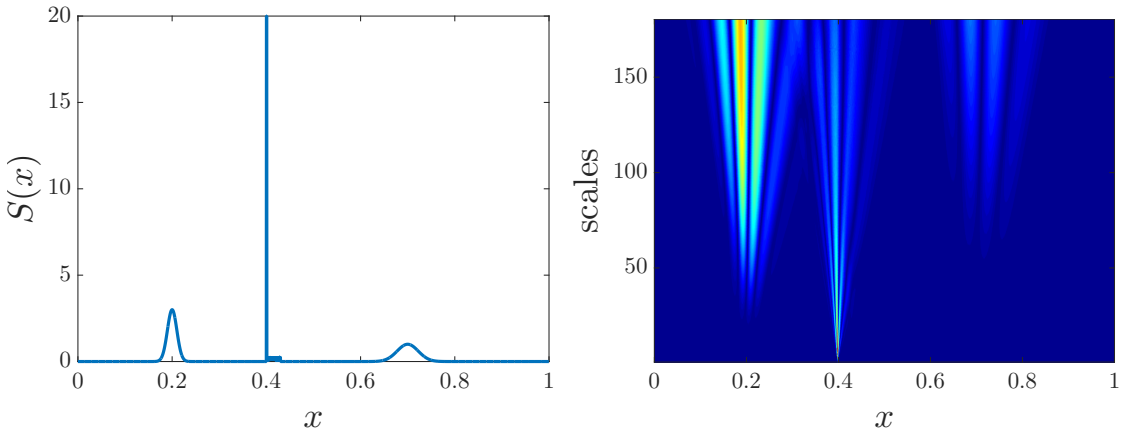


(b) Filtering (mean method) : second step

Aftosmis, M. : Upwind method for simulation of viscous flow on adaptively refined meshes. AIAA journal 32(2), 268–277 (1994)

Warren, G.P., Anderson,W.K., Thomas, J.L., Krist, S.L. : Grid convergence for adaptive methods. AIAA paper 1592, 1991 (1991)

- Mean method or deviation $\rightarrow$ pb : set α or (α, β)
- Filtering : two-steps $\rightarrow$ pb : two-step ? Is it enough ? still to set α
- Filtering : wavelets



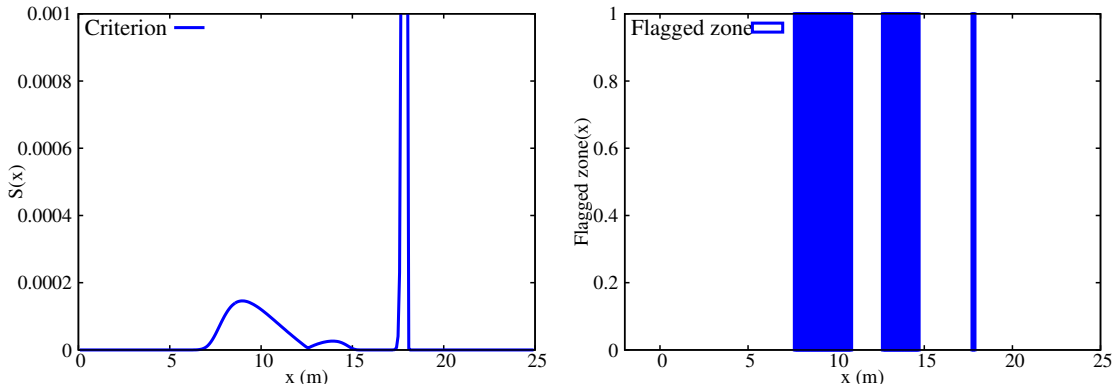
(c) S

(d) Wavelet coefficients ($x =$ space, $y =$ scale)

Figure: Illustration of the wavelet transformation for a given mesh refinement criterion computed with the Daubechies wavelet with four vanishing moments (warm colors correspond to large coefficients and cold colors to small coefficients)

Leonard, S., Terracol, M., Sagaut, P. : A wavelet-based adaptive mesh refinement criterion for large-eddy simulation. Journal of Turbulence 7(64) (2006)

- Mean method or deviation $\rightarrow$ pb : set α or (α, β)
- Filtering : two-steps $\rightarrow$ pb : two-step ? Is it enough ? still to set α
- Filtering : wavelets$\rightarrow$ pb : efficient but the cost ! still to set α
- Local maxima$\rightarrow$ pb : efficient but the cost ! still to set α

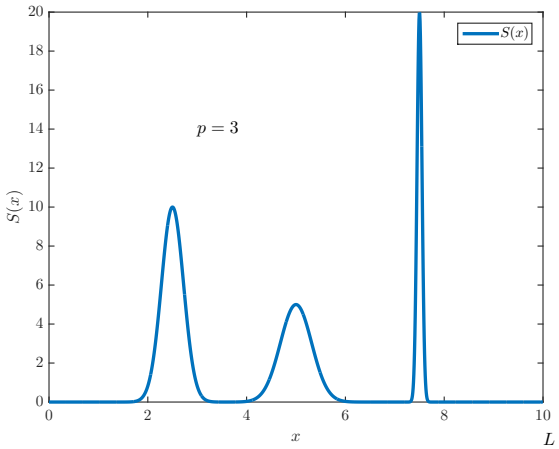


(a) S (b) Region to be refined

Figure: Illustration of the local maxima method for a mesh refinement criterion involving multiple scales

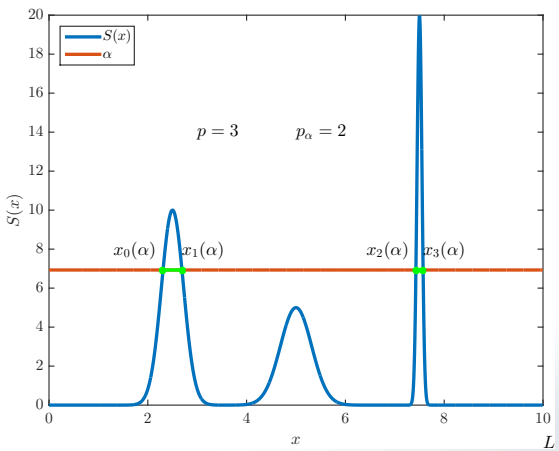
Distribution function (L-meas $\{S(x) > \alpha\}$)

- Assumptions and notations
  - S is smooth and has p local maxima.
  - $S(0) = S(L) = S'(0) = S'(L) = 0$
  - $0 < S_\infty = \max\limits_{x \in (0,L)} S(x) < \infty$



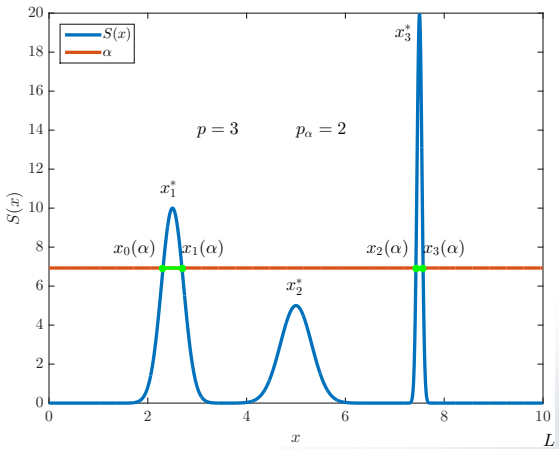
Distribution function (L-meas $\{S(x) > \alpha\}$)

- Assumptions and notations
  - S is smooth and has p local maxima.
  - $S(0) = S(L) = S'(0) = S'(L) = 0$
  - $0 < S_\infty = \max_{x \in (0,L)} S(x) < \infty$
- Then
  - $$\begin{aligned} Z_\alpha &= \{x \in (0,L);\ \varphi_\alpha(x) = S(x) - \alpha = 0 \text{ and } S'(x) \neq 0\} \neq \emptyset \\ &= \{x_0(\alpha) < x_1(\alpha) < \cdots < x_{2p_\alpha - 2}(\alpha) < x_{2p_\alpha - 1}(\alpha)\} \end{aligned}$$
  - $\#Z_\alpha = 2p_\alpha$



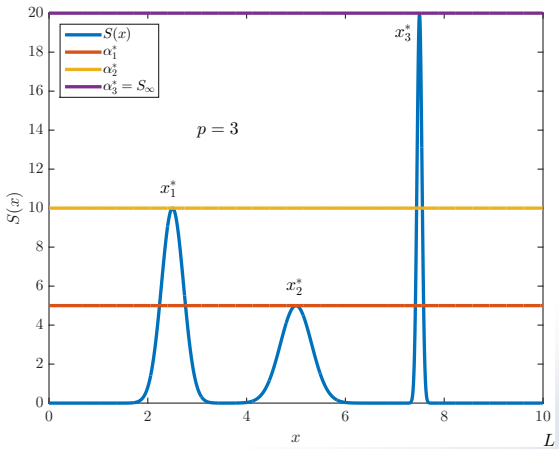
Distribution function (L-meas $\{S(x) > \alpha\}$)

- Assumptions and notations
  - S is smooth and has p local maxima.
  - $S(0) = S(L) = S'(0) = S'(L) = 0$
  - $0 < S_\infty = \max_{x \in (0,L)} S(x) < \infty$
- Then
  - $$\begin{aligned} Z_\alpha &= \{x \in (0, L);\ \varphi_\alpha(x) = S(x) - \alpha = 0 \text{ and } S'(x) \neq 0\} \neq \emptyset \\ &= \{x_0(\alpha) < x_1(\alpha) < \cdots < x_{2p_\alpha - 2}(\alpha) < x_{2p_\alpha - 1}(\alpha)\} \end{aligned}$$
  - $\# Z_\alpha = 2p_\alpha$
  - there exist sequence $(x_k^*)_{1 \leqslant k \leqslant p}$



Distribution function (L-meas $\{S(x) > \alpha\}$)

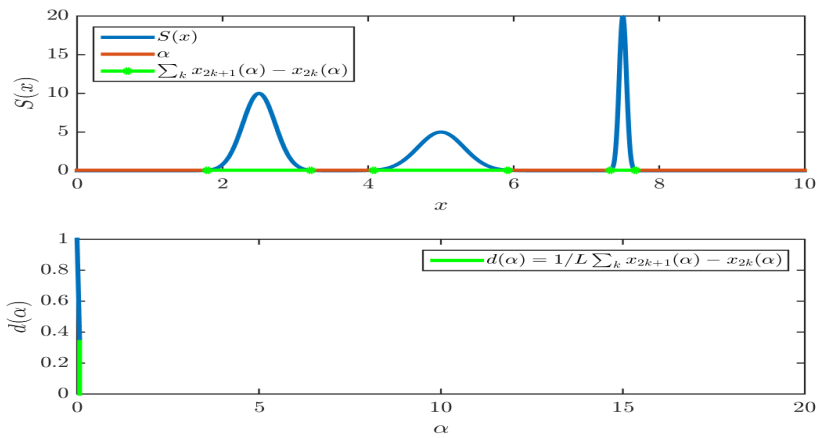
- Assumptions and notations
  - S is smooth and has p local maxima.
  - $S(0) = S(L) = S'(0) = S'(L) = 0$
  - $0 < S_\infty = \max\limits_{x \in (0,L)} S(x) < \infty$
- Then
  - $$\begin{aligned} Z_\alpha &= \{x \in (0,L);\ \varphi_\alpha(x) = S(x) - \alpha = 0 \text{ and } S'(x) \neq 0\} \neq \emptyset \\ &= \{x_0(\alpha) < x_1(\alpha) < \cdots < x_{2p_\alpha - 2}(\alpha) < x_{2p_\alpha - 1}(\alpha)\} \end{aligned}$$
  - $\# Z_\alpha = 2p_\alpha$
  - there exist sequences $(x_k^*)_{1 \leqslant k \leqslant p}$ and $(\alpha_k^*)_{1 \leqslant k \leqslant p}$ such that
    - $\forall k = 1, \ldots, p$
    - $S(x_k^*) = \alpha_k^*$ and $S'(x_k^*) = 0,\ S''(x_k^*) < 0$
    - $x_k^* \notin Z_\alpha$ and $\alpha_p^* = S_\infty$



Distribution function (L-meas $\{S(x) > \alpha\}$)

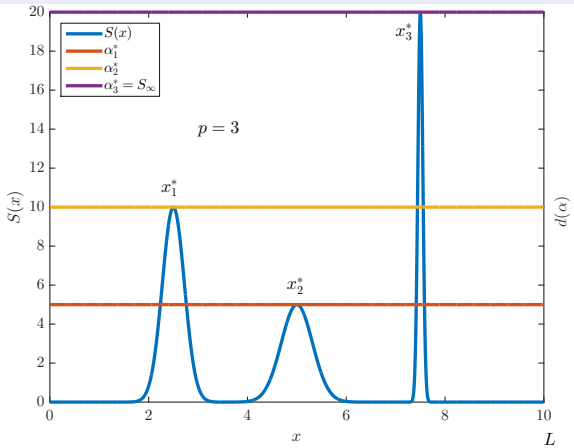
- Assumptions and notations
  - S is smooth and has p local maxima.
  - $S(0) = S(L) = S'(0) = S'(L) = 0$
  - $0 < S_\infty = \max_{x \in (0,L)} S(x) < \infty$
- With these settings, we define the distribution d

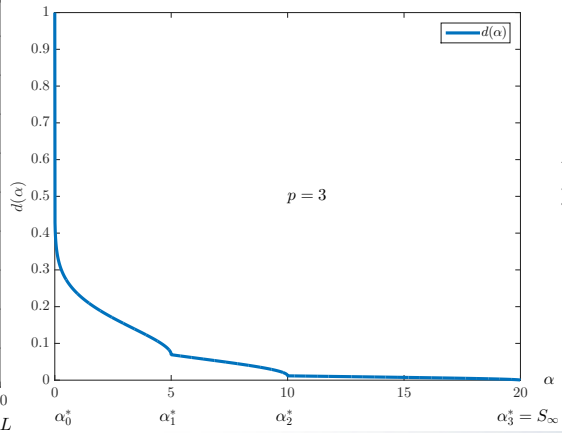
$$\alpha \in [0, S_\infty] \mapsto \; d(\alpha) := \begin{cases} 1 & \text{if} \quad \alpha = 0 \,, \\ \dfrac{1}{L} \displaystyle\sum_{k=1}^{p_\alpha} x_{2k+1}(\alpha) - x_{2k}(\alpha) & \text{if} \quad 0 < \alpha < S_\infty \,, \\ 0 & \text{if} \quad \alpha = S_\infty \,. \end{cases}$$

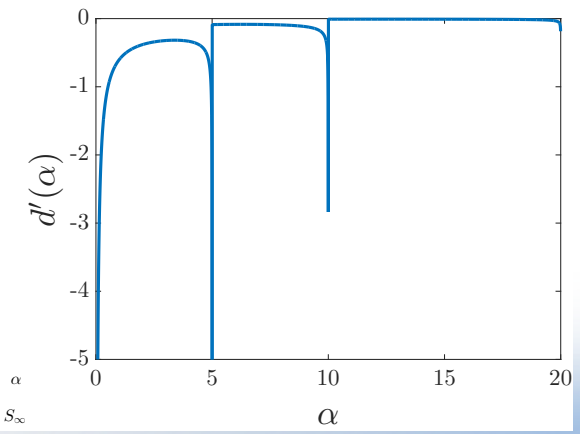


Properties

1. $\int_0^{S_\infty} d(\alpha)\, d\alpha = \int_0^{L} S(x)\, dx.$
2. $d \in C^0([0, S_\infty], \mathbb{R}^+)$ *and* d' *satisfies*
   1. $\forall \alpha \in [0, S_\infty],\ d'(\alpha) < 0$
   2. $\forall k \in [\![0, p]\!],\ \lim_{\alpha \to \alpha_k^*} d'(\alpha) = -\infty$ *with the convention* $\alpha_0^* := 0$
3. $d \in C^l\left(D_*, \mathbb{R}^+\right)$ *on the set* $D^* := \bigcup_{k=0}^{p-1} (\alpha_k^*, \alpha_{k+1}^*)$.

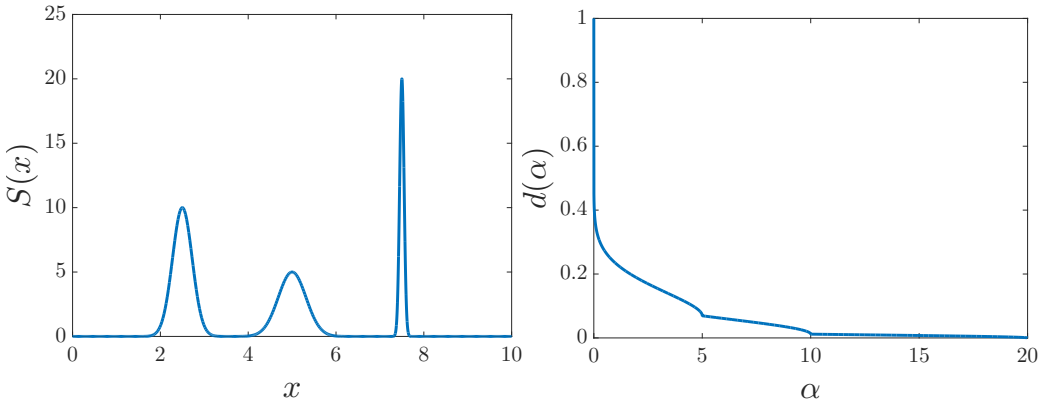






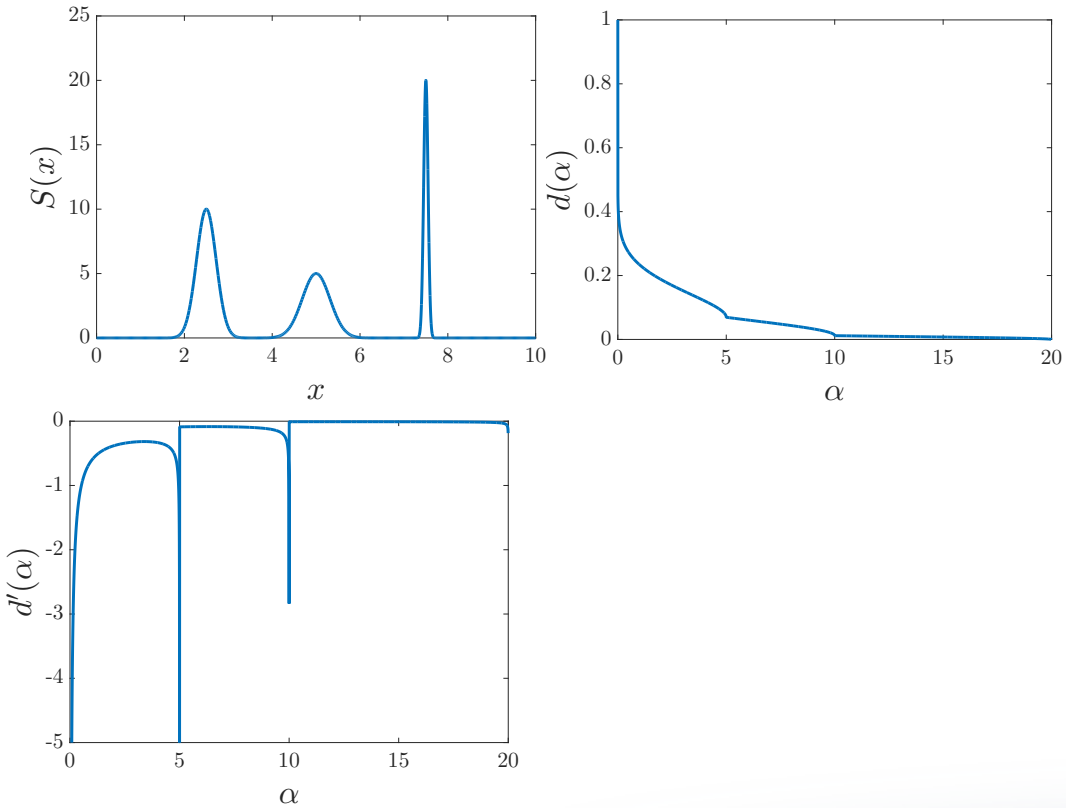
Automatic thresholding : distribution function (L-meas $\{S(x) > \alpha\}$)

- Hard to define α using only d



Automatic thresholding : distribution function (L-meas $\{S(x) > \alpha\}$)

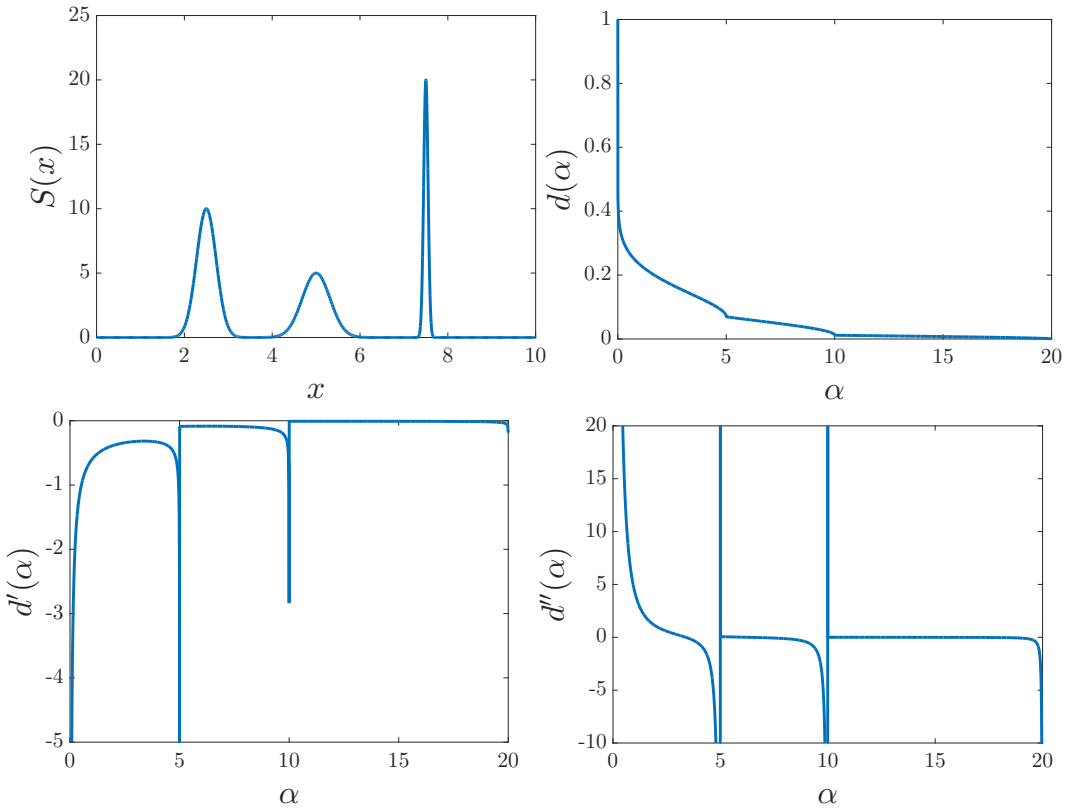
- Hard to define α using only d
- However, one can set $\alpha{=}\alpha_D$ or
  - either $\alpha = \alpha_D = \min\{\alpha \mid d'(\alpha) = -1\}$



Dannenhoffer, J.F. : Grid adaptation for complex two-dimensional transonic flows. Ph.D. thesis, Massachusetts Institute of Technology (1987)

Automatic thresholding : distribution function (L-meas $\{S(x) > \alpha\}$)

- Hard to define α using only d
- However, one can set $\alpha{=}\alpha_D$ or $\alpha = \alpha_P$
  - either $\alpha = \alpha_P = \min\{\alpha \mid d''(\alpha) = 0\}$

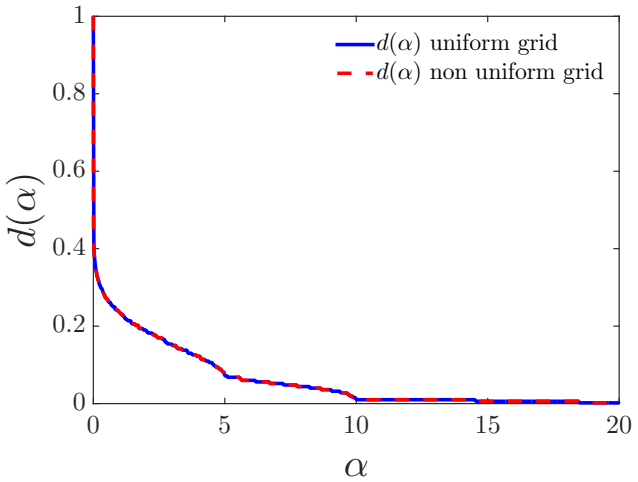


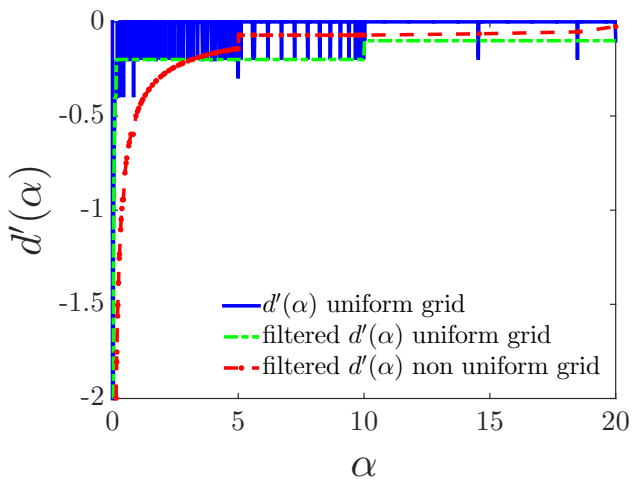
Powell, K.G., Murman, E.M. : An embedded mesh procedure for leading-edge vortex flows. In : NASA, Langley Research Center, Transonic Symposium : Theory, Application, and Experiment, vol. 1 (1989)

Automatic thresholding : distribution function (L-meas $\{S(x) > \alpha\}$)

- Hard to define α using only d
- However, one can set $\alpha{=}\alpha_D$ or $\alpha = \alpha_P$
- In practice, d is defined by $d(\alpha) = \sum_{j=0}^{M-1} d_j \; \mathbb{1}_{(\alpha_j, \alpha_{j+1})}(\alpha)$ with

$$d_j = \#\{k \; ; \; S_k^n > \alpha_j\} \text{ and } (\alpha_j)_{0 \leqslant j \leqslant M} = \left(S_m \left(\frac{j}{M} \right)^{\beta} \right)_{0 \leqslant j \leqslant M} \qquad \beta \geqslant 1$$





Dannenhoffer, J.F. : Grid adaptation for complex two-dimensional transonic flows. Ph.D. thesis, Massachusetts Institute of Technology (1987)

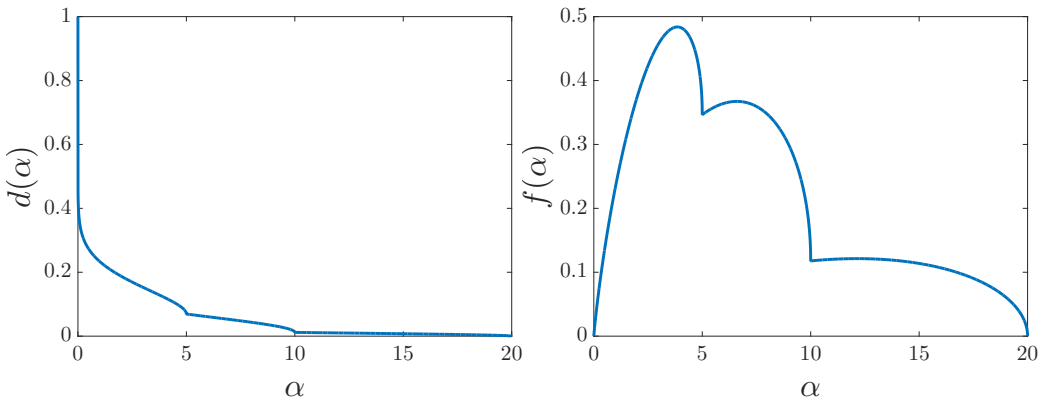
Automatic thresholding : distribution function (L-meas $\{S(x) > \alpha\}$)

- Hard to define α using only d
- However, one can set $\alpha{=}\alpha_D$ or $\alpha = \alpha_P$
- α is usually too small (too sensitive to β) ! ! ! $\rightarrow$ detects small fluctuations

Automatic thresholding : distribution function (L-meas $\{S(x) > \alpha\}$)

- Hard to define α using only d
- However, one can set $\alpha{=}\alpha_D$ or $\alpha = \alpha_P$
- α is usually too small (too sensitive to β) ! ! ! $\rightarrow$ detects small fluctuations
- We propose to set

$$\alpha = \alpha_{PE} = \min\{\alpha \in [0, \bar{S}] \mid \max\{\alpha d(\alpha)\}\}$$



Pons, K., Ersoy, M. Adaptive mesh refinement method. Part 1 : Automatic thresholding based on a distribution function

Pons, K., Ersoy, M. Adaptive mesh refinement method. Part 2 : Application to tsunamis propagation

Properties

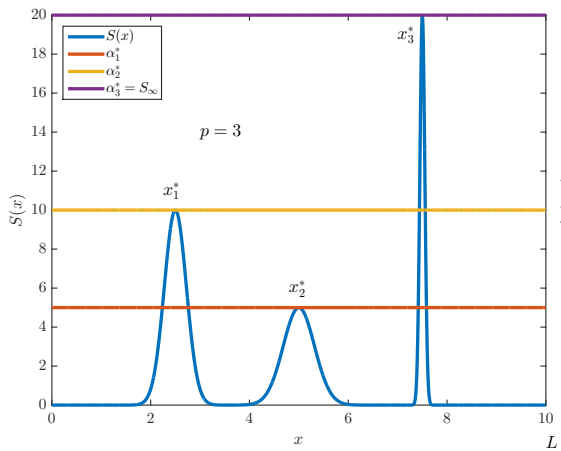
Assume that S is twice differentiable and has p local maxima. Then, there exists

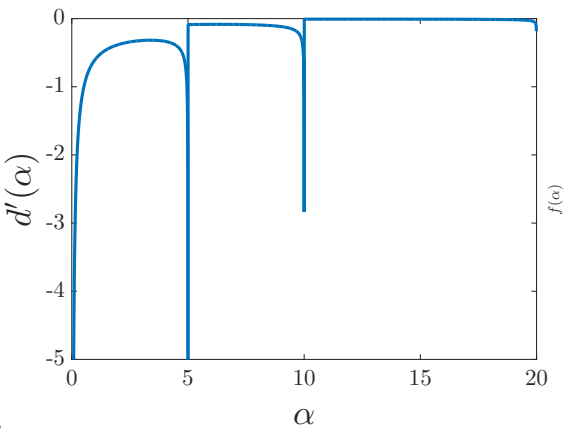
$$\forall k = 0 \dots p-1,\ \alpha^{**}_{k+1} \in (\alpha^{*}_{k}, \alpha^{*}_{k+1}) \text{ such that } d''(\alpha^{**}_{k+1}) = 0$$

and the function f has p local maxima $\overline{\alpha}_1, \dots, \overline{\alpha}_p$ such that

$$\forall k = 1 \dots p,\ \overline{\alpha}_k \in (\alpha^{**}_{k}, \alpha^{*}_{k})\ .$$

As a consequence $\alpha_{PE} > \alpha_P > \alpha_D$.





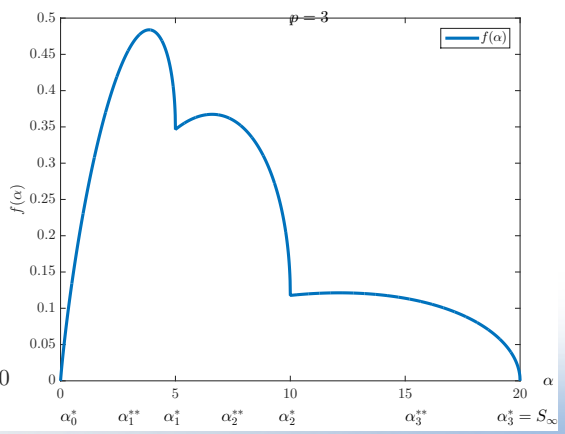
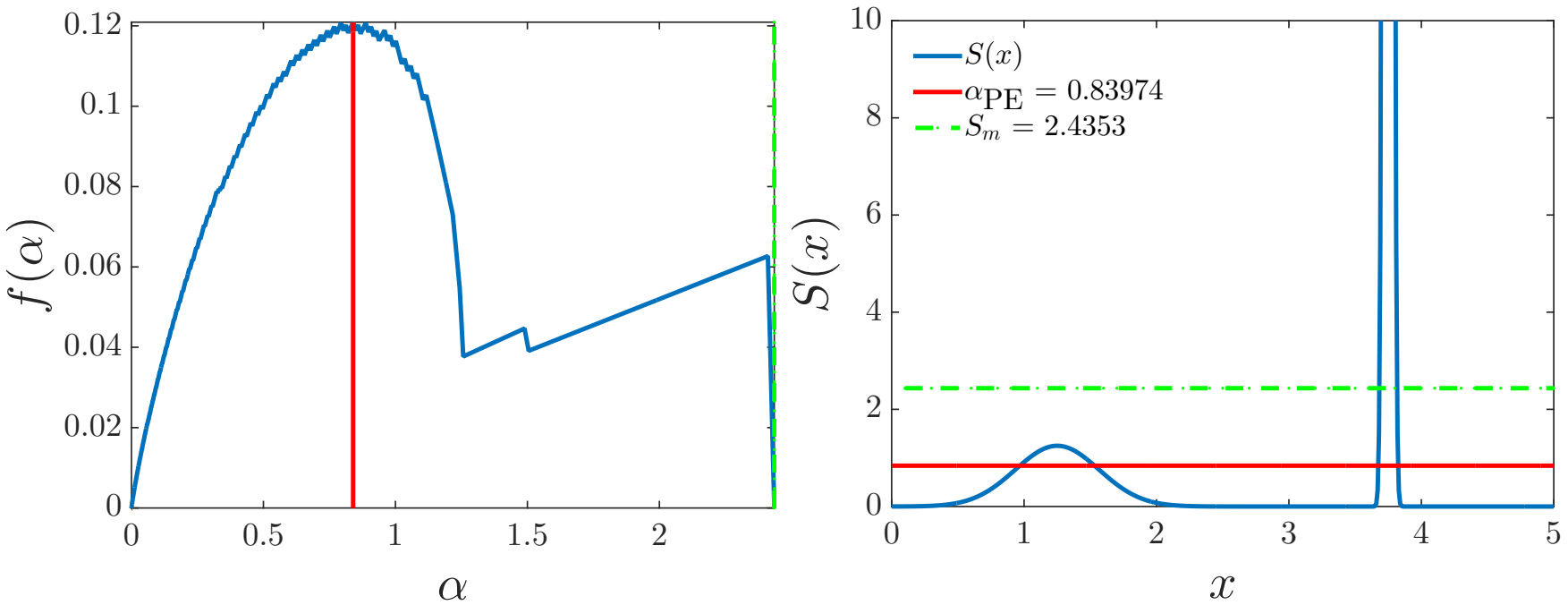


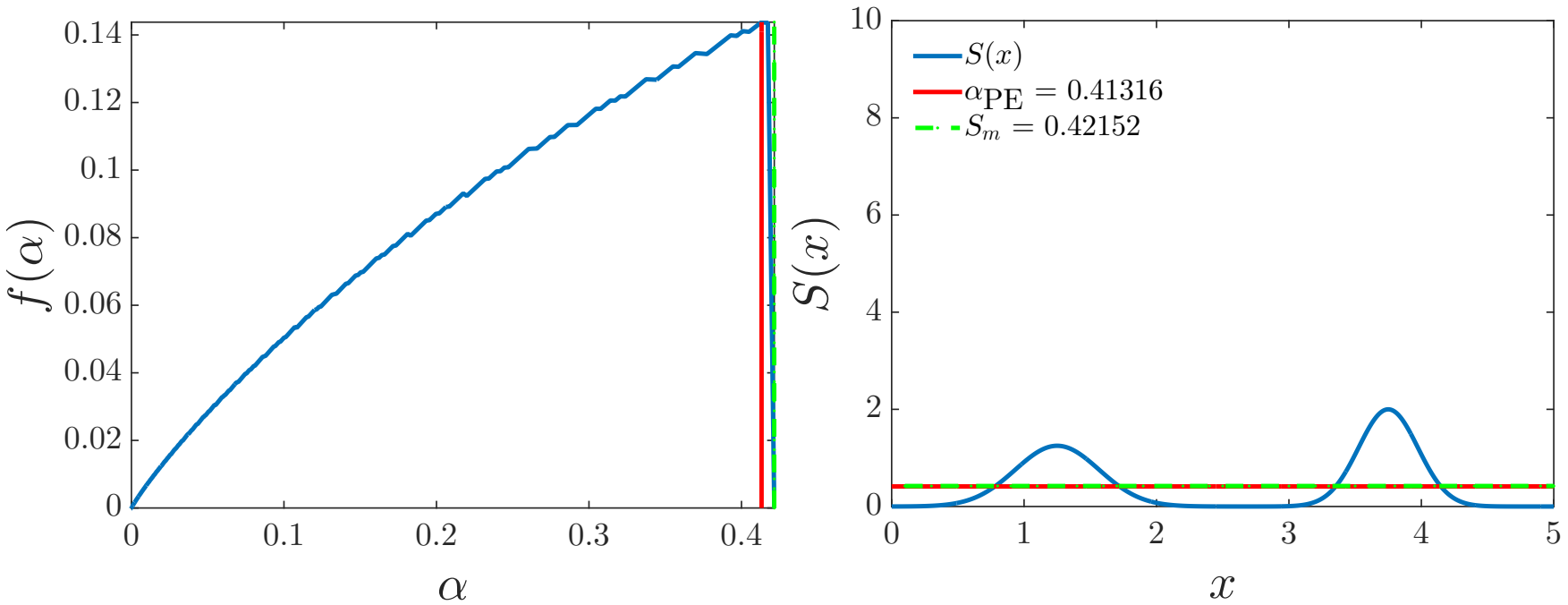
Illustration : α_{PE} threshold for "discontinuous" flows



(a) $f(\alpha)$ (b) $S(x)$

Figure: The function f for the mesh refinement criterion $S(x) = 200\exp(-1000(x-1.25)^2) + 1.25\exp(-5(x-3.75)^2)$ representing a shock type solution

Illustration : α_{PE} threshold for "smooth" flows



(a) $f(\alpha)$ (b) $S(x)$

Figure: The function f for the mesh refinement criterion $S(x) = 2\exp(-10(x-1.25)^2) + 1.25\exp(-5(x-3.75)^2)$ representing a smooth flow

Outline

A Dam-break problem for the Saint-Venant equations :

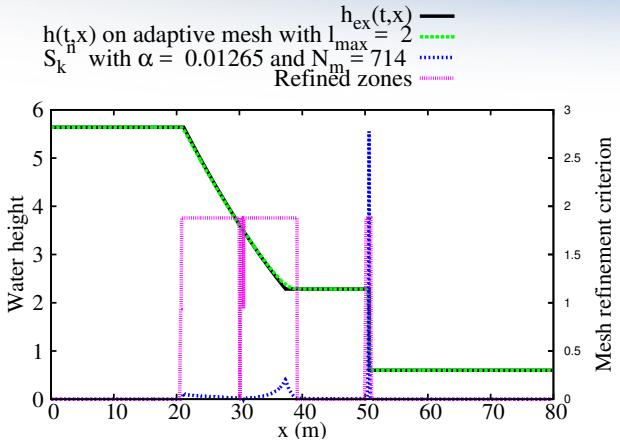
$$\begin{cases} \partial_t h + \partial_x (hu) = 0, \\ \partial_t (hu) + \partial_x \left(hu^2 + gh^2/2 \right) = -ghZ'(x) \end{cases}$$

where

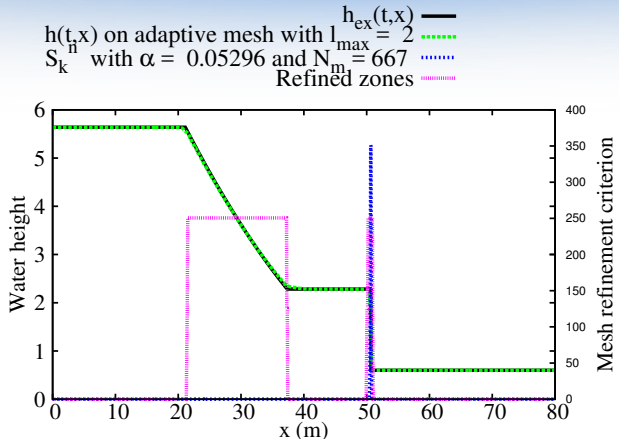
$h(t,x)$	:	density
$u(t,x)$	:	velocity of the water column
g	:	gravity strength
Z	:	topography

Kévin Pons, Mehmet Ersoy. Adaptive mesh refinement method. Part 1 : Automatic thresholding based on a distribution function

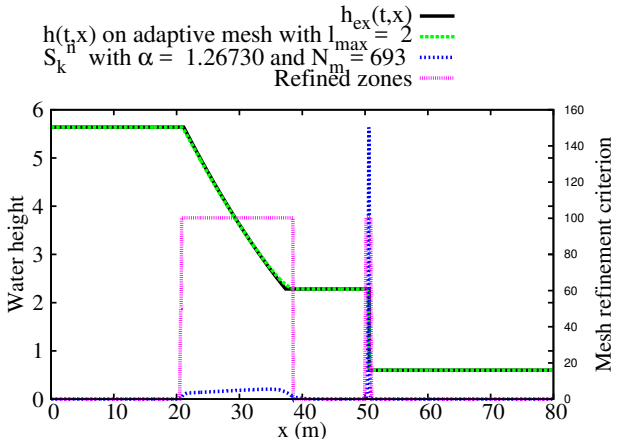
Numerical validation[PE]



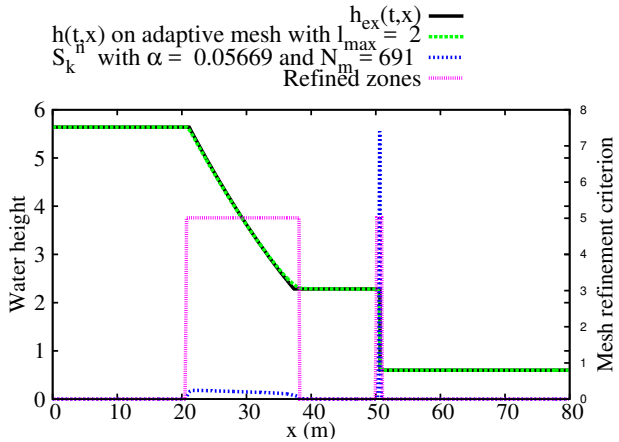
(a) A priori error



(b) Numerical density of entropy production

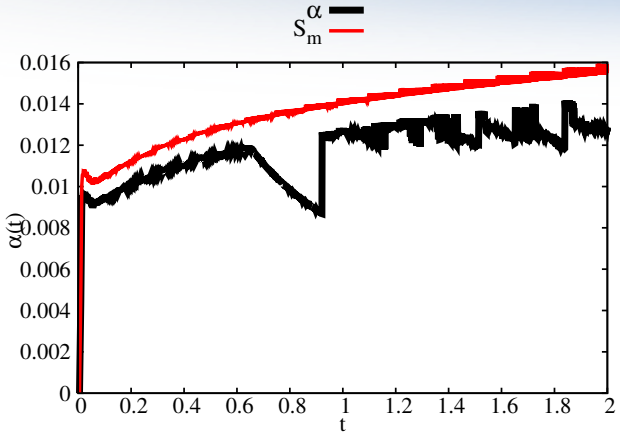


(c) A posteriori error

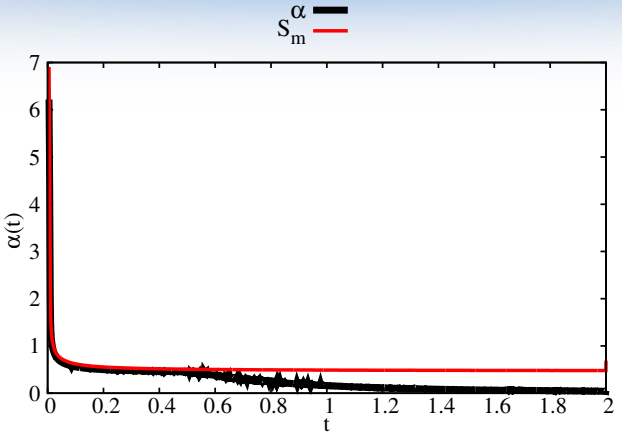


(d) Gradient of h

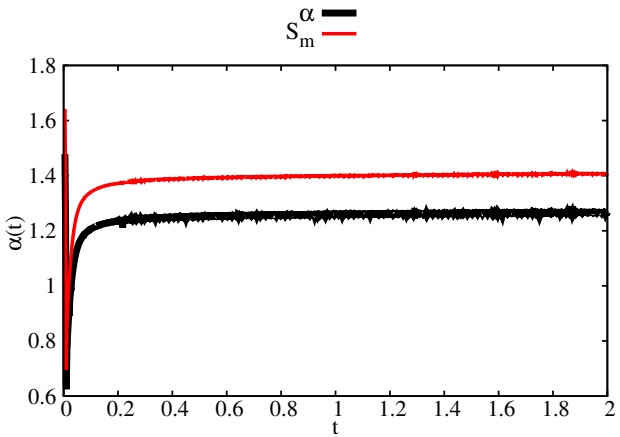
Figure: Numerical results for the water height at time $t = 2$ s



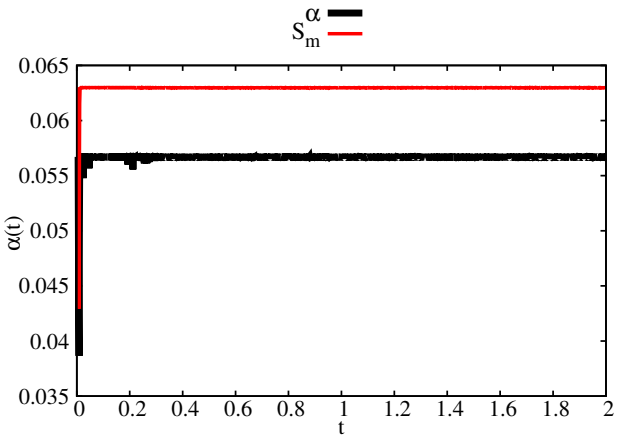
(a) A priori error



(b) Numerical density of entropy production



(c) A posteriori error



(d) Gradient of h

Figure: Time evolution of the threshold parameter and the mean value S_m

Test case I : Solitary wave propagation over a two dimensional reef

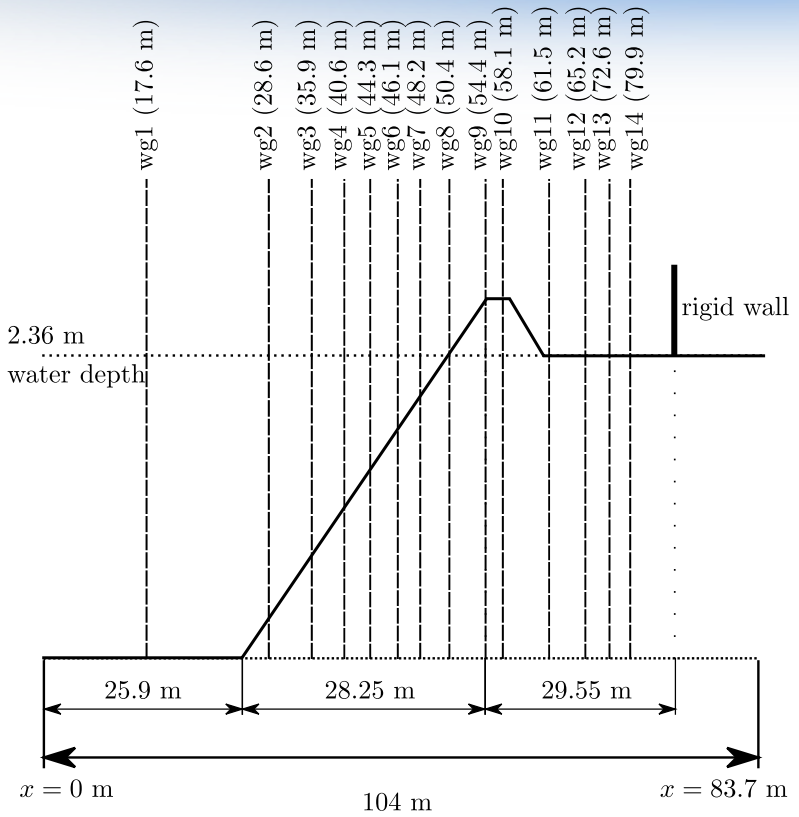


Figure: Experimental settings and wave gauges locations

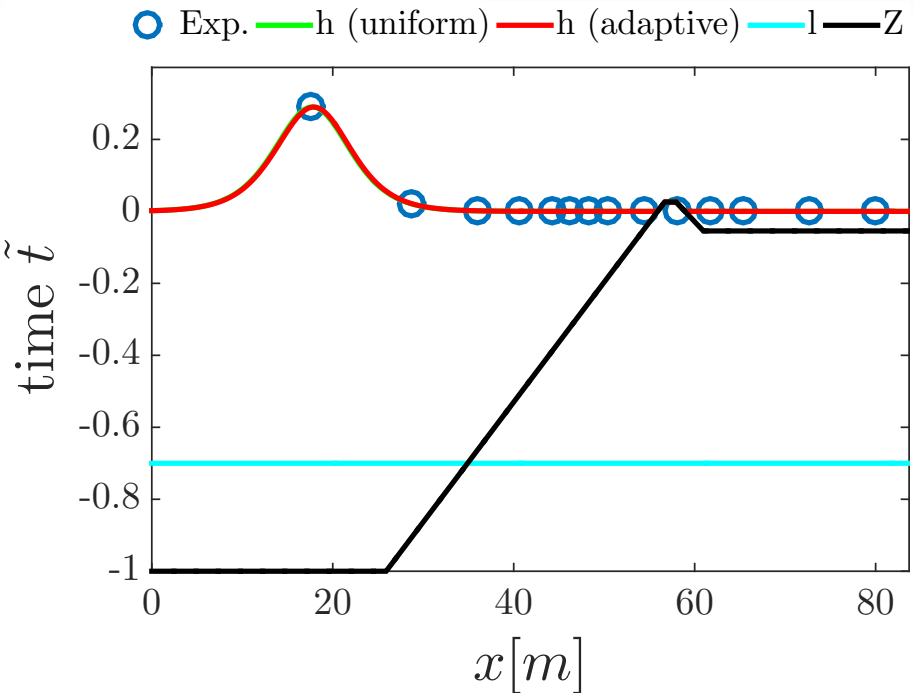
Kévin Pons, Mehmet Ersoy, Frédéric Golay, Richard Marcer. Adaptive mesh refinement method. Part 2 : Application to tsunamis propagation

Roeber, V., Cheung, K.F. : Boussinesq-type model for energetic breaking waves in fringing reef environments. Coastal Engineering 70, 1–20 (2012)

Roeber, V., Cheung, K.F., Kobayashi, M.H. : Shock-capturing boussinesq-type model for nearshore wave processes. Coastal Engineering 57(4),

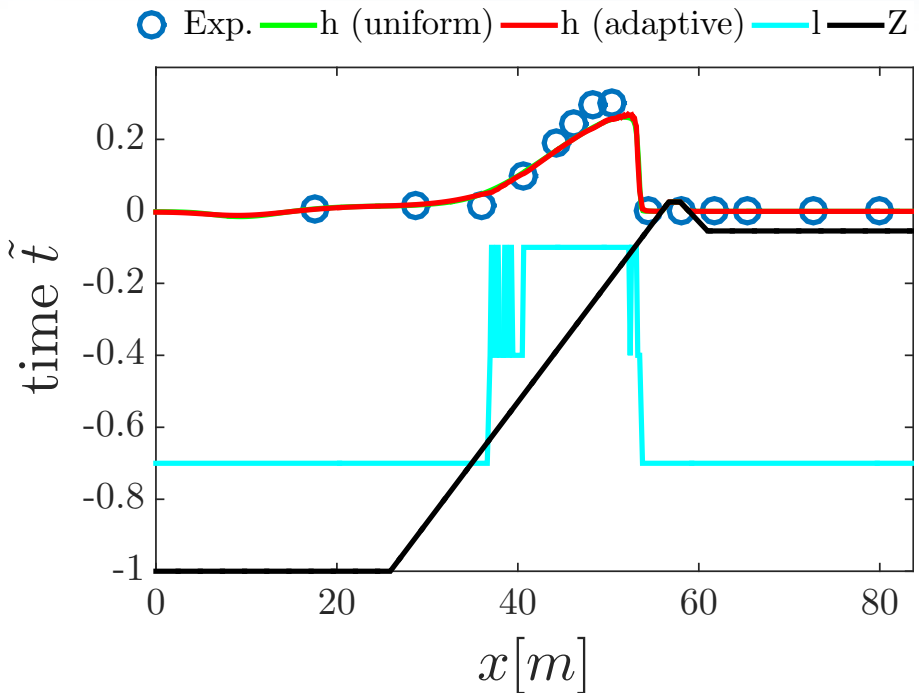
	Adaptive mesh simulation	Uniform mesh simulation
Simulation time	240	240
Number of cells	200-560	1000
Re-meshing time step	0.05 s	not applicable
Time order integration	1	1
Space order integration	1	1
CFL	0.99	0.99

Table: Numerical parameters



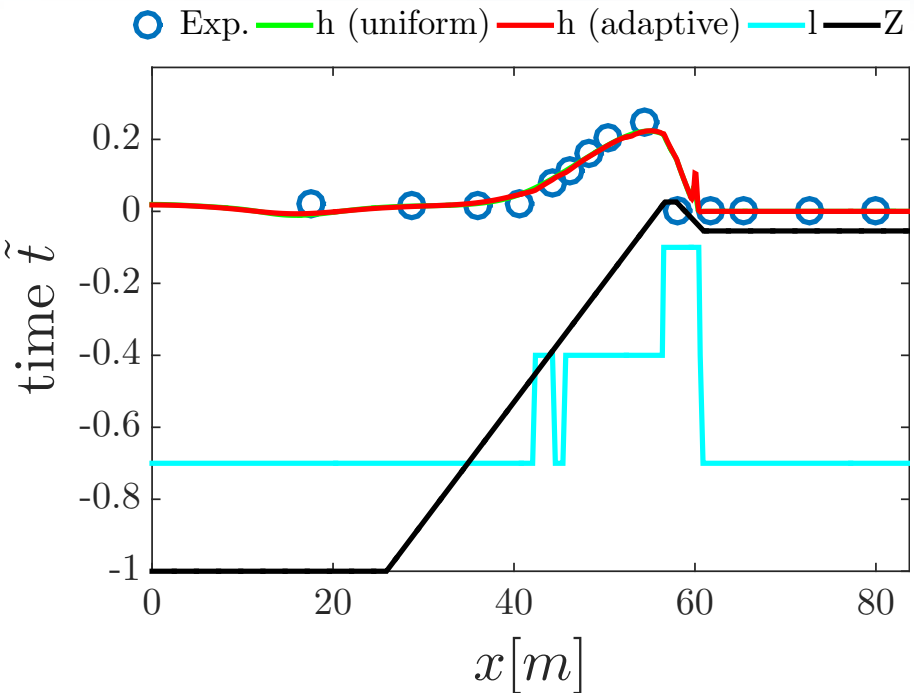
(a) $\tilde{t} = 55.03$

Figure: Surface profiles of solitary wave propagation over an exposed reef crest. Confrontation of experimental data (blue circles) to numerical data computed on a uniform grid (solid green line) and on an adaptive grid (solid red lines). The solid cyan line represents the mesh level and the black one the bathymetry



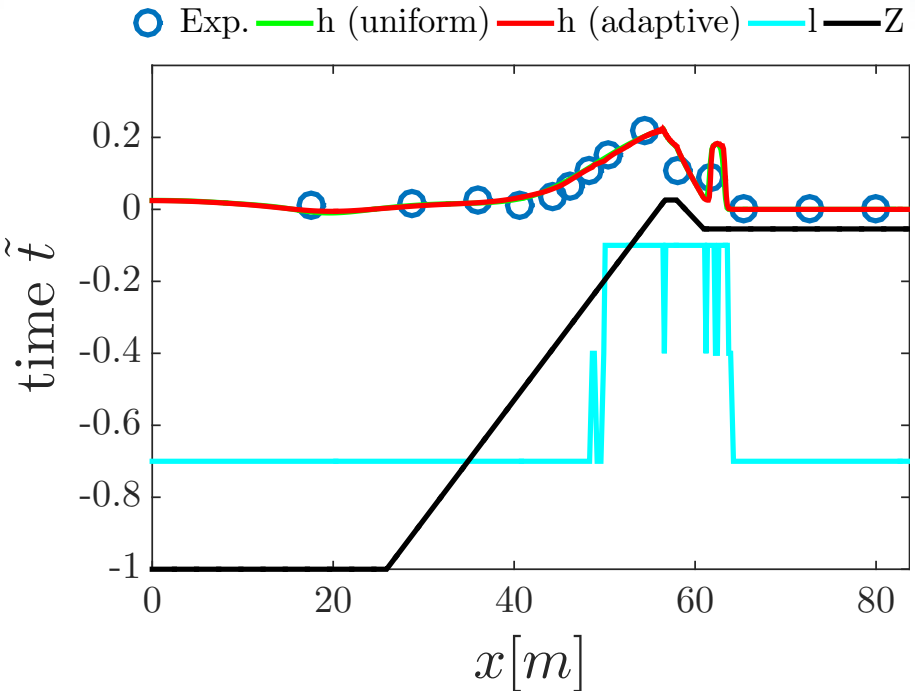
(a) $\tilde{t} = 66.53$

Figure: Surface profiles of solitary wave propagation over an exposed reef crest. Confrontation of experimental data (blue circles) to numerical data computed on a uniform grid (solid green line) and on an adaptive grid (solid red lines). The solid cyan line represents the mesh level and the black one the bathymetry



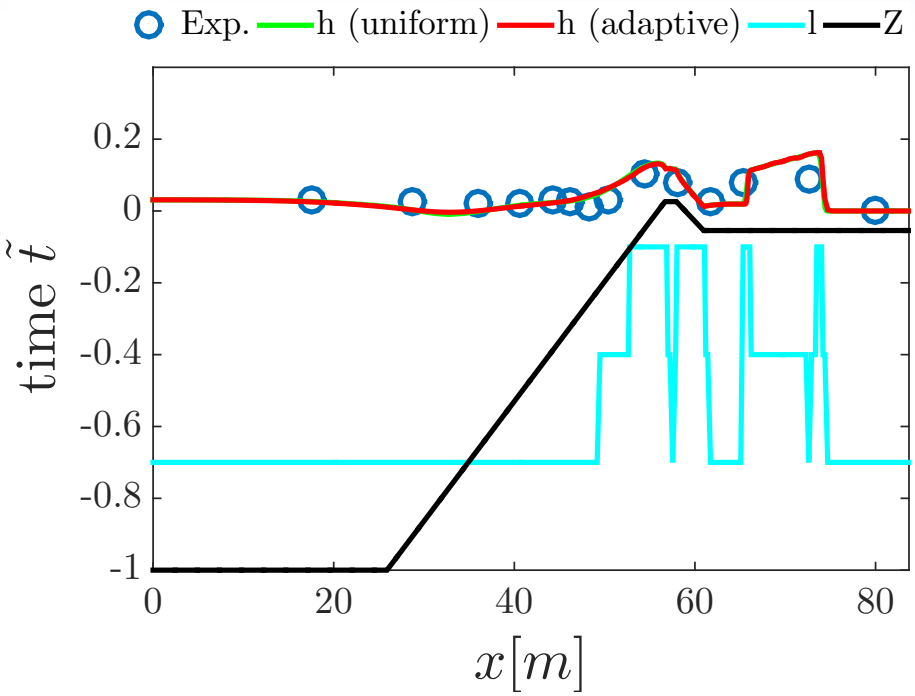
(a) $\tilde{t} = 69.13$

Figure: Surface profiles of solitary wave propagation over an exposed reef crest. Confrontation of experimental data (blue circles) to numerical data computed on a uniform grid (solid green line) and on an adaptive grid (solid red lines). The solid cyan line represents the mesh level and the black one the bathymetry



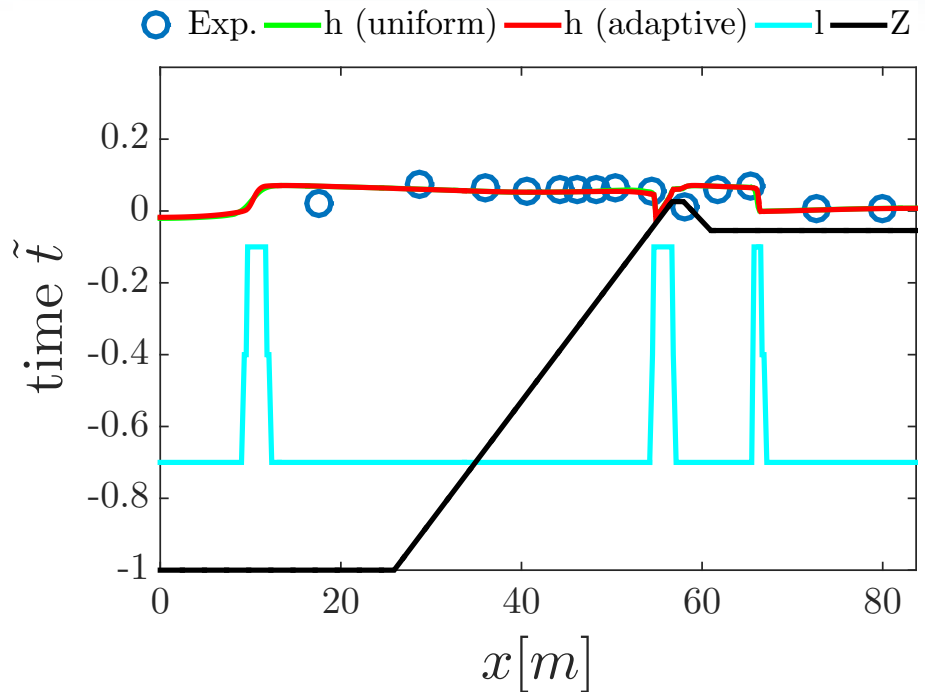
(a) $\tilde{t} = 70.68$

Figure: Surface profiles of solitary wave propagation over an exposed reef crest. Confrontation of experimental data (blue circles) to numerical data computed on a uniform grid (solid green line) and on an adaptive grid (solid red lines). The solid cyan line represents the mesh level and the black one the bathymetry



(a) $\tilde{t} = 76.33$

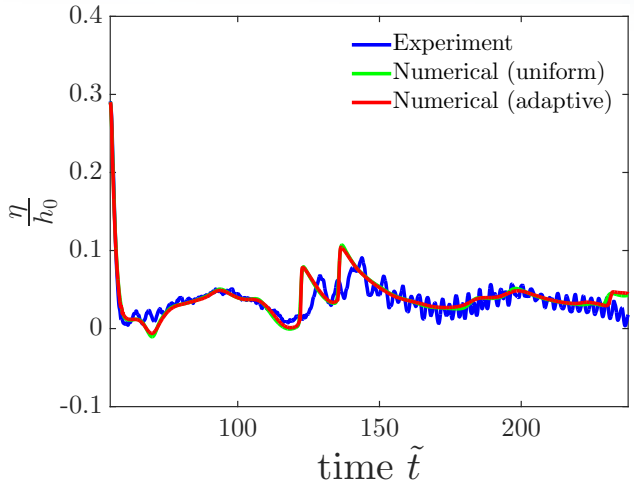
Figure: Surface profiles of solitary wave propagation over an exposed reef crest. Confrontation of experimental data (blue circles) to numerical data computed on a uniform grid (solid green line) and on an adaptive grid (solid red lines). The solid cyan line represents the mesh level and the black one the bathymetry



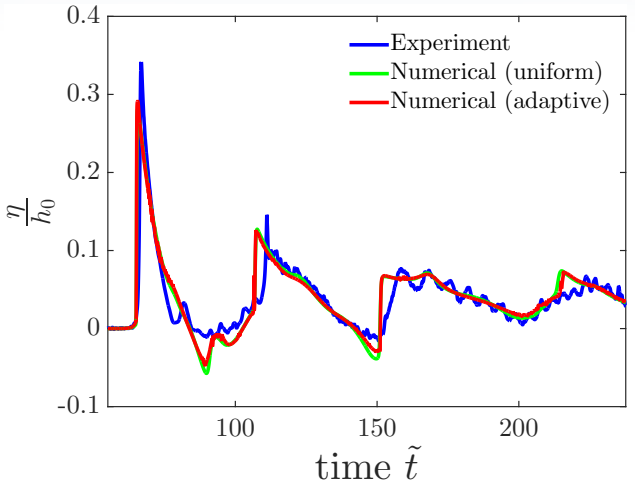
(a) $\tilde{t} = 125.03$

Figure: Surface profiles of solitary wave propagation over an exposed reef crest. Confrontation of experimental data (blue circles) to numerical data computed on a uniform grid (solid green line) and on an adaptive grid (solid red lines). The solid cyan line represents the mesh level and the black one the bathymetry

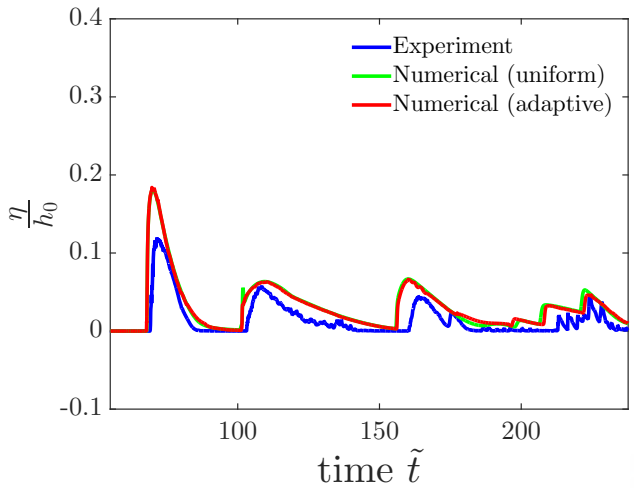
Test case I : Solitary wave propagation over a two dimensional reef



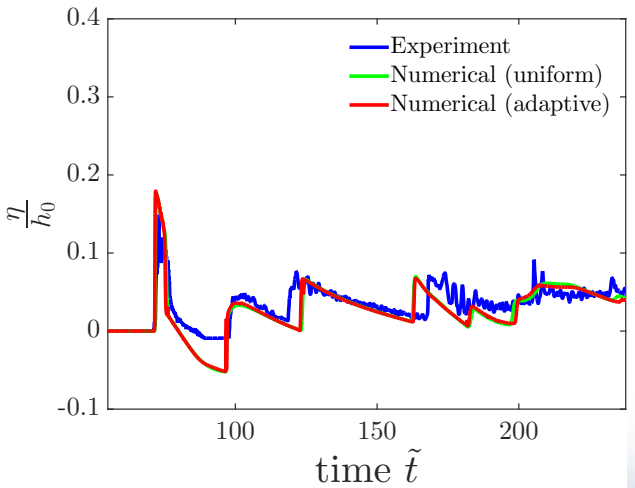
(a) $x = 17.6$m



(b) $x = 50.4$m

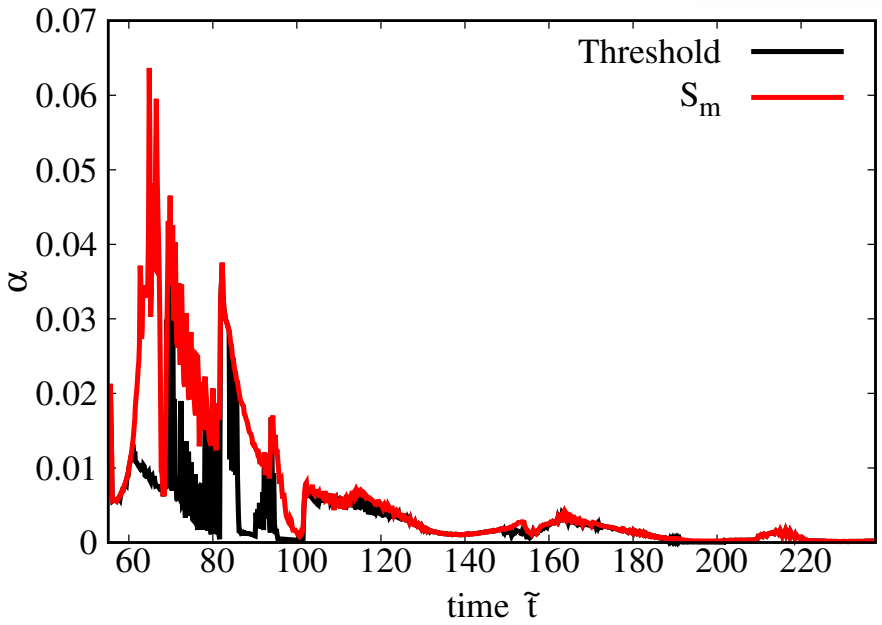


(c) $x = 58.1$m

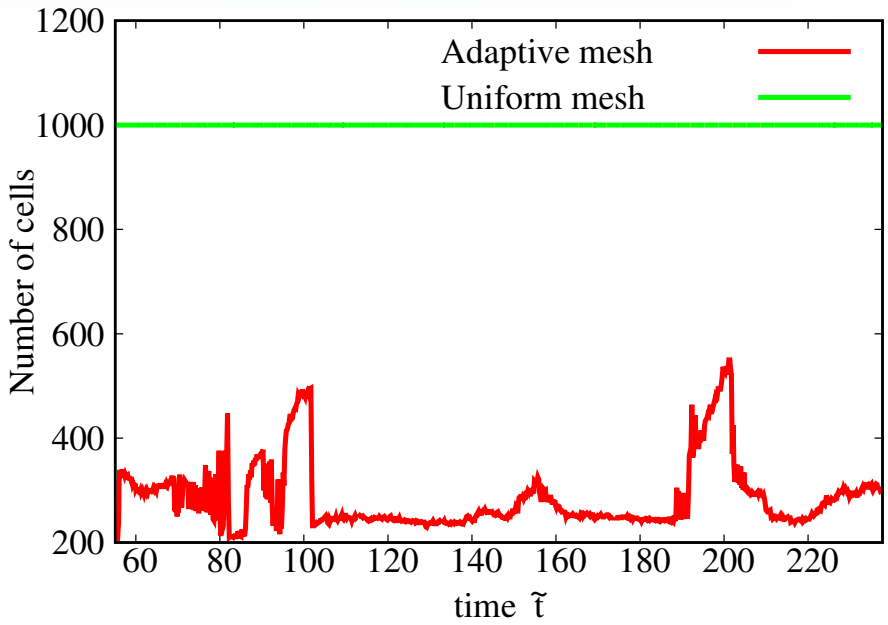


(d) $x = 65.2$m

Test case I : Solitary wave propagation over a two dimensional reef



(a) Threshold



(b) Number of cells

Figure: Time evolution of the mesh refinement threshold and the number of cells : 95 s against 210 s

Test case II : Solitary wave propagation over an irregular 3-d shallow shelf

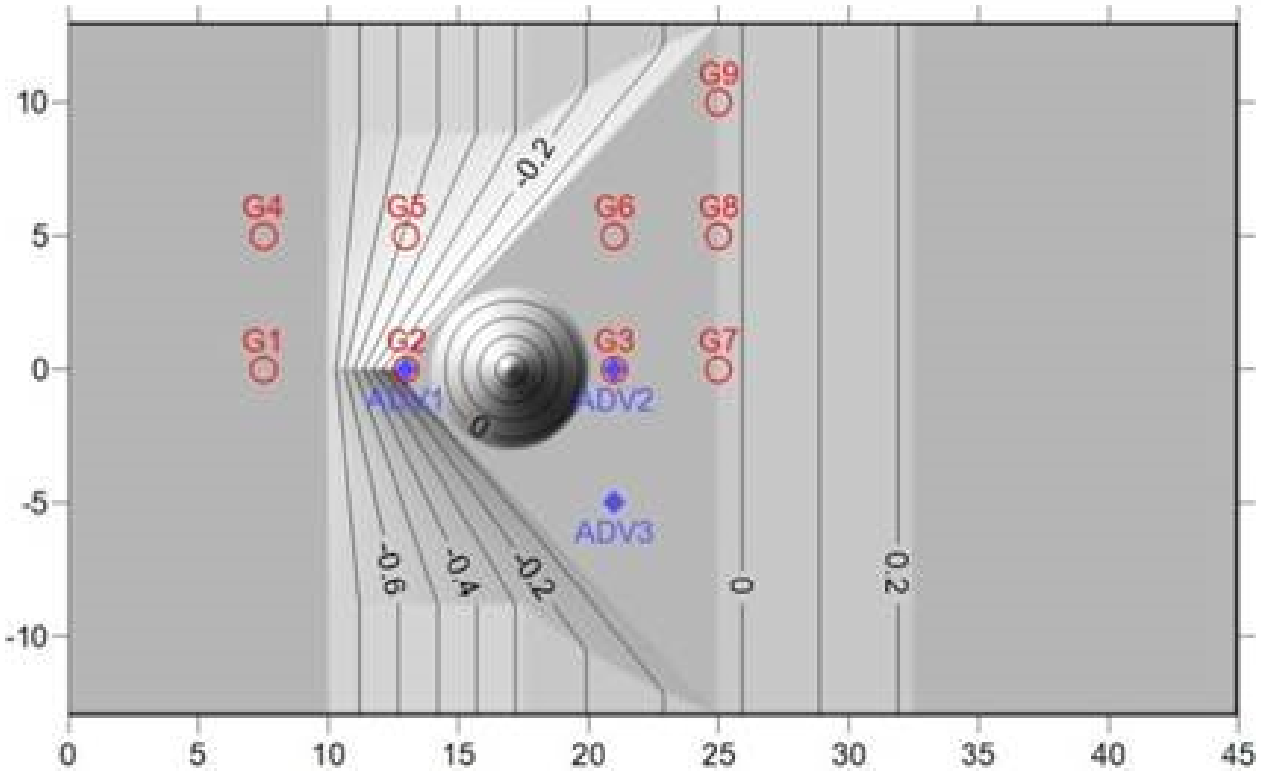


Figure: Experimental settings

Kévin Pons, Mehmet Ersoy, Frédéric Golay, Richard Marcer. Adaptive mesh refinement method. Part 2 : Application to tsunamis propagation.

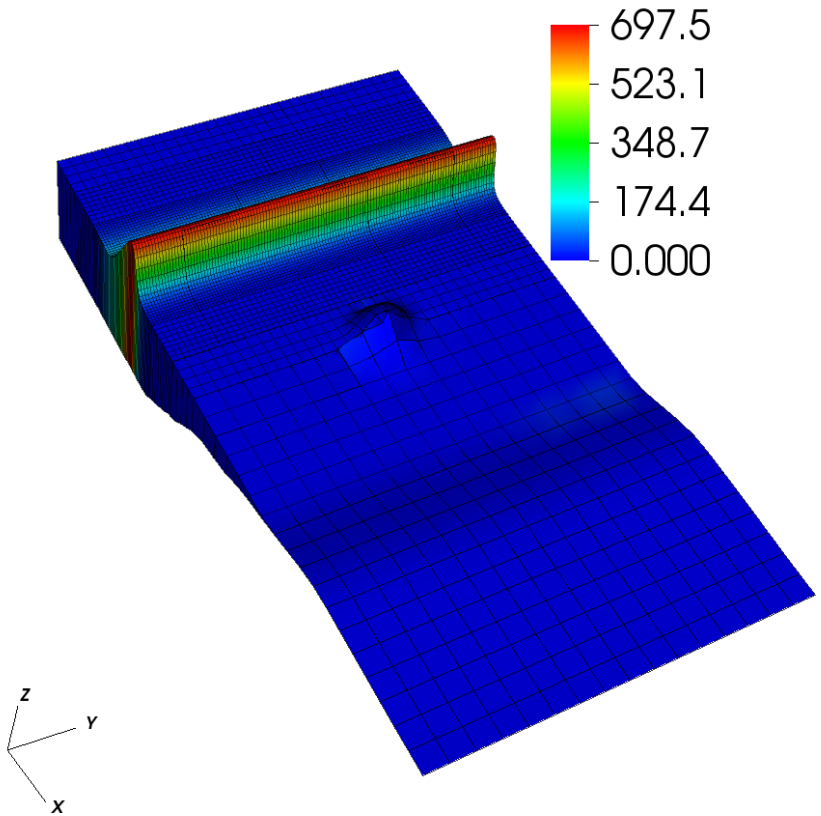
Lynett, P.J., Swigler, D., Son, S., Bryant, D., Socolofsky, S. : Experimental study of solitary wave evolution over a 3d shallow shelf. Coastal Engineering Proceedings 1(32), 1 (2011).

Test case II : Solitary wave propagation over an irregular 3-D shallow shelf

	Adaptive mesh simulation	Uniform mesh simulation
Simulation time	30 s	30 s
Number of blocks	128	128
Number of cells	7 500-25 000	33 000
Re-meshing time step	0.25 s	not applicable
Time order integration	2	2
Space order integration	2	2
CFL	0.5	0.5

Table: Numerical parameters

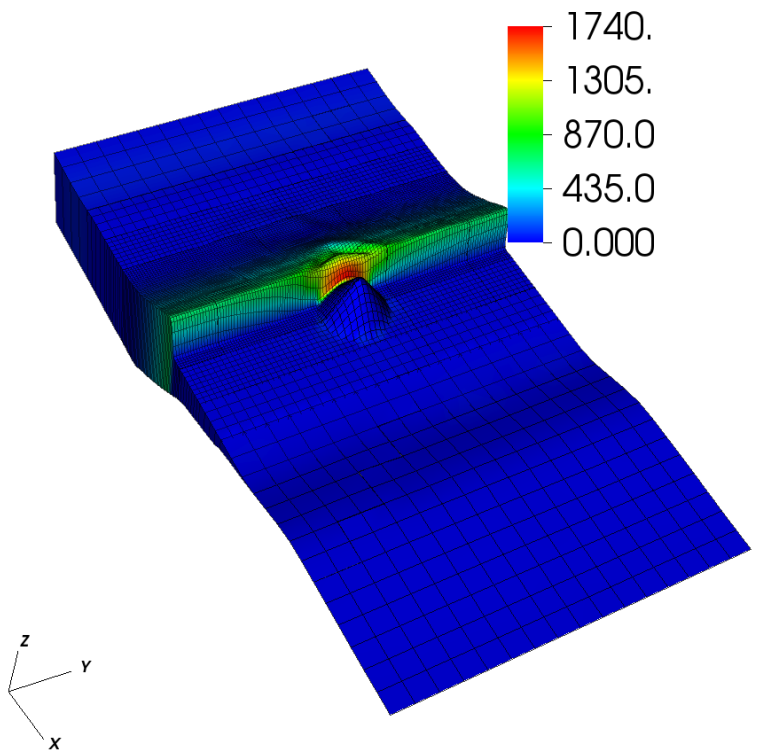
Test case II : Solitary wave propagation over an irregular 3-d shallow shelf



(a) t=0.5s

Figure: Numerical water height (coloration is issue from the kinetic energy)

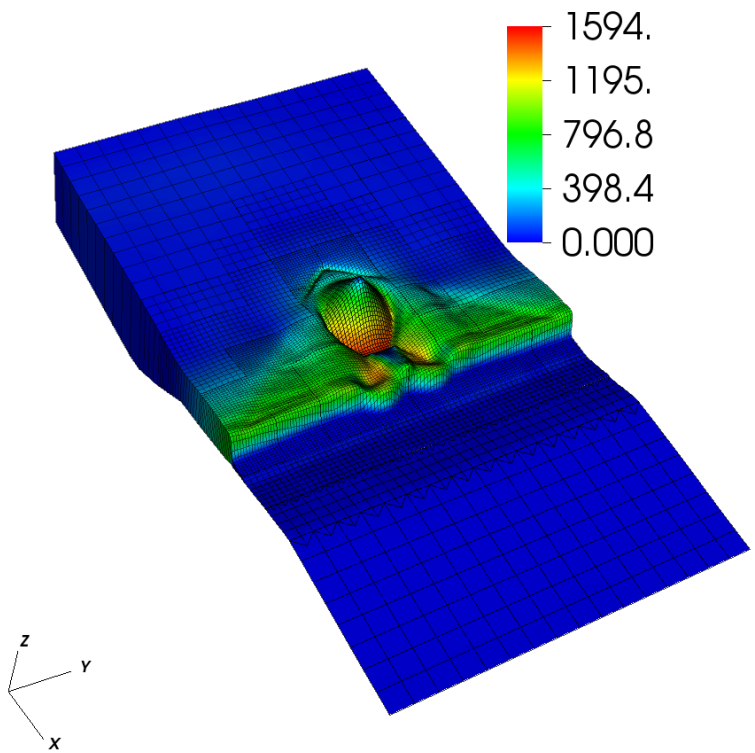
Test case II : Solitary wave propagation over an irregular 3-d shallow shelf



(a) t=2.5s

Figure: Numerical water height (coloration is issue from the kinetic energy)

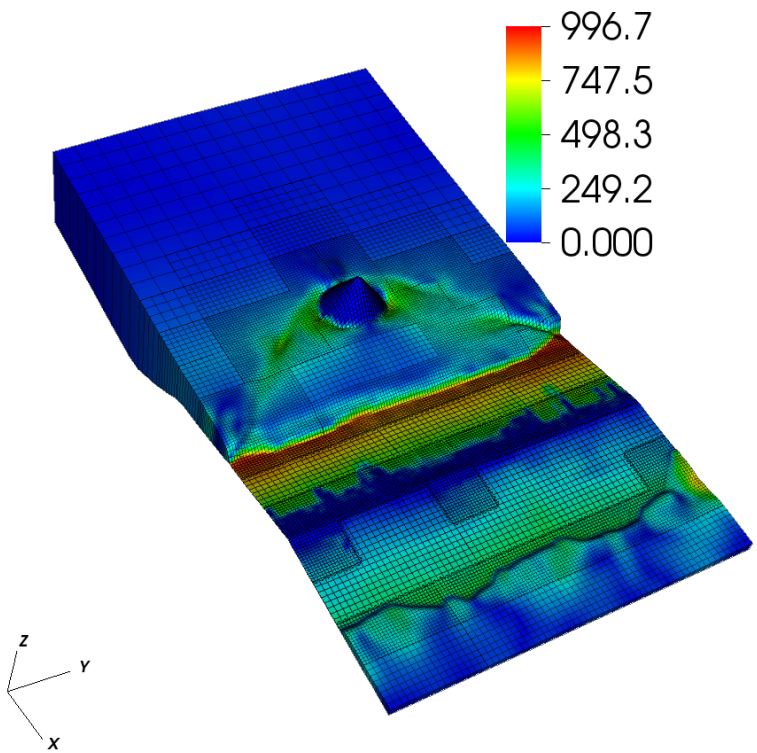
Test case II : Solitary wave propagation over an irregular 3-d shallow shelf



(a) t=5.75s

Figure: Numerical water height (coloration is issue from the kinetic energy)

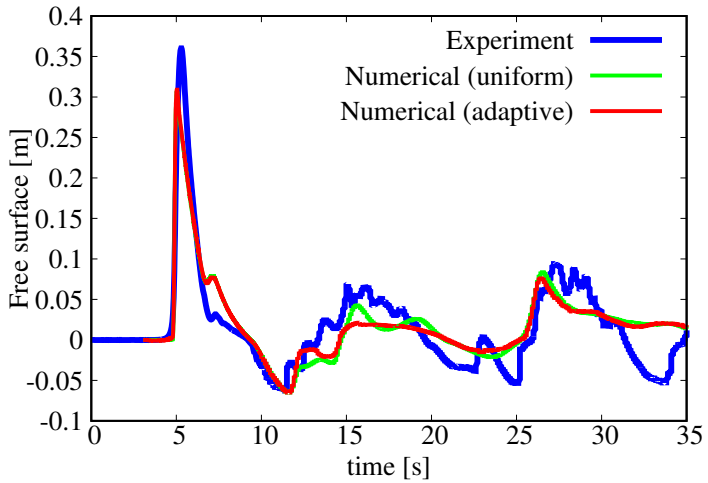
Test case II : Solitary wave propagation over an irregular 3-d shallow shelf



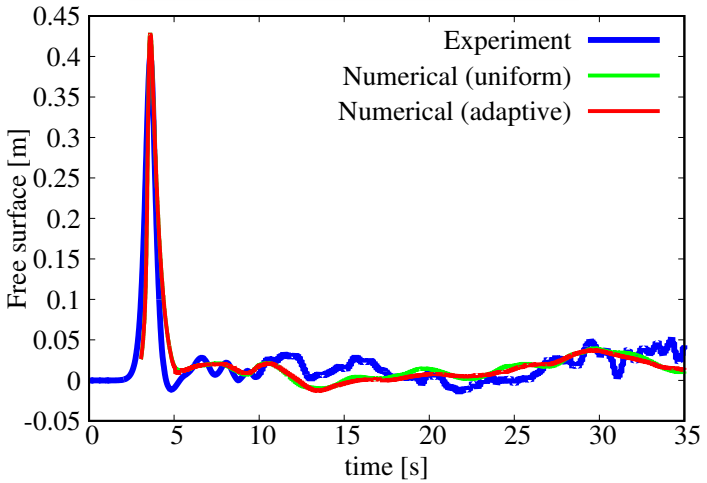
(a) t=23.75s

Figure: Numerical water height (coloration is issue from the kinetic energy)

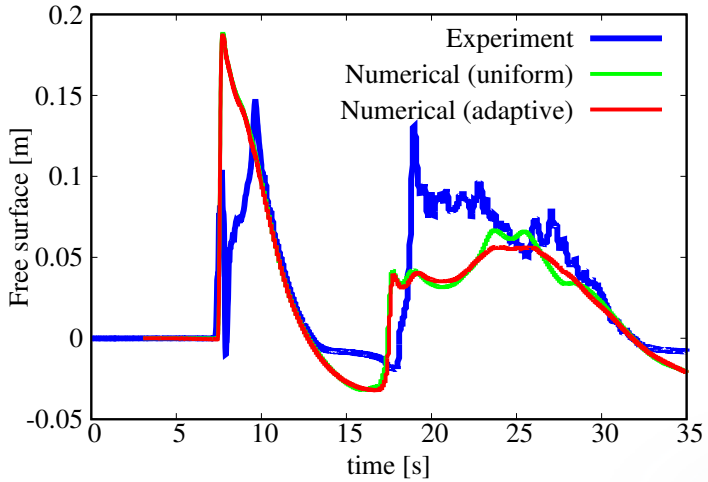
Test case II : Solitary wave propagation over an irregular 3-d shallow shelf



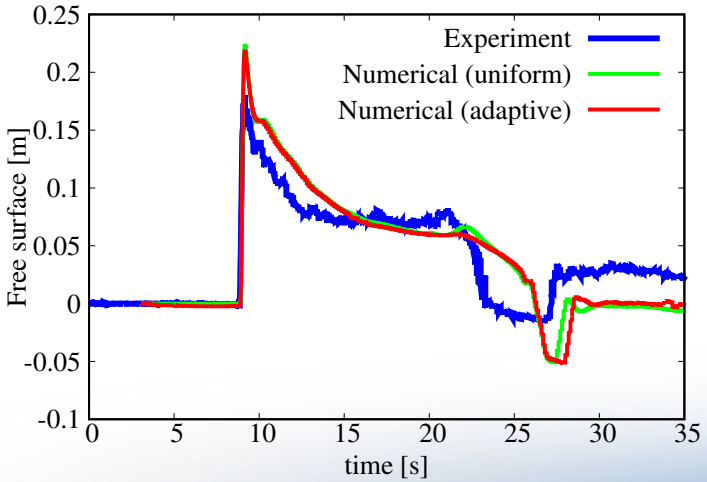
(a) WG2 results



(b) WG4 results

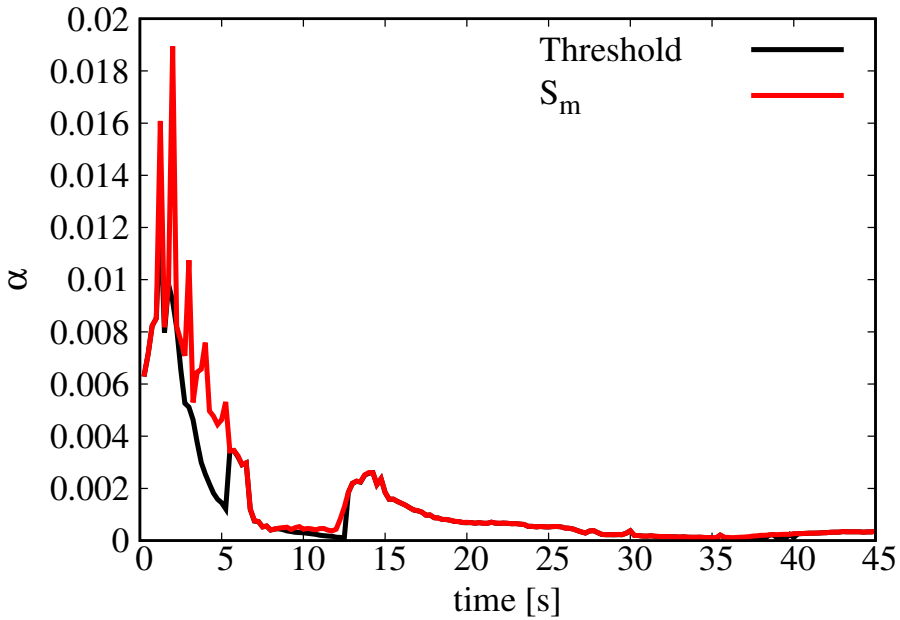


(c) WG6 results

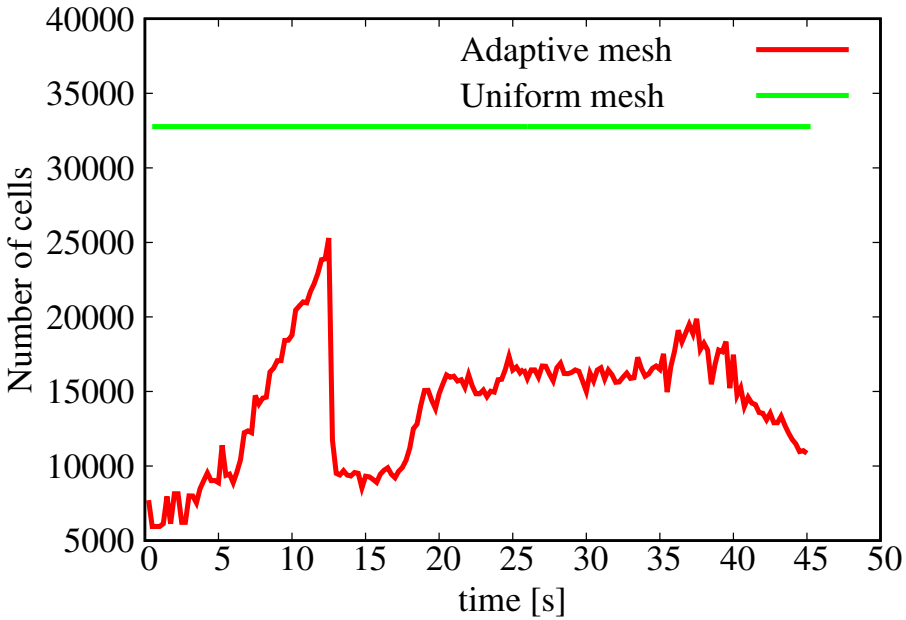


(d) WG7 results

Test case II : Solitary wave propagation over an irregular 3-D shallow shelf



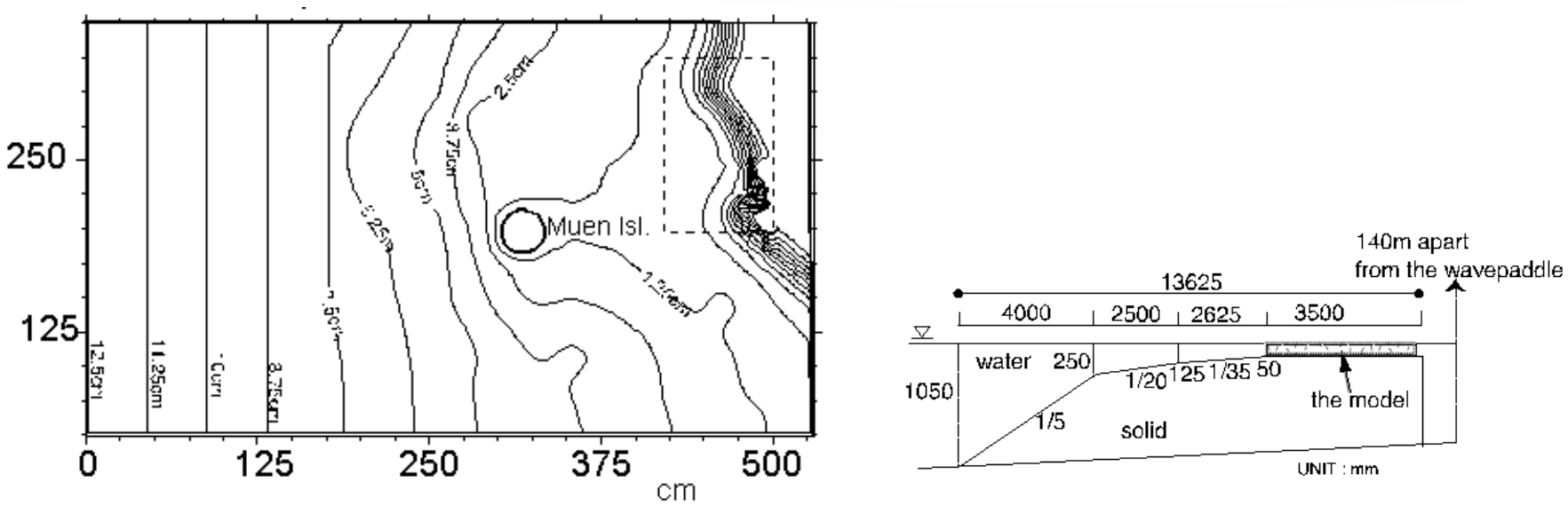
(a) Threshold



(b) Number of cells

Figure: Time evolution of the mesh refinement threshold and the number of cells : speed up the computation by 2.5 time

Test case III : Tsunami runup onto a complex three dimensional Monai-Walley beach



(a) Top view (b) Side view

Figure: Settings

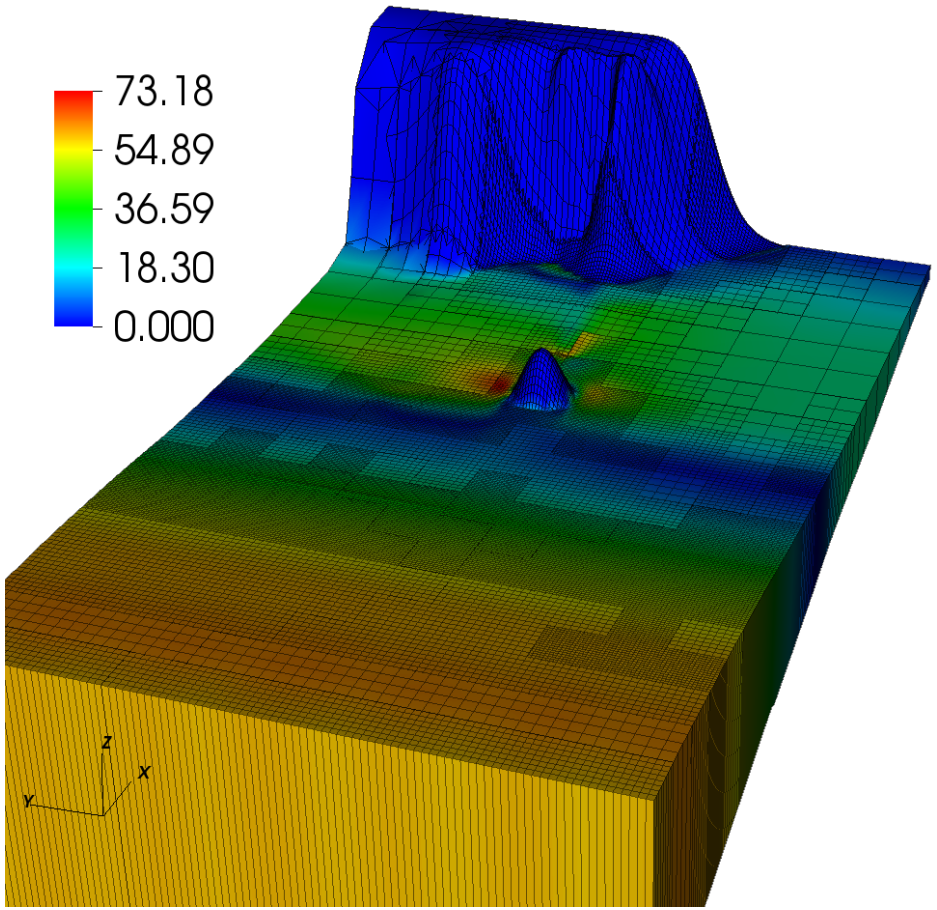
Kévin Pons, Mehmet Ersoy, Frédéric Golay, Richard Marcer. Adaptive mesh refinement method. Part 2 : Application to tsunamis propagation

Test case III : Tsunami runup onto a complex three dimensional Monai-Walley beach

	Adaptive mesh simulation	Uniform mesh simulation
Simulation time	30 s	30 s
Number of blocks	240	240
Number of cells	8 000-40 000	62 000
Re-meshing time step	0.25 s	not applicable
Time order integration	2	2
Space order integration	1	1
CFL	0.5	0.5

Table: Numerical parameters

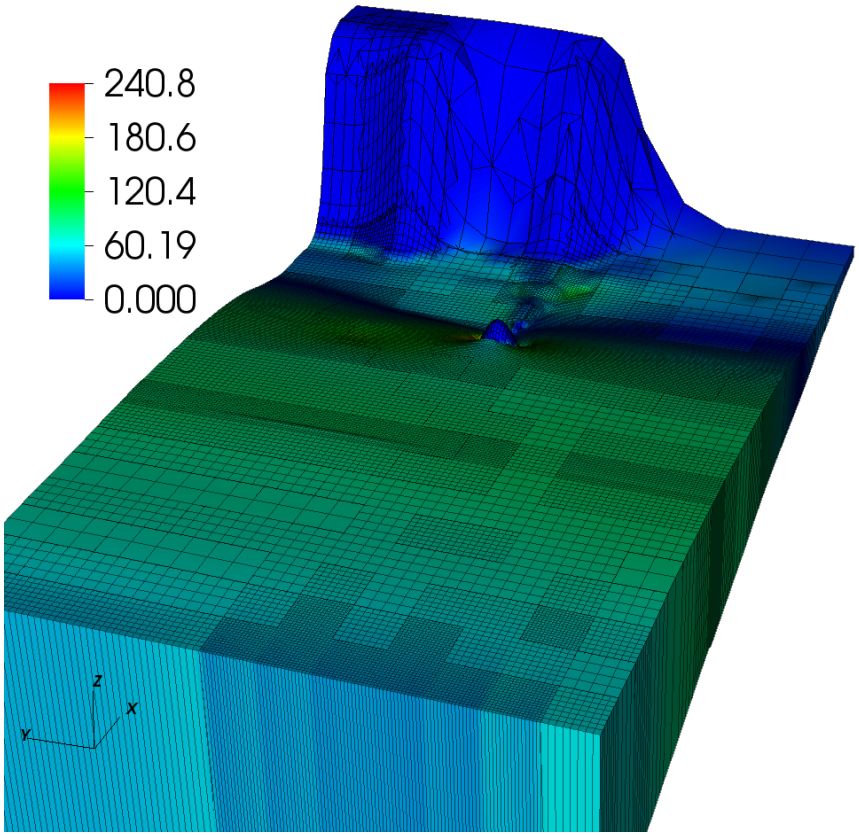
Test case III : Tsunami runup onto a complex three dimensional Monai-Walley beach



(a) $t = 11.25$ s

Figure: Numerical water height (coloration is issue from the kinetic energy)

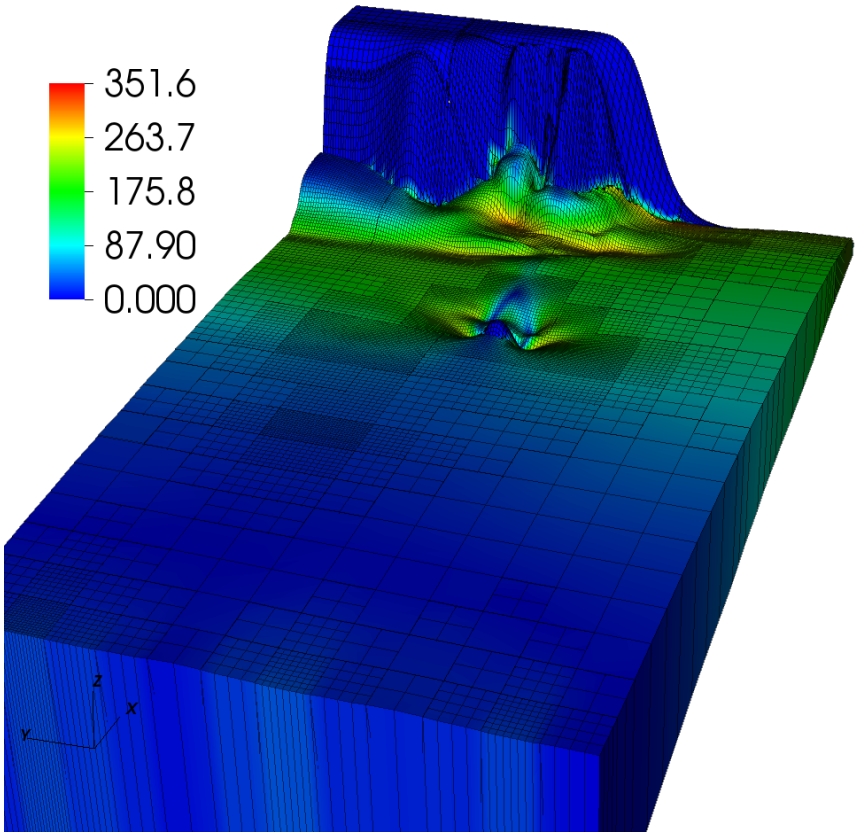
Test case III : Tsunami runup onto a complex three dimensional Monai-Walley beach



(a) $t = 13.25$ s

Figure: Numerical water height (coloration is issue from the kinetic energy)

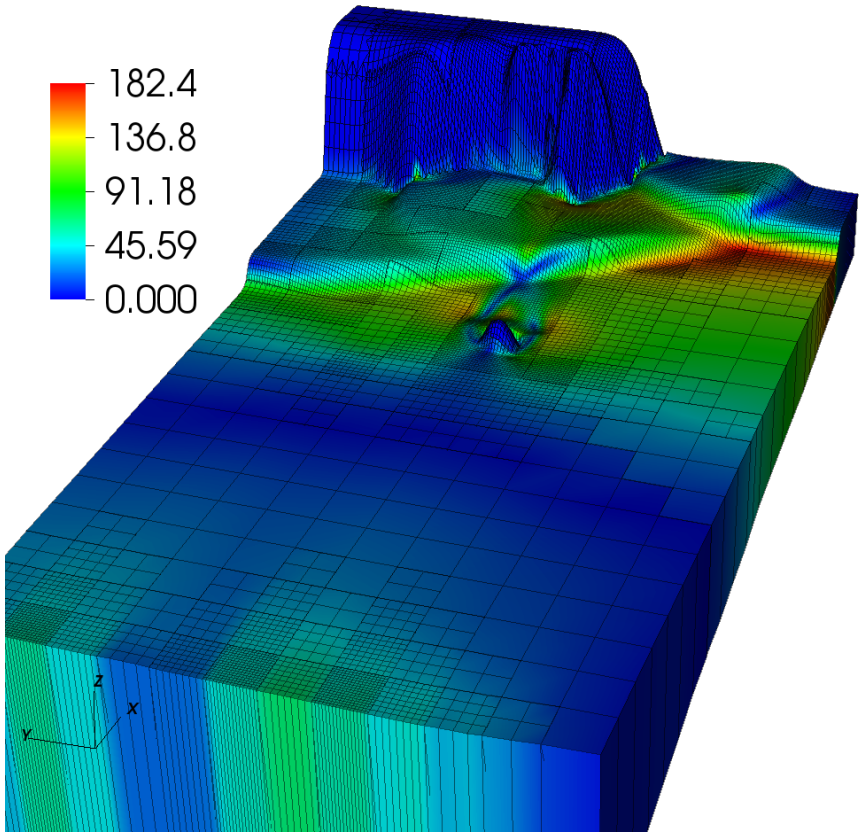
Test case III : Tsunami runup onto a complex three dimensional Monai-Walley beach



(a) $t = 16$ s

Figure: Numerical water height (coloration is issue from the kinetic energy)

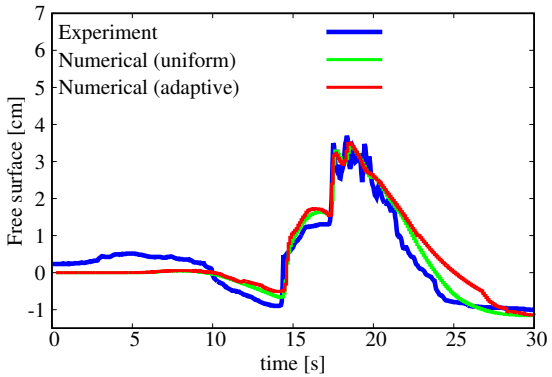
Test case III : Tsunami runup onto a complex three dimensional Monai-Walley beach



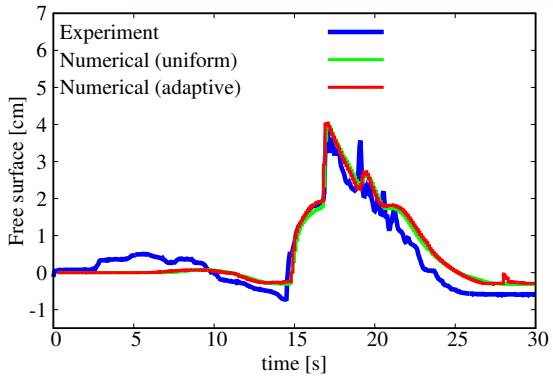
(a) $t = 17.5$ s

Figure: Numerical water height (coloration is issue from the kinetic energy)

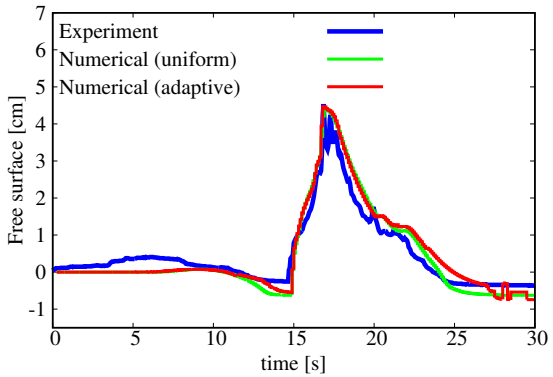
Test case III : Tsunami runup onto a complex three dimensional Monai-Walley beach



(a) Gauge 1



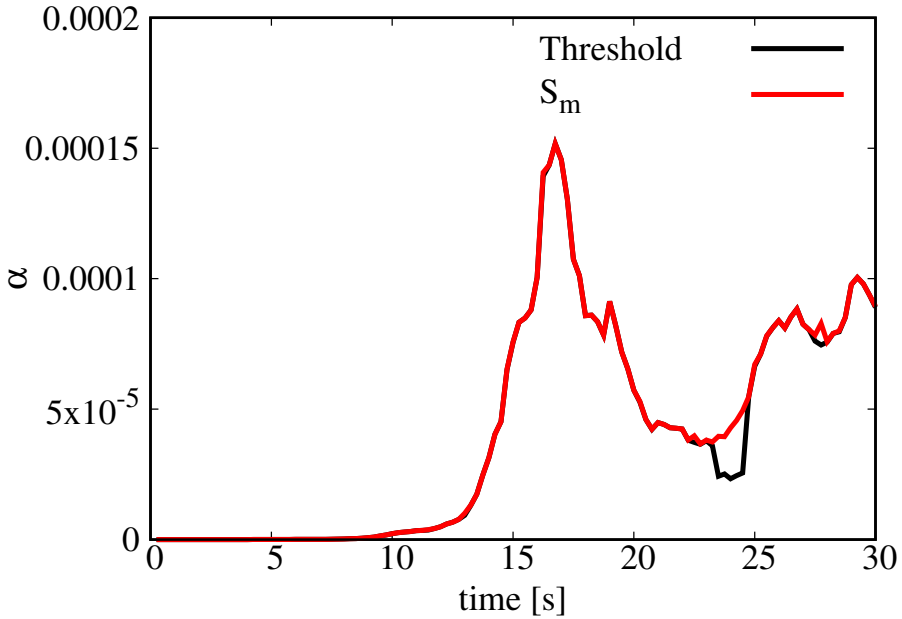
(b) Gauge 2



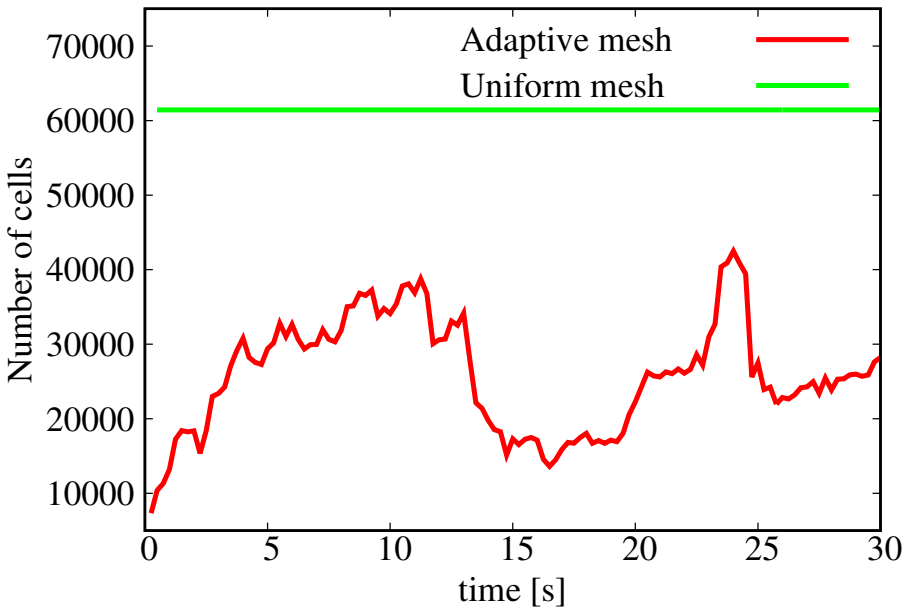
(c) Gauge 3

Figure: Free surface results at different positions : experimental data versus numerical simulation with and without mesh adaptivity

Test case III : Tsunami runup onto a complex three dimensional Monai-Walley beach



(a) Threshold



(b) Number of cells

Figure: Time evolution of the mesh refinement threshold and the number of cells : speed up the computation by 3 time

Thank you

for your

attention

Example

Let us consider the transport equation :

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

Back to slide

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 :

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

Back to slide

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 $\psi(w) = w^2$.

Back to slide

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 $\psi(w) = w^2$.

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

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

▸ Back to slide

Time restriction

- Explicit adaptive schemes : time consuming due to the restriction

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

Time restriction, local time stepping approach

- Explicit adaptive schemes : time consuming due to the restriction

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

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

Muller S., Stiriba Y., *Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping.* SIAM J. Sci. Comput., 30(3) :493 ?531, 2007.

Time restriction, local time stepping approach & Aims

- Explicit adaptive schemes : time consuming due to the restriction

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

- 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.

Muller S., Stiriba Y., *Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping*. SIAM J. Sci. Comput., 30(3) :493 ?531, 2007.

Illustration

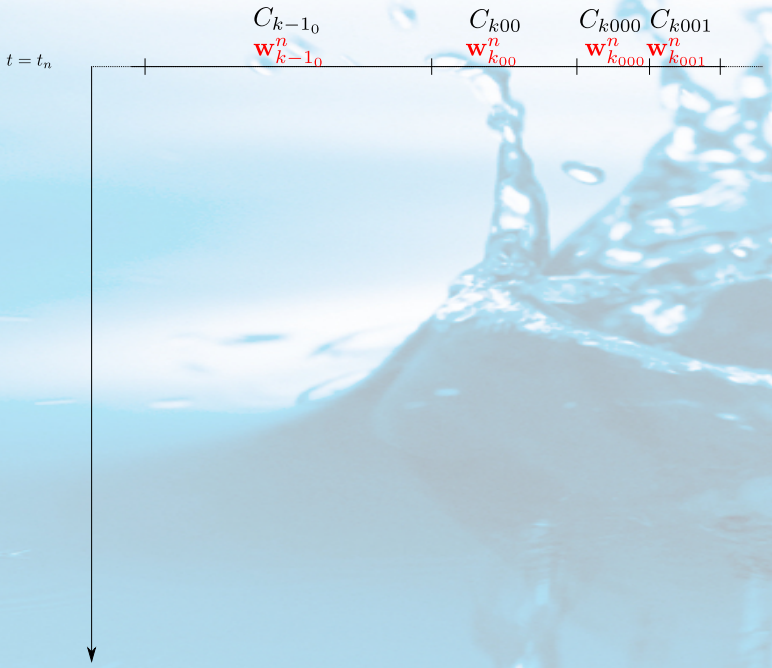
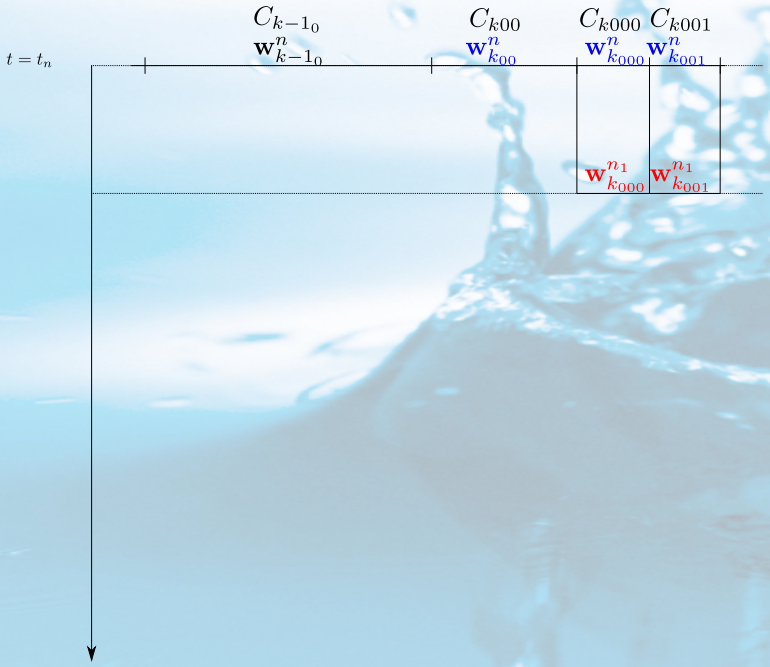


Figure: $t = t_n$

with

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

Illustration



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

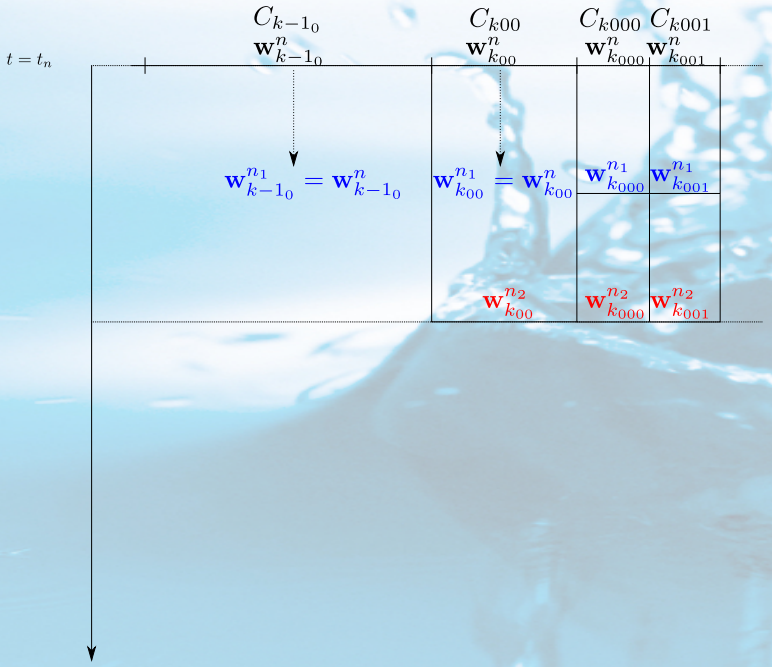
$$\boldsymbol{w}^{n_1}_{k_{001}} = \boldsymbol{w}^n_{k_{001}} - \frac{\delta t_n}{h_{k_{001}}} \delta \boldsymbol{F}^n_{k_{000},k_{001},k+1_b}$$

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

with

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

Illustration



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

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

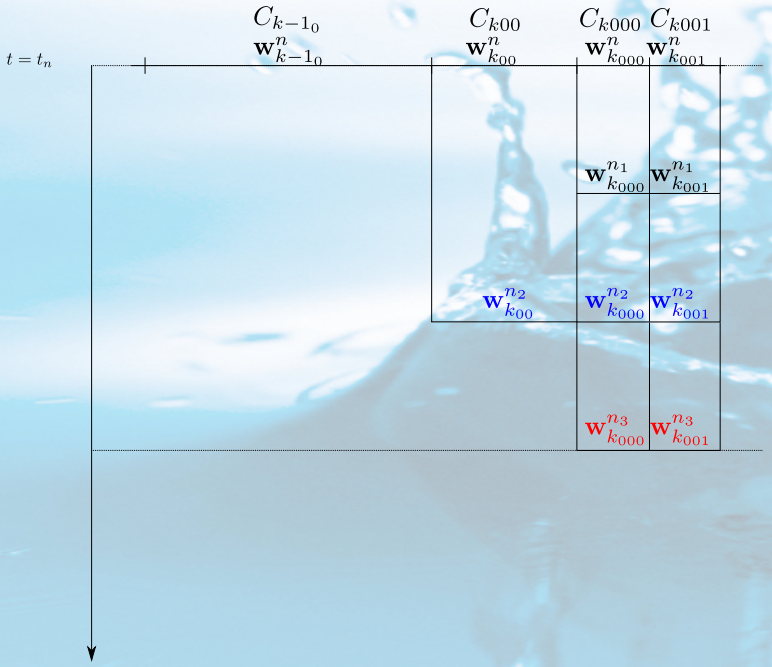
$$\boldsymbol{w}^{n_2}_{k_{001}} = \boldsymbol{w}^{n_1}_{k_{001}} - \frac{\delta t_n}{h_{k_{001}}} \delta \boldsymbol{F}^{n_1}_{k_{000},k_{001},k+1_b}$$

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

with

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

Illustration



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

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

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

with

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

Illustration

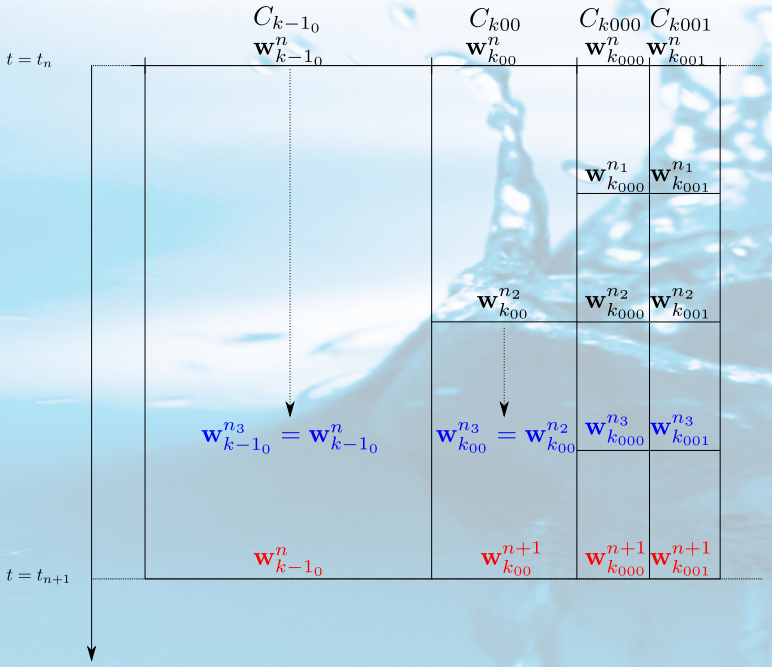


FIGURE: $t_{n+1} = t_n + 4\delta t_n$

$$\boldsymbol{w}^{n+1}_{k-1_0} = \boldsymbol{w}^{n_3}_{k-1_0} - \frac{\delta t_n}{h_{k-1_0}} \delta \boldsymbol{F}^{n_3}_{k-2_b, k-1_0, k00}$$

$$\boldsymbol{w}^{n+1}_{k00} = \boldsymbol{w}^{n_3}_{k00} - \frac{\delta t_n}{h_{k00}} \delta \boldsymbol{F}^{n_3}_{k-1_0, k00, k000}$$

$$\boldsymbol{w}^{n+1}_{k000} = \boldsymbol{w}^{n_3}_{k000} - \frac{\delta t_n}{h_{k000}} \delta \boldsymbol{F}^{n_3}_{k00, k000, k001}$$

$$\boldsymbol{w}^{n+1}_{k001} = \boldsymbol{w}^{n_3}_{k001} - \frac{\delta t_n}{h_{k001}} \delta \boldsymbol{F}^{n_3}_{k000, k001, k+1_b}$$

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

Shu and Osher test case

Mesh coarsening parameter α : 0.001 ,

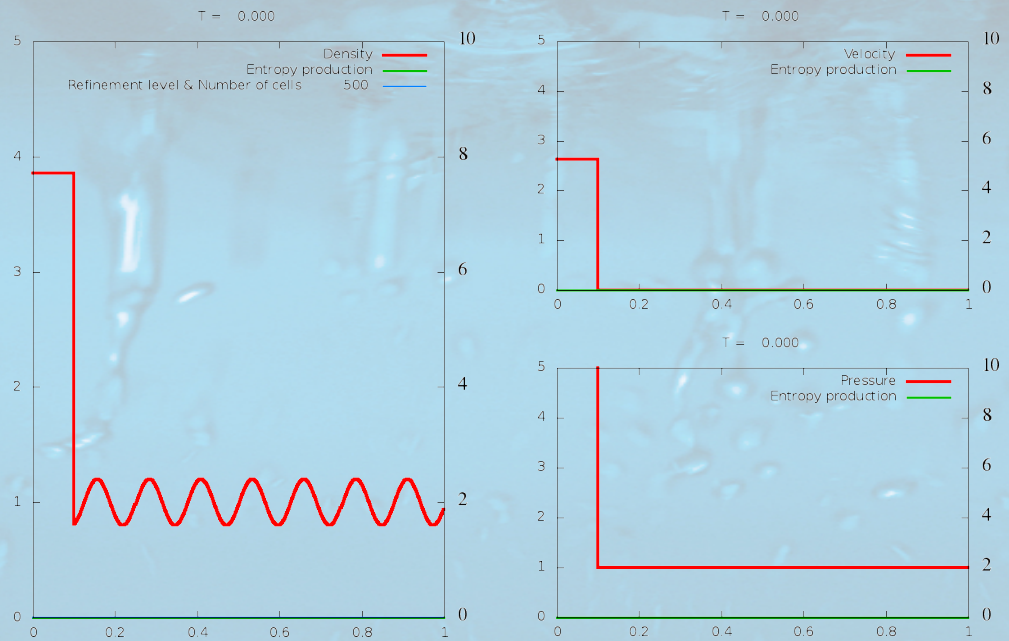
Mesh refinement parameter $\bar{S}$: $\dfrac{1}{|\Omega|}\sum_{k_b} S^n_{k_b}$

CFL : 0.219,

Simulation time (s) : 0.18,

Initial number of cells : 500,

Maximum level of mesh refinement : $L_{\max} = 4$.



Efficiency of the local time stepping method

	$\mathcal{P}$	$\|\rho - \rho_{ref}\|_{l_x^1}$	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

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

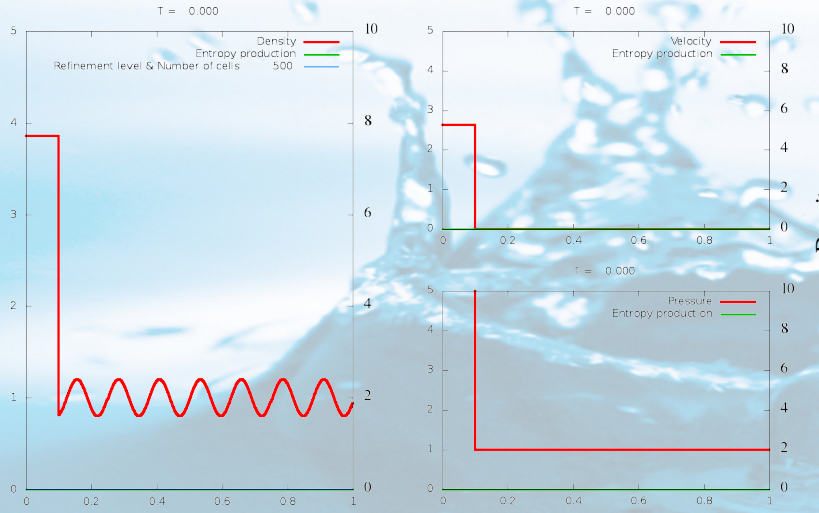
TABLE: Shu and Osher test case : comparison of numerical schemes of order 1 and 2

M. Ersoy, F. Golay, L. Yushchenko. *Adaptive multi-scale scheme based on numerical entropy production for conservation laws.* CEJM, Central European Journal of Mathematics, 11(8), pp 1392-1415, 2013.

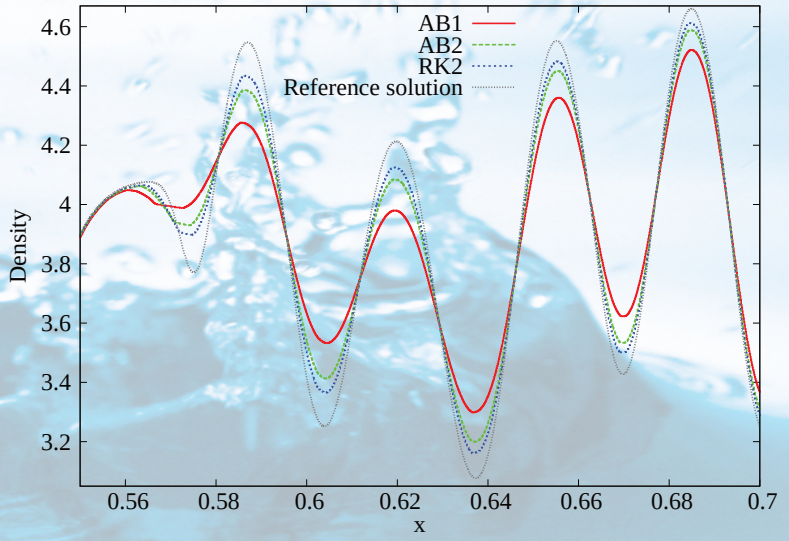
	$\mathcal{P}$	$\|\rho - \rho_{ref}\|_{l^1_x}$	cpu-time	$N_{L_{\max}}$	maximum number of cells
AB1	0.288	$4.74\,10^{-2}$	181	1574	2308
AB1M	0.288	$4.80\,10^{-2}$	120	1572	2314
AB2	0.287	$2.75\,10^{-2}$	170	1391	2023
AB2M	0.286	$2.74\,10^{-2}$	108	1357	1994
RK2	0.285	$2.08\,10^{-2}$	299	1375	2005

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

M. Ersoy, F. Golay, L. Yushchenko. *Adaptive multi-scale scheme based on numerical entropy production for conservation laws.* CEJM, Central European Journal of Mathematics, 11(8), pp 1392-1415, 2013.



(a) Density and numerical density of entropy production.



(b) Zoom on oscillating region.

Figure: Shu and Osher test case.

Back to slide

TWO AND THREE DIMENSIONAL CASE : BB-AMR

- Main difficulty : mesh and data structure.
  For fast computation, the following are required
  - parallel treatment
  - hierarchical grids

TWO AND THREE DIMENSIONAL CASE : BB-AMR

- Main difficulty : mesh and data structure.
  Some interesting issues :
  - 2D quad-tree [1],
  - 3D octree [2],
  - 2/3D anisotropic AMR [3].

Zhang, M., and W.M. Wu. 2011. *A two dimensional hydrodynamic and sediment transport model for dam break based on finite volume method with quadtree grid.* Applied Ocean Research 33 (4) : 297 – 308.

Losasso, F., F. Gibou, and R. Fedkiw. 2004. *Simulating Water and Smoke with an Octree Data Structure.* ACM Trans. Graph. 23 (3) : 457–462, 2004.

Hachem, E., S. Feghali, R. Codina, and T. Coupez. *Immersed stress method for fluid structure interaction using anisotropic mesh adaptation.* International Journal for Numerical Methods in Engineering 94 (9) : 805–825, 2013.

TWO AND THREE DIMENSIONAL CASE : BB-AMR

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

TWO AND THREE DIMENSIONAL CASE : BB-AMR

- Main difficulty : mesh and data structure.
- The strategy adopted :
  1. 1 domain= 1 block=1 cpu : "failure" $\rightarrow$ synchronization depends on the finest domain

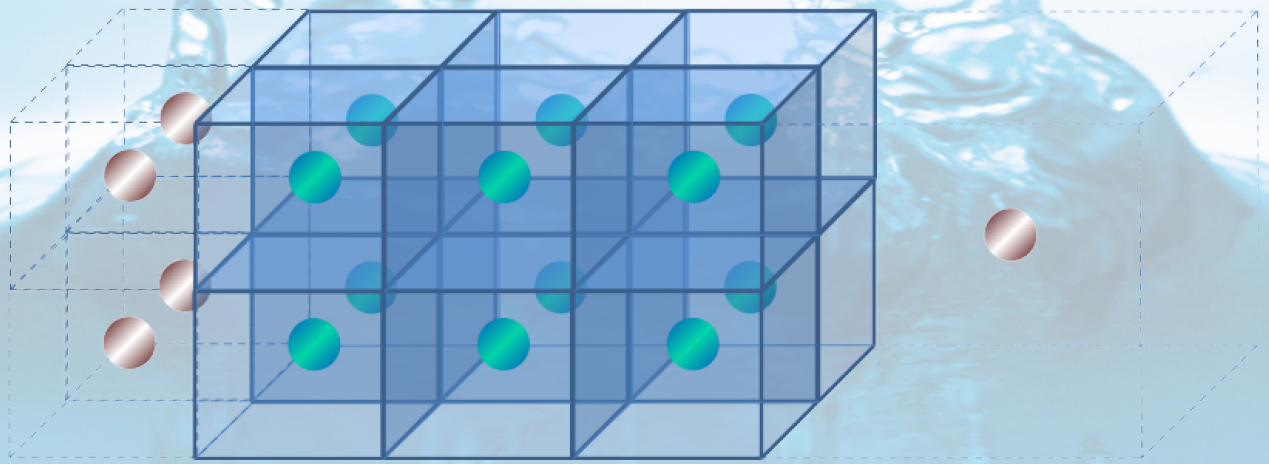
- Main difficulty : mesh and data structure.
- The strategy adopted :
  1. 1 domain= 1 block=1 cpu : "failure" → synchronization depends on the finest domain
  2. 1 domain= n × blocks = 1cpu : "good compromise" → each domain has almost the same number number of cells → "better" synchronization

TWO AND THREE DIMENSIONAL CASE : BB-AMR

- Main difficulty : mesh and data structure.
- The strategy adopted :
  1. 1 domain= 1 block=1 cpu : "failure" $\rightarrow$ synchronization depends on the finest domain
  2. 1 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 . . .

TWO AND THREE DIMENSIONAL CASE : BB-AMR

- Main difficulty : mesh and data structure.
- The strategy adopted :
- Management of domain's interfaces, projection step, . . .

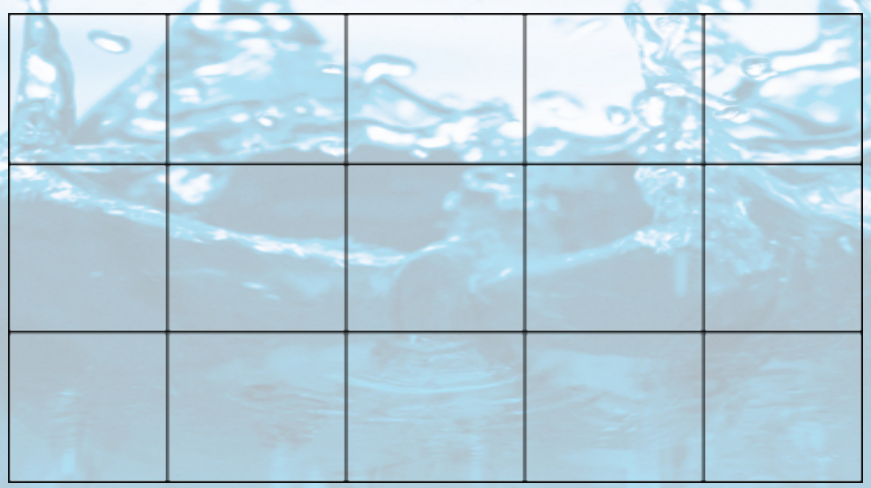


How it works ?

- each domain has almost the same number of cells

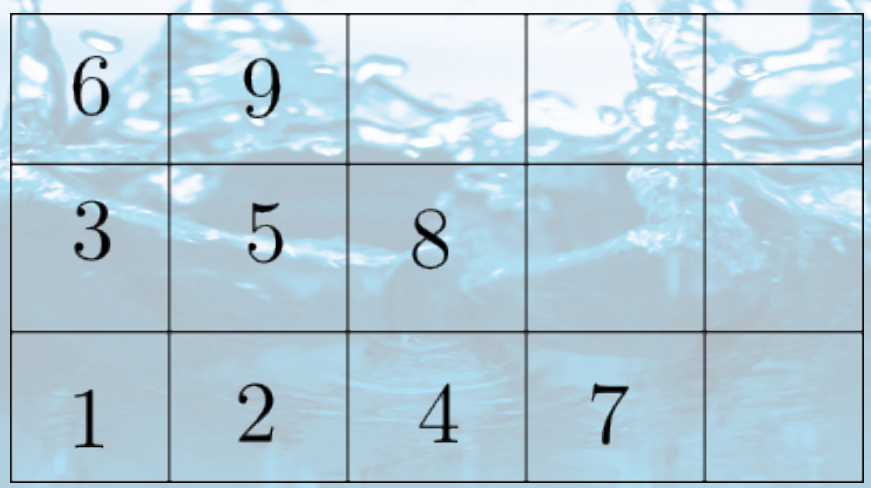
How it works ?

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



How it works ?

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



How it works ?

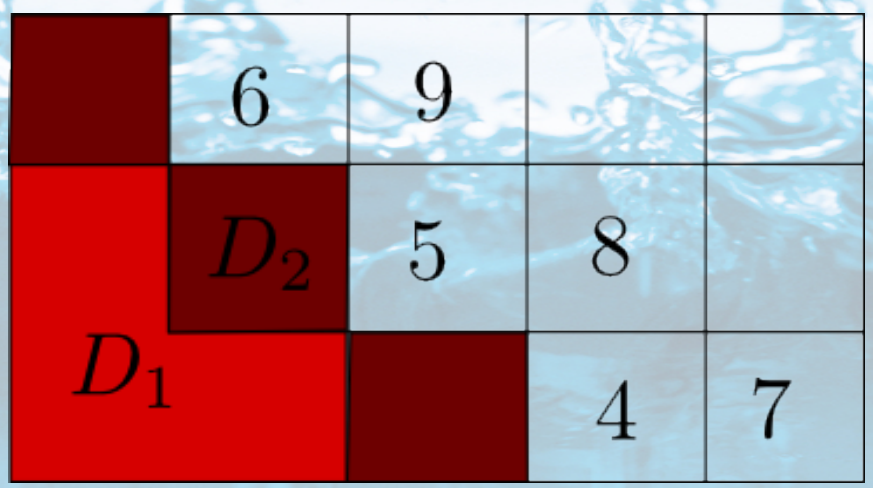
- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering

How it works ?

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

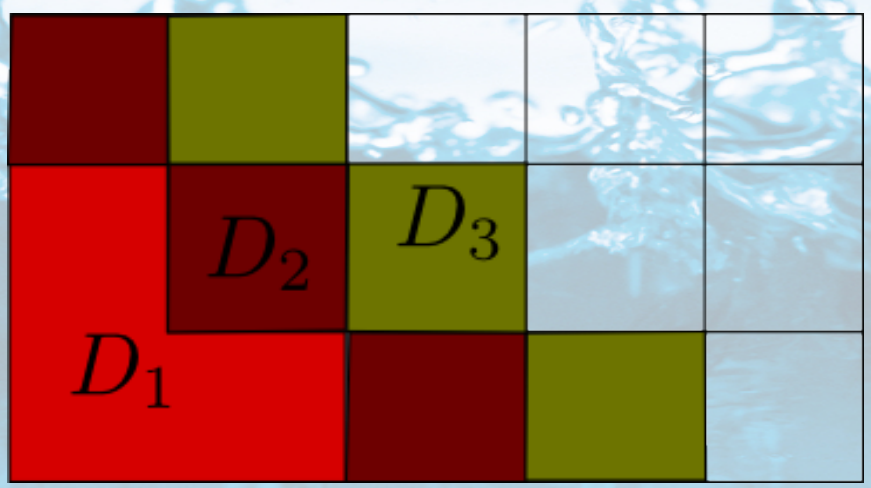
How it works ?

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



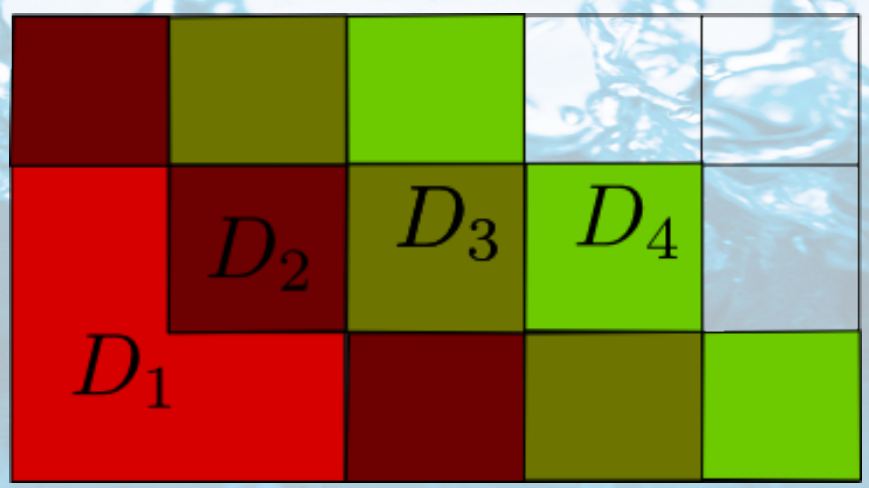
How it works ?

- 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

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 . . .

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 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 kept constant on a time interval
  - AMR time-step computed through the smallest block and not the smallest cell $T_{n+1} - T_n = \Delta T_{\text{AMR}}$ is given by the CFL

$$\Delta T_{\text{AMR}} \leqslant \beta \frac{\min_k h_{\text{block}_k}}{\max_k \|\boldsymbol{u}_{\text{block}_k}\|}, \ 0 < \beta \leqslant 1.$$

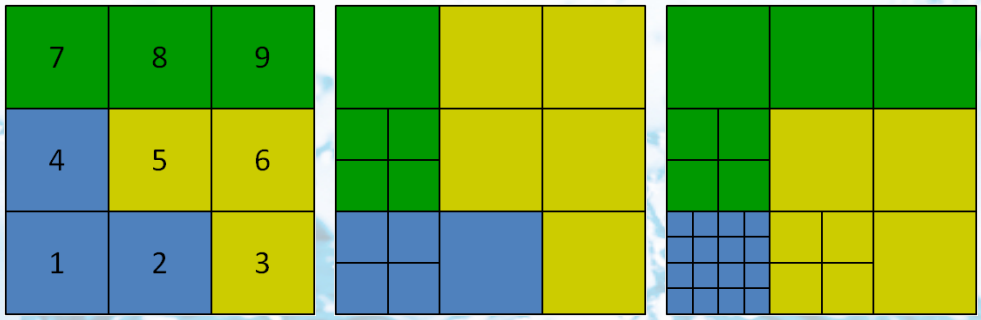
DOMAIN= N × BLOCKS = 1CPU

How it works ?

- each domain has almost the same number of cells
- domain are defined using Cuthill-McKee numbering
- more sophisticated numbering exists ...
- re-numbering and re-meshing being expensive
  - the mesh should 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.

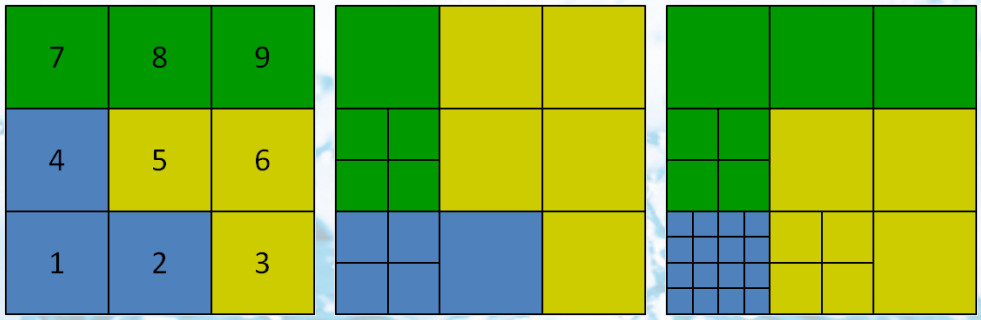


(a) AMR T_0 (b) AMR T_1 (c) AMR T_2

Back to slide

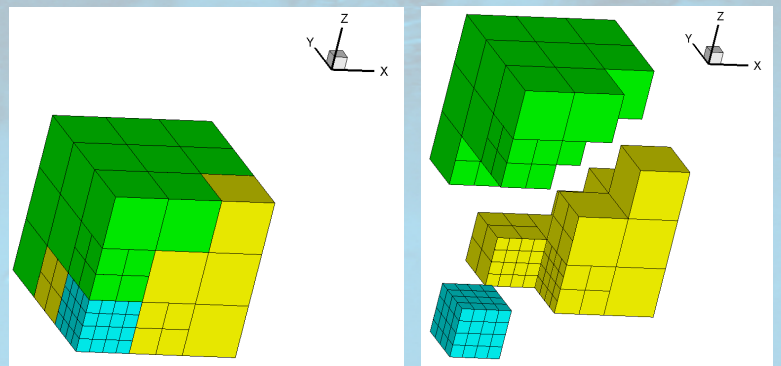
Examples :

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



(f) AMR T_0 (g) AMR T_1 (h) AMR T_2

- A three dimensional example of BB-AMR with 3 domains and 27 blocks.



(i) Block-based mesh (j) Domain decomposition

Back to slide