Adaptive mesh refinement method. Part 2: Application to tsunamis propagation

Kévin Pons, Richard Marcer, Mehmet Ersoy, Frederic Golay, Richard Marcer Principia

Adaptive mesh refinement method. Part 2: Application to tsunamis propagation.

Kévin Pons, Mehmet Ersoy, Frédéric Golay and Richard Marcer

Abstract Numerical simulations of multi dimensional large scale fluid-flows such as tsunamis, are still nowadays a challenging and a difficult problem. To this purpose, a parallel finite volume scheme on adaptive unstructured meshes for multi dimensional Saint-Venant system is presented. The adaptive mesh refinement (AMR) method is based on a block-based decomposition (called BB-AMR) which allows quick meshing and easy parallelisation. The BB-AMR method is equipped with an Automatic Mesh Refinement Threshold (AMRT) which allows to improve the overall accuracy, efficiency and save the computational time of the AMR method. The AMRT method is constructed from the decreasing rearrangement function of the mesh refinement criterion which allows to catch relevant scales to refine. In practice, we are able to reach a good balance between the accuracy and the computational cost with an automatic threshold instead of a "user"-fixed one. This approach makes the AMR method robust and almost parameterless. The numerical method is validated successfully for several tsunamis propagation test cases.

Kévin Pons
Principia S.A.S., Zone Athélia 1, 215 voie Ariane, 13705 La Ciotat cedex, France, e-mail: Kevin.Pons@principia.fr,
Université de Toulon, IMATH EA 2134, 83957 La Garde, France, e-mail: Kevin.Pons@univ-tln.fr

Mehmet Ersoy
Université de Toulon, IMATH EA 2134, 83957 La Garde, France, e-mail: Mehmet.Ersoy@univ-tln.fr

Frédéric Golay
Université de Toulon, IMATH EA 2134, 83957 La Garde, France, e-mail: Frederic.Golay@univ-tln.fr

Richard Marcer
Principia S.A.S., Zone Athélia 1, 215 voie Ariane, 13705 La Ciotat cedex, France, e-mail: Richard.Marcer@principia.fr

1 Introduction

Tsunamis are generally referred to any impulsed gravity wave. It can be generated by many sources. The most common tsunamis are a consequence of earthquakes and landslides. Such events can displace a very large volume of water from its equilibrium. The displaced water mass, under the gravity action, attempts to come back to its equilibrium generating gravity waves. Depending of the tsunami source, the order of magnitude of a tsunami wave length is generally larger than ten kilometers. Since this horizontal length is much larger than the oceans depth, vertically integrated model are generally used for tsunamis propagation. One of the most simple and one of the most relevant model among depth-averaged equations is the Saint-Venant model (also called Non Linear Shallow Water – NLSW), see for instance [21, 22, 32]. Nevertheless, due to the non-dispersive behaviour of this set of equations, the numerical results for *some tsunamis* problem may lead to under forecasting, for instance, premature breaking-waves. To be more accurate in that case, the use of dispersive and fully nonlinear equations are necessary, such as Green-Naghdi equations ([38, 33, 10]) or some non-hydrostatic models ([46, 26]). However, almost all dispersive models are based on the hyperbolic structure of the Saint-Venant equations. Therefore, in this work, we restrict ourselves to the Saint-Venant model and the generalisation to dispersive models can be easily adapted.

Solving accurately this model over very large ranges in spatial scale with high resolution inexorably leads to heavy computational time [31, 32, 1]. In principle, Adaptive Mesh Refinement (AMR) [6] or Adaptive Moving Mesh (AMM) [2] methods allow to solve in a reasonable CPU time these equations adjusting the computational effort locally to maintain a quiet uniform level of accuracy. The first one relies on macro cells which can be refined (and then possibly coarsened) while the other method allows all mesh nodes to move to generate big or small cells (with a fixed nodes number for all the simulation). The zones where the mesh needs to be moved or refined is determined thanks to a mesh refinement criterion. Depending on the application, equations and numerical methods, a variety of different criteria might be used based on an error estimation procedure or a feature detection technique. Basically, the zones where a given threshold is exceeded are moved or refined. In particular, for AMR methods, the more the threshold is small, the more accurate is the result to the expense of the computational time. We focus in the rest of the paper to AMR method.

- The overall efficiency and accuracy of the AMR method can be improved using an Automatic Mesh Refinement Threshold (AMRT) [29] based on the decreasing rearrangement of the mesh refinement criterion function. The decreasing rearrangement provides a description of the criterion in terms of local maxima which are sorted canonically from the smallest to the largest. Thus, this method allows in practice to set the threshold automatically small enough to catch the most relevant scales to refine with a simple mesh refinement criterion. The method does not require accurate error estimate generally hard to obtain and provided case by case. The method needs only an error indicator as a mesh refinement criterion

to provide relevant numerical results having a good balance between the accuracy and the computational cost of the numerical method. Moreover, the AMRT method is almost *parameterless* while the most of the AMR methods are based on a "user calibrated" threshold, case-dependent, which requires time and effort to set it correctly (see for instance [11]). One can easily adapt the *parameterless* method to AMM methods.

- Next, the overall performance of the method can be improved in a parallel framework and more specifically with a Block-Based AMR method [16]. Each block corresponds to the initial unstructured mesh and according to the mesh refinement procedure each block can be sub-divided. The cells of each block are redistributed in a fixed number of domains. The number of domain being fixed, each domain are loaded in a given MPI process. The BB-AMR technique provides an efficient control of the shared memory leading to well-balanced computational time between cores by domain-like decomposition. Nevertheless, data locality is critical to obtain good performance since the memory access times are not uniform and may become expensive. Consequently, the re-meshing step has to be carefully managed. Taking advantage of such a block-based structure (as in domain decomposition), we can define two different time-step: the first one is based on the (micro-)CFL δt_n condition (i.e computed through the finest cells) which ensures the stability of the numerical scheme. The second one is a (macro-)CFL $\Delta t_n \geqslant \delta t_n$ condition based on the finest block and it corresponds to the re-meshing time step. Thus, the global stability of the scheme is preserved for any time and the re-meshing cost is minimised.

In this paper, we present the AMRT method and we focus on its efficiency for the two-dimensional Saint-Venant equations in the context of tsunami propagation. The method is validated through several test cases.

The paper is organized as follows: the first section of the paper is devoted to the presentation of the tsunami model (Sect. 2.1) and its finite volume approximation (Sect. 2.2). The second section is dedicated to the presentation of the multi dimensional finite volume solver on unstructured meshes with AMRT (Sect. 3.1) and the Block-Based Adaptive Mesh Refinement technique (Sect. 3.2). The computational results are compared to experimental data in Sect. 4. The first test case (Sect. 4.1) concerns the flume experiments conducted at Oregon State University [36, 35]. These experiments involve the propagation, run-up and reflection of high amplitude solitary waves on a reef. The second test case (Sect. 4.2) is the propagation of a solitary wave over a complex three dimensional shallow shelf [25]. In this experiment, the propagation, run-up, drying and flooding phenomena are involved. The last test case (Sect. 4.3) concerns the Monai-Walley tsunami run-up onto a complex three dimensional beach [23].

Unless otherwise indicated, bold characters are used for vector notation and $\mathbf{w}^T$ denotes the transpose of the vector $\mathbf{w}$.

2 Equations, properties and finite volume approximation

In this section, we present the Saint-Venant equations which are the standard and one of most the simple governing model used for tsunami propagation. We first recall its mathematical and physical properties which are useful for numerical purposes, such as the entropy inequality. Then, we present the multi dimensional finite volume approximation.

2.1 Saint-Venant equations and entropy inequality

The two dimensional non linear Saint-Venant system is

$$\begin{aligned} \partial_t h + \operatorname{div}(h\boldsymbol{u}) &= 0 \\ \partial_t (h\boldsymbol{u}) + \operatorname{div}\left(h\boldsymbol{u}\otimes\boldsymbol{u} + \frac{g}{2}h^2 I\right) &= -gh\nabla Z \end{aligned} \tag{1}$$

where the unknowns $h(\boldsymbol{x},t)$ and $\boldsymbol{u}(\boldsymbol{x},t) = (u_1(\boldsymbol{x},t), u_2(\boldsymbol{x},t))$ are respectively the height of the water and the depth-averaged velocity of the water at a space-time point $(\boldsymbol{x},t)$, $\boldsymbol{x} = (x_1,x_2) \in \mathbb{R}^2$ is the space coordinate, $t > 0$ is the time, g is the gravitational constant set to 9.81m/s^2, $Z(\boldsymbol{x})$ is the bottom surface elevation and I is the identity matrix.

The three equations above express respectively, the conservation laws of mass and momentum driven by the fluxes

$$\boldsymbol{f}_1(\boldsymbol{w}) = \begin{pmatrix} hu_1 \\ hu_1^2 + \frac{g}{2}h^2 \\ hu_1u_2 \end{pmatrix} \text{ and } \boldsymbol{f}_2(\boldsymbol{w}) = \begin{pmatrix} hu_2 \\ hu_1u_2 \\ hu_2^2 + \frac{g}{2}h^2 \end{pmatrix}$$

and forced through the source $\boldsymbol{G}(\boldsymbol{x},\boldsymbol{w}) = (0, -gh\partial_{x_1}Z(\boldsymbol{x}), -gh\partial_{x_2}Z(\boldsymbol{x}))^T$ where the conservative variable $\boldsymbol{w} = (h, hu_1, hu_2)^T$. Noting $\boldsymbol{f}(\boldsymbol{w}) = (\boldsymbol{f}_1(\boldsymbol{w}), \boldsymbol{f}_2(\boldsymbol{w}))^T$, System (1) reads in its vectorial form

$$\partial_t \boldsymbol{w} + \operatorname{div}\boldsymbol{f}(\boldsymbol{w}) = \boldsymbol{G}(\boldsymbol{x},\boldsymbol{w}) \ . \tag{2}$$

System (2) is strictly hyperbolic on the set $\{h(\boldsymbol{x},t) > 0\}$. It is well-known that solving equations (2) with high accuracy is a challenging problem since solutions can and will breakdown at a finite time, even if the initial data are smooth, and develop complex structure (shock wave interactions). In such a situation, the uniqueness of the (weak) solution is lost and is recovered by completing the system (2) with an entropy inequality of the form:

$$\mathscr{S}(\boldsymbol{x},t) = \frac{\partial s(\boldsymbol{x},\boldsymbol{w})}{\partial t} + \operatorname{div}\boldsymbol{\psi}(s(\boldsymbol{x},\boldsymbol{w})) \leq 0 \ , \tag{3}$$

where $(s, \boldsymbol{\psi})$ stands for a convex entropy-entropy flux pair [37, 4]. The quantity $\mathscr{S}$ is called the entropy production. This inequality in equation (3) is used to select the physical relevant solution. Moreover, the entropy satisfies a conservation equation only in regions where the solution is smooth and an inequality when the solution develops shocks. In simple case, it can be proved that the missing term in (3) to make it an equality is a Dirac mass. Therefore, the entropy production can be seen as a "smoothness indicator" at the numerical level, see for instance [34, 11].

For the Saint-Venant system (2), a convex entropy-entropy flux pair is given by

$$s(\boldsymbol{x},\boldsymbol{w}) = \frac{h|\mathbf{u}|^2}{2} + \frac{gh^2}{2} + ghZ(\boldsymbol{x}), \quad \boldsymbol{\psi}(s(\boldsymbol{x},\boldsymbol{w})) = \left(s(\boldsymbol{x},\boldsymbol{w}) + \frac{gh^2}{2}\right)\mathbf{u}$$

where $|\mathbf{u}|^2 = u_1^2 + u_2^2$.

2.2 Finite volume approximation

The computational domain $\Omega \subset \mathbb{R}^d$, $d = 1$ or 2, is split into a set of control volumes, also referred as cells, $\Omega = \cup_k C_k$ of mesh size $|C_k|$. The source term is upwinded at the cell interface through the hydrostatic reconstruction [3]. As a consequence the obtained numerical scheme is well-balanced on fixed grid [3, 39] that is to say, it preserves the still water steady states

$$h + Z = \text{constant} \tag{4}$$

and the positiveness of the water height. In the framework of adaptive mesh, we refer to [22, 9, 30] and the reference therein.

2.2.1 Semi-discrete approximation

In order to get the semi-discrete approximation we proceed in two steps:

Step 1: we first construct an approximation for the homogeneous system. On a given cell C_k, noting $w_k(t)$

$$\boldsymbol{w}_k(t) \simeq \frac{1}{|C_k|}\int_{C_k} \boldsymbol{w}(\boldsymbol{x},t)\, dx_1 dx_2$$

the approximation of the mean value of the unknown $\boldsymbol{w}(\boldsymbol{x},t)$ on C_k at time t, and integrating (2) over each cell, we obtain:

$$\int_{C_k} \frac{\partial \boldsymbol{w}(t)}{\partial t} dx_1 dx_2 + \sum_a \int_{\partial C_{k/a}} \boldsymbol{f}(t,\boldsymbol{w}) \cdot \boldsymbol{n}_{k/a} ds = 0 \tag{5}$$

where $\boldsymbol{n}_{k/a}$ denotes the unit normal vector on the boundary $\partial C_{k/a}$ between cells k and a.
Next, $F\left(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a}\right)$ the flux approximation being written

$$\boldsymbol{F}\left(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a}\right) \approx \frac{1}{|\partial C_{k/a}|}\int_{\partial C_{k/a}} \boldsymbol{f}(t,\boldsymbol{w})\cdot\boldsymbol{n}_{k/a}ds\ ,$$

the semi-discrete finite volume approximation of equations (2) (see for instance [15, 41, 13]) is obtained:

$$\frac{\partial \boldsymbol{w}_k(t)}{\partial t} + \frac{1}{|C_k|}\sum_a |\partial C_{k/a}|\boldsymbol{F}\left(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a}\right) = 0 \tag{6}$$

where $\boldsymbol{F}\left(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a}\right)$ is defined via the Godunov solver, *i.e.* computed with the exact solution of the 1D Riemann problem at the interface k/a with the states $\boldsymbol{w}_k(t)$ and $\boldsymbol{w}_a(t)$. A second order approximation in space, the MUSCL reconstruction is also implemented (see for instance [15, 41, 11]). To be consistent with the above discretisation of the equations (2), note that the entropy inequality (3) is approximated as follows

$$\mathscr{S}_k(t) = \frac{\partial s_k(t)}{\partial t} + \frac{1}{|C_k|}\sum_a |\partial C_{k/a}|\boldsymbol{\phi}\left(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a}\right) \tag{7}$$

where $\boldsymbol{\phi}(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a})$ represents the entropy flux calculated from the resolution of the Riemann problem at the interface of cells k and a as done above.

Step 2: To take into account the source term $\boldsymbol{G}$, we use the hydrostatic reconstruction. For each interface, the numerical flux $\boldsymbol{F}\left(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a}\right)$ is replaced by the flux at left hand side $\boldsymbol{F}_k\left(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a},\Delta Z_{k/a}\right)$ and the right hand side $\boldsymbol{F}_a\left(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a},\Delta Z_{k/a}\right)$ of the interface k/a. These new fluxes are given by

$$\begin{aligned}
\boldsymbol{F}_k\left(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a},\Delta Z_{k/a}\right) &= \boldsymbol{F}\left(\boldsymbol{w}_k^*,\boldsymbol{w}_a^*,\boldsymbol{n}_{k/a}\right) + \begin{pmatrix} 0 \\ \frac{g}{2}(h_k^2-(h_k^*)^2)n_1 \\ \frac{g}{2}(h_k^2-(h_k^*)^2)n_2 \end{pmatrix} \\
\boldsymbol{F}_a\left(\boldsymbol{w}_k(t),\boldsymbol{w}_a(t),\boldsymbol{n}_{k/a},\Delta Z_{k/a}\right) &= \boldsymbol{F}\left(\boldsymbol{w}_k^*,\boldsymbol{w}_a^*,\boldsymbol{n}_{k/a}\right) + \begin{pmatrix} 0 \\ \frac{g}{2}(h_a^2-(h_a^*)^2)n_1 \\ \frac{g}{2}(h_a^2-(h_a^*)^2)n_2 \end{pmatrix}
\end{aligned}$$

by means of reconstructed states (satisfying the still water steady state equation (4))

$$\begin{aligned}
\mathbf{w}_k^* = (h_k^*,\mathbf{u}_k)\ , \quad \mathbf{w}_a^* = (h_a^*,\mathbf{u}_a)\ , \\
h_k^* = \max(0,h_k-\max(0,\Delta Z_{k/a}))\ , \quad h_a^* = \max(0,h_a-\max(0,-\Delta Z_{k/a}))\ .
\end{aligned}$$

In theses formula, $\Delta Z_{k/a}$ stands for the jump of Z across the interface k/a. The scheme is by construction well-balanced, *i.e.*, the still water steady states are ex-

actly satisfied (see [30] for further details). Finally, the numerical entropy fluxes are replaced with the reconstructed one as done above for the numerical fluxes $\boldsymbol{F}$.

2.2.2 Time integration

The time integration of eqs. (6) and (3) can be achieved in a classical way either by a Runge-Kutta or Adams-Bashforth scheme. For instance, by integrating eq. (6) (and (3)) during the time step $]t_n, t_{n+1}[$ of length δt_n and by evaluating the numerical fluxes at time t_n, the well-known second order Runge-Kutta method is

$$\boldsymbol{w}_k(t_{n+1}) = \boldsymbol{w}_k(t_n) - \frac{\delta t_n}{|C_k|} \sum_a |\partial C_{k/a}| \boldsymbol{F}\left(\boldsymbol{w}_k(t_{n+1/2}), \boldsymbol{w}_a(t_{n+1/2}), \boldsymbol{n}_{k/a}\right) \tag{8}$$

where

$$\boldsymbol{w}_k(t_{n+1/2}) = \boldsymbol{w}_k(t_n) - \frac{\delta t_n}{2|C_k|} \sum_a |\partial C_{k/a}| \boldsymbol{F}\left(\boldsymbol{w}_k(t_n), \boldsymbol{w}_a(t_n), \boldsymbol{n}_{k/a}\right) \ .$$

Remark 1. Note that, even if the Adams-Bashforth scheme is known to be less stable and less accurate, it can be easily handled in the framework of local time stepping to save computational time as done by Ersoy *et al.* [11]. Therefore, the code can support sub-cycling in time.

To be consistent with the discretisation of equation (8), a discrete version of the entropy production, called *numerical density of entropy production* $\mathscr{S}_k^n$, is defined by integrating the equation (7) with a second order Runge-Kutta scheme.

3 BB-AMR method

Defining a robust mesh refinement criterion for multi dimensional configurations is not enough to design a suitable numerical solver. The mesh refinement threshold and the treatment of data are also crucial points and in particular the way to distribute refined cells and to share the memory in a parallel process. This point is handled within an Automatic Mesh Refinement Threshold framework in a hierarchical Block-Based way called BB-AMR. The global strategy to adapt the mesh and to manage data are presented.

3.1 AMR and Automatic thresholding methods

Numerical approximation of equations (7) leads to the so-called numerical density of entropy production which is a measure of the amount of violation of the entropy

equation (as a measure of the local residual as in [6, 18, 20, 19]). Therefore, the numerical density of entropy production provides information on the need to locally refine the mesh (*e.g.* if the solution develops discontinuities) or to coarsen the mesh (*e.g.* if the solution is smooth and well-approximated), see for instance [34, 12]. As shown in [34], if the numerical entropy flux is well-chosen, the entropy production $\mathscr{S}$ built on first order monotone schemes is almost everywhere negative definite. Some positive overshoots of the entropy production $\mathscr{S}$ are possible for non monotone data but their amplitude decreases fast under grid refinement for smooth flow. Moreover, as numerically shown for one dimensional hyperbolic equations [12, 29] and for multi dimensional flows [16], the support of the relative error almost coincides with the support of the numerical density of entropy production for first and second order schemes but the absolute magnitude $\max_k \left|\mathscr{S}_k^n\right|$ is not representative of the error. Therefore, it is important to emphasize that the numerical density of entropy production is not an error estimate, or derived from, but only an error indicator. Thus, the numerical density of entropy production does not allow to relate to the equidistribution of the error as for *a priori* or *a posteriori* based error criteria which are generally hard to obtain.

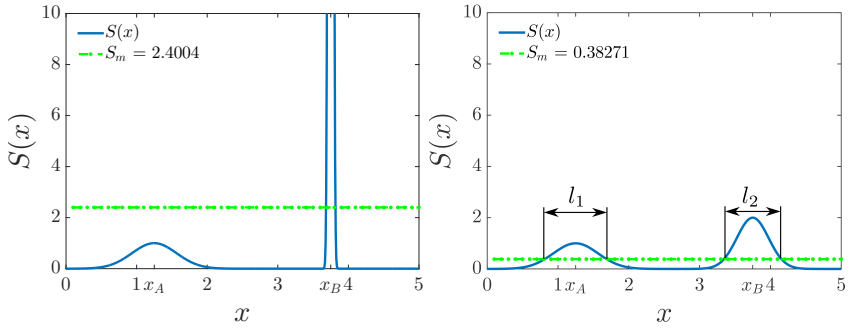
In what follows, we define the quantity $S_k^n = \left|\mathscr{S}_k^n\right|$ also referred as the local numerical entropy production. The local numerical entropy production is compared to $S_m = \frac{1}{|\Omega|}\sum_k S_k^n$. In practice, two coefficients $0 \leqslant \beta_{\min} \leqslant \beta_{\max} \leqslant 1$ are thus defined to determine the ratio of numerical production of entropy leading to mesh refinement or mesh coarsening.

For each cell C_k:

- if $S_k^n > S_m\beta_{\max}$, the mesh is refined and,
- if $S_k^n < S_m\beta_{\min}$ the mesh is coarsened whenever it is possible following the rule defined later.

This approach is certainly the most simple and the most relevant for smooth flows. For discontinuous flows, the comparison of the error indicator to the mean value S_m may lead to some problems to catch relevant small scales to be refined. In Fig. 1, we have illustrated formally a discontinuous flow, see Fig. 1(a), and a smooth one, see Fig. 1(b). In Fig. 1(a), the peak localised at $x = x_B = 3.75$ corresponds either to a sharp gradient or a discontinuity in the solution while the other part is the region where the solution is smooth. One can see in Fig. 1(a) that if the parameters $\beta_{\min}$ is not set small enough then the smooth part is not refined and only a small region around the sharp gradient or the discontinuity of the solution is refined. This is exactly what we mean by "some problems to catch relevant small scales to be refined". As illustrated in Fig. 1(b) and in general for smooth flows, the mean value S_m allows to catch relevant scales to be refined.

Therefore, the parameters $\beta_{\min}$ and $\beta_{\max}$ have to be well-chosen (see for instance [29, Section 4] for a review of some thresholding methods). The threshold parameters $\beta_{\min}$ and $\beta_{\max}$ are thus user-calibrated for each problem to reach a good compromise between computational cost and accuracy. More precisely, $\beta_{\min}$ and $\beta_{\max}$ allow to set a percentage of mesh refinement and mesh coarsening with respect to the quantity S_m. It is not surprising that these settings will deteriorate or improve the



(a) $S(x)$ with $a = 200$, $b = 1000$ (b) $S(x)$ with $a = 2$, $b = 10$

Fig. 1 Comparison of the mean value S_m for two given criterion profiles $S(x) = a\exp(-b(x - x_B)^2) + \exp(-5(x - x_A)^2)$ where $x_A = 1.25$ and $x_B = 3.75$. S represents either a discontinuous flow (left) or a smooth flow (right).

accuracy of the numerical solution. For instance, the more $\beta_{\min}$ and $\beta_{\max}$ is small, the more accurate are the results to the expense of the computational time. Calibrating such parameters is a computational waste of time especially for real life numerical simulations which can take several hours to several weeks of computations. Unless to have an error estimate, the adaptive method with an error indicator has a major drawback concerning the selection of the threshold parameters $\beta_{\min}$ and $\beta_{\max}$. For the sake of simplicity, assuming $\beta_{\min} = \beta_{\max} = 1$, one way to overcome this problem is to set the threshold α automatically. Since $\alpha = S_m$ is in general too high for discontinuous flows and large enough for smooth flows (as illustrated in Fig. 1), it is then natural to look for α in the interval $(0, S_m]$.

In what follows, we focus on the one dimensional problem because it is easier to visualise. Keeping in mind that we want to catch relevant small scales to be refined, we look for an automatically set threshold $\alpha \in (0, S_m]$ such that for each cell C_k, if $S_k^n > \alpha$ the mesh is refined otherwise the mesh is coarsened.

Remark 2. As it is, such a strategy is time consuming because the mesh would flipping back and forth between fine and coarse grids. Therefore in practice, once the threshold parameter α is set "correctly", one can define the two quantities $\beta_{\min}$ and $\beta_{\max}$ to define a buffer region where the mesh is left alone. For instance, the top 10% is refined and the bottom 10% is coarsened while the rest of the mesh remains at the same level of refinement. Let us also add that even if $\beta_{\min}$ is set to 1, the method will detect automatically pertinent scales.

Following [29], a relevant way to define automatically the threshold α, referred as α_{PE} in the sequel, is to construct it such that

$$f(\alpha_{\text{PE}}) = \max_{0<\alpha\leqslant S_m} f(\alpha) \text{ with } f(\alpha) = \alpha d(\alpha) \tag{9}$$

In this expression, $d : \mathbb{R} \to \mathbb{R}^+$, $\alpha \mapsto \text{meas}\{S(x) > \alpha\}$ is the decreasing rearrangement, also known as the distribution function, of the mesh refinement criterion S. meas$\{X\}$ stands for the Lebesgue measure of the set X. For instance in Fig. 1(b), for $\alpha = S_m$, $d(\alpha) = \text{meas}\{S(x) > \alpha\} = l1 + l2$. The decreasing rearrangement d provides a complete description of the criterion in terms of local maxima which are sorted canonically from the smallest to the largest. Following [29], assuming that the function S admits a finite number of strict local maxima, then d is only continuous and each singularity of its derivative corresponds to a local maximum as displayed in Fig. 2 for discontinuous flows and smooth flows. For instance in Fig. 2(b), the first singularity of d is localised at $\alpha = \alpha_A = 1$ and it corresponds to the first local maximum $\alpha_A = S(x_A)$ of S at $x_A = 1.25$ in Fig. 1(b). The second singularity at $\alpha = \alpha_B = 2$ (see Fig. 2(b)) represents the maximum $\alpha_B = S(x_B)$ of S at $x = x_B = 3.75$ (see Fig. 1(b)). For discontinuous flows, due to the large peak, a zoom in is necessary to display the local maximum in x_A, from the distribution function d, as shown in Fig. 4(a).

However in these examples with the distribution function only, it is hard to compute numerically a threshold α close to the first local maximum α_A which represents the so-called "relevant scales to be refined". Therefore, we use the "weighted distribution function" $f(\alpha) = \alpha d(\alpha)$ which is represented in Fig. 3 and 4(b) for discontinuous flows and smooth flows. As proved in [29, Corollary 1], the function f transforms almost the inflexion points of d in local maximum. As a consequence, the relevant threshold α_{PE} is in general smaller than α_A for discontinuous flows or equal to S_m for smooth flows. Moreover, the threshold α_{PE} is easier to compute numerically from f than d.

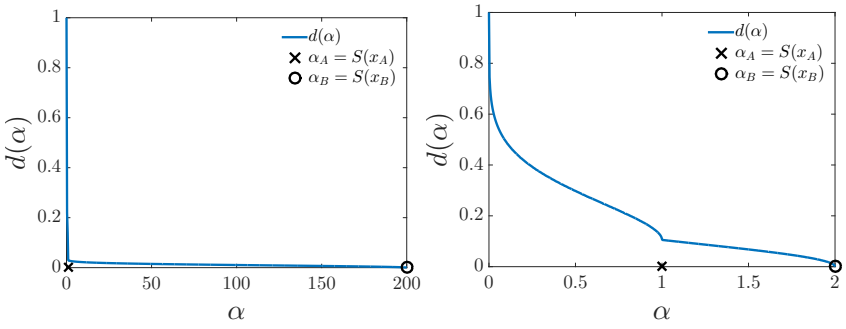
We have illustrated the computed threshold for the previous example for discontinuous flows in Fig. 5(a) (see also a zoom in displayed in Fig. 4(b)) and for smooth flows in Fig. 5(b). With this method in practice, we can compute the relevant threshold to refine for both flows. Namely, the threshold α_{PE} verifies the desired behaviour $\alpha_{\text{PE}} < S_m$ for discontinuous flows and $\alpha_{\text{PE}} \approx S_m$ for smooth flows. Thus, the presented AMR process, in comparison with existing methods, is *almost parameterless*.

Following [29], from a numerical viewpoint, the distribution function is defined as follows. For each time step, let us consider a given discrete mesh refinement criterion $S(x) = \sum_{k=1}^{N} S_k^n \mathbb{1}_{C_k}(x)$ where N is the number of cells at time t_n and $\mathbb{1}_{C_k}(x)$ is the indicator function of the mesh C_k. Let $(\alpha_j)_{0 \leqslant j \leqslant M}$ be an increasing sequence of $M+1$ threshold parameter such that

$$(\alpha_j)_{0 \leqslant j \leqslant M} = \left(S_m \left(\frac{j}{M} \right)^{\beta} \right)_{0 \leqslant j \leqslant M} \tag{10}$$

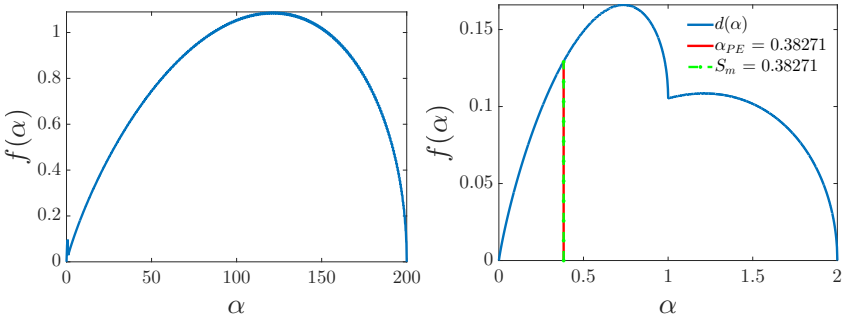
Then, the distribution function $d_j = d(\alpha_j)$ is given by

$$d_j = \#\{k \; ; \; S_k^n > \alpha_j\} \tag{11}$$



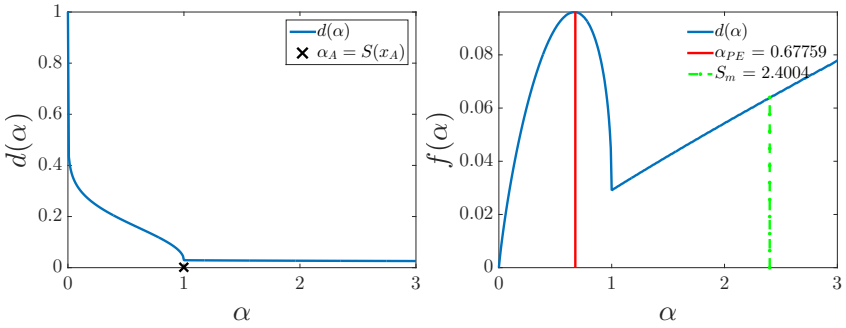
(a) $d(\alpha)$ for $S(x)$ with $a = 200$, $b = 1000$ (b) $d(\alpha)$ for $S(x)$ with $a = 2$, $b = 10$

Fig. 2 Decreasing rearrangement $d(\alpha)$ of $S(x) = a\exp(-b(x-x_B)^2) + \exp(-5(x-x_A)^2)$, where $x_A = 1.25$ and $x_B = 3.75$, for discontinuous flow (left) and smooth flow (right).



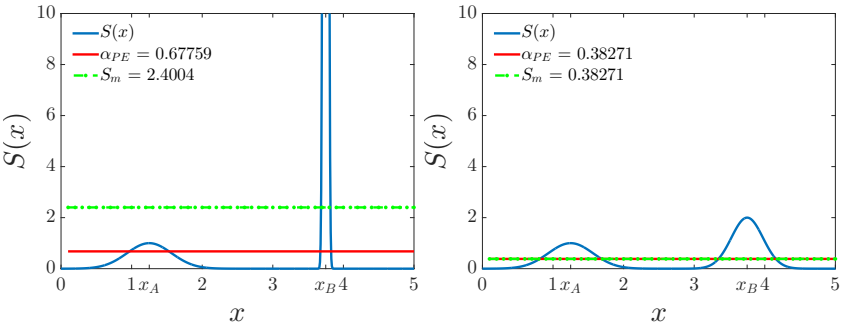
(a) $d(\alpha)$ for $S(x)$ with $a = 200$, $b = 1000$ (b) $S(x)$ for $S(x)$ with $a = 2$, $b = 10$

Fig. 3 Function f of $S(x) = a\exp(-b(x-x_B)^2) + \exp(-5(x-x_A)^2)$, where $x_A = 1.25$ and $x_B = 3.75$, for discontinuous flow (left) and smooth flow (right).



(a) $d(\alpha)$ for $S(x)$ with $a = 200$, $b = 1000$ (b) $f(\alpha)$ for $S(x)$ with $a = 200$, $b = 1000$

Fig. 4 Zoom in on $d(\alpha)$ and $f(\alpha)$, of $S(x) = a\exp(-b(x-x_B)^2) + \exp(-5(x-x_A)^2)$, where $x_A = 1.25$ and $x_B = 3.75$, for discontinuous flow.



(a) $f(\alpha)$ for $S(x)$ with $a = 200$, $b = 1000$ (b) $f(\alpha)$ for $S(x)$ with $a = 2$, $b = 10$

Fig. 5 Automatic threshold α_{PE} of $S(x) = a\exp(-b(x-x_B)^2) + \exp(-5(x-x_A)^2)$, where $x_A = 1.25$ and $x_B = 3.75$, for discontinuous flow (left) and smooth flow (right).

where # is the number of elements in the set $\{S_k^n > \alpha_j\}$. In [29, Section 4], it is shown that the method is quite insensitive to the parameter β, set to 2 here. For all applications M is fixed to $M = 1000$ points. Interested readers can found details in [29] and a review of existing threshold methods.

3.2 Mesh refinement process and data management

For the one dimensional case, the local mesh refinement procedure is constructed following dyadic tree applied at each time step. "Macro cells" are used to be easily refined by generating hierarchical grids. Each cell can be split in two. For stability reasons, the mesh refinement level cannot exceed 2 between two adjacent cells as detailed in [11] and recalled hereafter. By convention, if the maximum the level of refinement is $l_{\max} = 1$ then the mesh is not adapted.

The multi dimensional extension of the mesh refinement procedure is a difficult task. Interesting works have been presented for 2D Cartesian grid or quad-tree [5, 43, 27, 44], octree for 3D simulations [24, 14], and anisotropic AMR [8, 17]. For the Saint-Venant system (1) the extension from 1D to 2D leads naturally to quad-tree meshing. But, the presence of a complex moving interface (composed of rarefaction and shocks) implies to re-mesh at each time step, which is obviously a costly process. Guided by the need to reach a relevant compromise between the contradictory aims of solution accuracy and computing speed, a regular block-based mesh approach is introduced, somehow like in [42, 45]. If the mesh is not refine at each time step, the patch where the grid should be refine must be enlarged. The grid generation in the framework of BB-AMR can be developed as follow.

- Firstly, the initial computational domain is divided in several quadrilateral elements, called blocks as represented in Fig. 6(a).

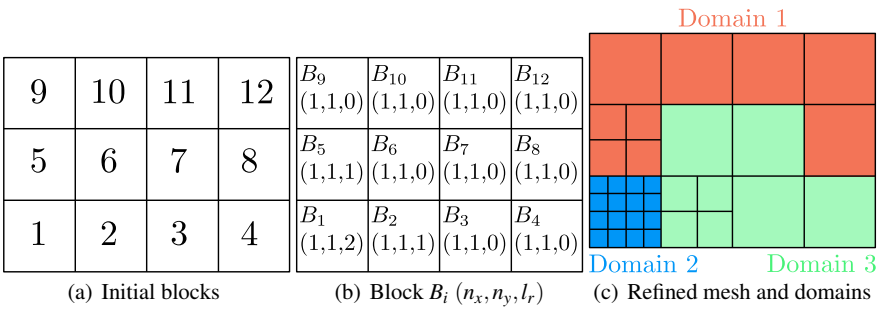
- Secondly, we define for each block a discretization n_x in x-direction, n_y in y-direction and a level of refinement l_r such that the total number of cell in x-direction of the block will be $2^{l_r} n_x$. As the mesh refinement level cannot exceed 2 between two adjacent blocks (see [11]), the level of mesh refinement is automatically adapted if necessary as illustrated in Fig. 6(b) for the blocks $B5$ and $B2$.
- Thirdly, in order to balance the CPUs load, the cells of each block are almost equi-distributed in a fixed number of domains according to the Cuthill-McKee numbering, see Fig. 6(c).
- The unstructured mesh[1] is built for each domain. At the interface between two domains, ghost cells are created in order to transfer the necessary data to the explicit time integration scheme at each time step. The number of domain being fixed, each domain are loaded in a given MPI process. These MPI processes are then load on a fixed number of cores (not necessary the same).
- The re-numbering and re-meshing being expensive, the mesh is unchanged during a "macro-time" step $\Delta t_n \geqslant \delta t_n$, called AMR time-step, given by the smallest size block $|B_k|$ (rather by the smallest size cell $|C_k| \leqslant |B_k|$) and the maximum velocity. The (micro-)time step δt_n is the usual time step, computed with the smallest size cell, required to the time advancement to ensure the stability of the scheme.

For each refined cells (or blocks), averaged values are projected onto each sub-cell and fluxes are computed as simply as possible to avoid heavy computation following Osher and Sanders [28]. Even if the Osher and Sanders projection leads to non conservative and non consistent scheme at the space-time grid between two levels of refinement, it has the advantage to be less time consuming than complex approach such as [5, 40, 11]. Let us also note that the apparent loss of consistency on the local truncation error seems not to affect the actual error of the scheme (see for instance [7, 39]). Therefore, a simple issue is proposed in [11]. As said above, the loss of the consistency and conservativity error can be minimised using smoothing grid technique, that is to say, we prevent two adjacent cells from having a level difference greater than two. More Details on the BB-AMR can be found in [16, 1].

Finally, to compute the threshold automatically in any dimensional framework, the decreasing rearrangement d of the mesh refinement criterion function S is defined (*cf.* eq. 11). Then, the threshold is automatically defined as explained in the previous section (*cf.* eq. 9). In the context of Block Based AMR, if one cell or more of any block has a criterion below the threshold, then the entire block is refined.

Remark 3 (A comment on the Osher and Sander projection for vertically averaged Saint-Venant system). To be rigorous with respect to the underlying physical properties of the system, the projection have to be mass, momentum and energetically conservative. The smoothing grid technique works well for the most of the system but may fails for vertically averaged models like Saint-Venant equations. The main

[1] The Cartesian grids used in this chapter are treacherous because there are often associated to structured mesh. This choice has been done to get a simple meshing tool but is unstructured by construction.



(a) Initial blocks (b) Block B_i (n_x, n_y, l_r) (c) Refined mesh and domains

Fig. 6 Illustration of the BB-AMR process

reason is related to the topography variation which set some difficulties for this simple projection method. As it is for any three dimensional models, the bathymetry does not act on the flow as a boundary condition but acts directly into the model as a source term. As a consequence, we have to pay attention to the new definition of the bathymetry onto the new refined or coarsened mesh as done in, for instance, [5, 22, 30] which propose to interpolate the free surface $\eta = h + Z$ instead of the conservative variable h. This interpolation strategy prevents the introduction of new extrema in the water surface elevation and preserves a flat sea surface. In that paper, the simple approach developed in [30] is used. This method is mass conservative and allows the computation of the solution close to or at dry states near the shore, being well-balanced by construction if the initial scheme is. For the ease of the reader and for the simplicity of the reader we describe the method in the one dimensional framework: let us denote C_0 the initial cell of length $|C_0|$ and, C_{00} and C_{01} the two sub-cells of length $|C_{00}|$ and $|C_{01}|$, then

step 1 (mass conservation property): For a constant density of water ρ, the mass is conserved if and only if

$$\int_{\text{old mesh}} \rho dV = \int_{\text{new mesh}} \rho dV \iff \rho |C_0| h_0 = \rho \left(|C_{00}| h_{00} + |C_{01}| h_{01} \right) \tag{12}$$

Keeping in mind that $h = \eta - Z$ and we impose $\eta = \text{cst}$ for steady states, this relation yields to

$$Z_0 = \eta \left(1 - \frac{|C_{00}|}{|C_0|} - \frac{|C_{01}|}{|C_0|} \right) + \frac{|C_{00}|}{|C_0|} Z_{00} + \frac{|C_{01}|}{|C_0|} Z_{01}$$

Straightforwardly, if $|C_{00}| = |C_{00}| = \frac{|C_0|}{2}$, the mass conservation is satisfied if $Z_0 = \frac{Z_{00}+Z_{01}}{2}$.

step 2 (mesh refinement): Once the bathymetry is defined, the water height is reconstructed from $h_i = \max(0, \eta_i - Z_i)$ and the free surface conservation $\eta = cste$

yielding to $h_{0i} = \max(0, h_0 + z_0 - z_{0i})$ where the max plays a crucial role near shore for dry states. The water speed u_{0i} is reconstructed form the momentum conservation relation similar to the previous one.

step 2 (mesh coarsening): In that case, to go from the fine grid values to a coarse-grid value we average by setting $h_0 = \frac{|C_{00}|h_{00}+|C_{01}|h_{01}}{|C_0|}$ and define the water speed $u_0 = \frac{|C_{00}|h_{00}u_{00}+|C_{01}|h_{01}u_{01}}{|C_0|h_0}$.

4 Numerical validations

We now focus on the overall performance of the BB-AMR scheme with automatic thresholding confronted to experimental and state-of-the-art numerical results[2]. For each test case, the numerical simulations are carried out on adaptive grids (using the BB-AMR method previously presented) and on uniform grids. The minimum adaptive cell size $\delta x_{adaptive}$ is equal to the uniform one $\delta x_{uniform}$. The comparisons of both simulations allow to show the reliability and the efficiency of the BB-AMR method. To simplify the analysis, we have considered only 1 domain, *i.e.*, 1 processor. Interested readers can found a detailed numerical study with several domains in [16]. For all computations,

- the MUSCL reconstruction and the Runge Kutta method are used to achieve a second order scheme in space and time.
- to simplify the analysis and to focus on the automatic threshold, we have chosen $\beta_{\min} = \beta_{\max} = 1$. Indeed, one can improve the above results by setting $0 < \beta_{\min} < \beta_{\max} < 1$.
- the CFL number is set to 0.9,
- For the sake of simplicity, the re-meshing time step is fixed constant to 0.25 s. This choice roughly verifies the macro CFL condition.
- the code is written in `Fortran 90`, compiled with `gfortran` on Linux machine `Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz`.

4.1 Solitary wave propagation over a two dimensional reef

This benchmark aims to reproduce a set of laboratory experiments carried out at the O.H. Hinsdale Wave Research Laboratory, Oregon State University (OSU, see Roeber *et al.* [36] and Roeber and Cheung [35]). These experiments involve the propagation, run-up, splash-up and reflection of high amplitude solitary waves on two-dimensional reefs. Their purpose is on one hand to investigate processes related to breaking, bore formation, dispersion, and passage from sub- to super-critical flows, and on the other hand, to provide data for the validation of near-shore wave

[2] The numerical soft used for the following test cases is the EOLENS code developed by the Institut de Mathématiques de Toulon (IMATH) and Principia.

models in fringing reef. Even if the Saint-Venant model (2) is not relevant for dispersive flows, we show, through this test case, that globally experimental data are well-reproduced and the cells are well-refined thanks to the automatic thresholding technique.

4.1.1 Experimental settings

The geometry of the test considered here is shown in Fig. 7. The length of the basin is of 104 m, however the computational domain is delimited by a reflecting wall placed at $x = 83.7$ m. The reef starts at $x = 25.9$ m with a nominal slope of $1/12$. The actual slope is such that the height of 2.36 m is reached after $x = 28.25$ m. At this station a 0.2 m height crest is mounted. The offshore slope of the crest is the same of the reef, and the length of its plateau is of 1.25 m. The on-shore side has a slope of $1/15$ giving a nominal length for the crest basis of 6.65 m (using the actual offshore slope, a crest basis of 6.644 m is obtained). This gives an offshore length of the crest slope (starting at $x = 28.25$ m) of 2.4 m. Thus, the bathymetry is

$$Z(\boldsymbol{x}) = \begin{cases} -2.5 & \text{if } x < 25.9\,, \\ \frac{(x-25.9)}{30.8}2.565 - 2.5 & \text{if } 25.9 < x < 56.7\,, \\ 0.065 & \text{if } 56.7 < x < 57.65\,, \\ \frac{(57.95-x)}{3}0.2 + 0.0650 & \text{if } < 56.7 < x < 60.95\,, \\ -0.1360 & \text{if } x > 60.95\,. \end{cases}$$

The initial depth at still water is $h_0 = 2.5$ m, yielding to a partially submerged crest, and a depth behind it (on-shore side) of 0.14 m. The initial solution consists of a solitary wave of amplitude $A = 0.75$ m providing a non linearity ratio of $\frac{A}{h_0} = 0.3$. For numerical purpose, the initial data consist of a solitary wave centered at $x_0 = 17.6$ m of amplitude

$$\eta(x,0) := h(x,0) + Z(\boldsymbol{x}) = \frac{A}{\cosh\left(\sqrt{\frac{3A}{(4h_0^3)}}(x - x_0)\right)^2} \tag{13}$$

and velocity

$$u(x,0) = \sqrt{g(h_0 + A)}\eta(0,x)/h_0 \tag{14}$$

which corresponds to the experimental data at the dimensionless time $\tilde{t} = t\sqrt{\frac{g}{h_0}} = 55.03$ as displayed in Fig. 8(a).

4.1.2 Numerical results vs experimental results

Figures 8(a) and 8(b) shows the propagation of the solitary wave over the slope starting at $x = 25.9$ m. Figure 8(c) represents the surface elevation at the dimen-

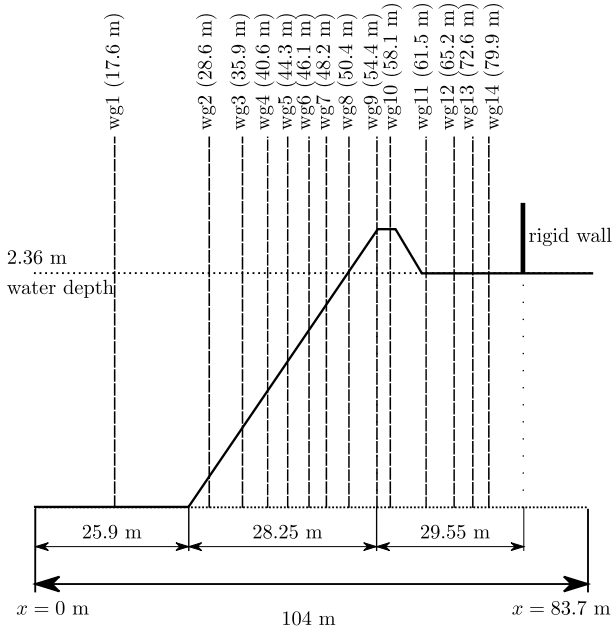


Fig. 7 Experimental settings and wave gauges locations

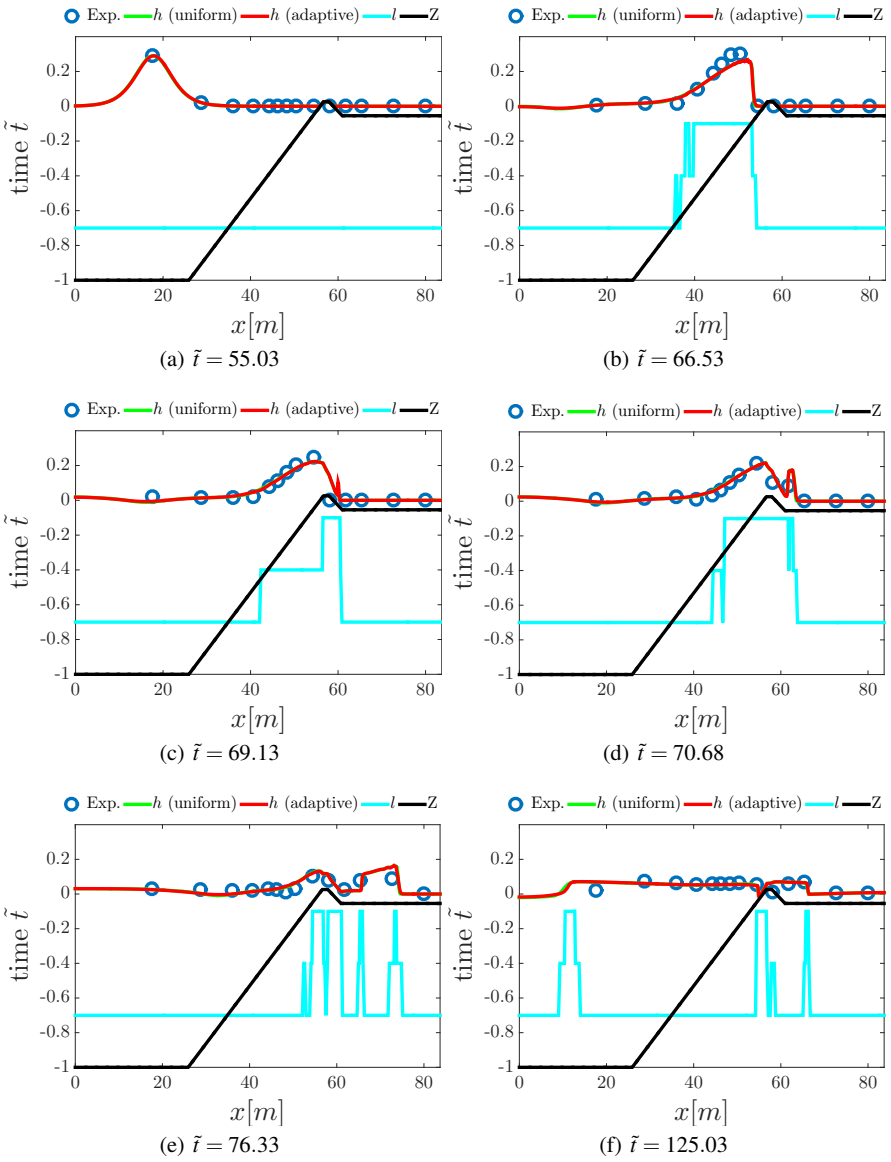
sionless time $\tilde{t} \approx 69$ where the wave develops into a plunging breaker onto the reef crest[3]. From an experimental viewpoint, $\tilde{t} \approx 69$ corresponds to the subsequent time of overturning of the free surface. In Fig. 8(d), the wave hitting the reef creates a downstream and upstream propagating bore as shown in Fig. 8(e). The downstream propagating bore is reflected at the end wall at $x = 83.7$ m and propagates in opposite direction as displayed in Fig. 8(f).

In Figs. 8(a)–8(f), the experimental surface elevation (represented with blue circles) for several dimensionless time $\tilde{t}$ are compared to the obtained numerical results (green and red lines-circles). The numerical solution is computed on a uniform grid (green lines-circles) composed of 1000 cells and on an adaptive grid (red lines-circles) initially composed of 200 blocks, composed initially with 1 cell, with $l_{\max} = 3$ where $l_{\max}$ is the maximum level of the mesh refinement. Free boundary condition at $x = 0$ m and reflecting boundary condition at $x = 83.7$ m are imposed.

The obtained numerical results, as shown in Figs. 8(a)–8(f), are rather well-computed when compared to experimental data. However, the wave height is slightly underestimated (see Fig. 8(c)) while the hydraulic bore height is overestimated (see Fig. 8(d)).

We now compare in Figs. 9(a)–9(f), the computed and the experimental water elevation at the wave gauges displayed in Fig. 7. The oscillating shape of the experimental measures almost corresponds to dispersive effects which cannot be detected

[3] The authors would thanks V. Roeber who provide the experimental data for this test case.



(a) $\tilde{t} = 55.03$ (b) $\tilde{t} = 66.53$

(c) $\tilde{t} = 69.13$ (d) $\tilde{t} = 70.68$

(e) $\tilde{t} = 76.33$ (f) $\tilde{t} = 125.03$

Fig. 8 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

with the Saint-Venant model (2). However, the shape of the numerical results almost coincides with the experimental data for wave gauges located before $x = 54.4$ m as shown in Figs. 9(a)–9(c). The point $x = 54.4$ m is located just before the crest and in particular it corresponds to the region where the overturning of the free surface is experimentally observed. Numerically a small shift is observed in Figs. 9(d)–9(f).

4.1.3 AMR performance

We now compare the performance of the adaptive and uniform method. The adaptive scheme uses an average of 281 (varying up to 506) cells against 1000 cells (see Fig. 10(b)) for the uniform one and the CPU-time is almost around 7 s against 30 s respectively. This computational time can be improved if the top 10% is refined and the bottom 10% is coarsened for instance.

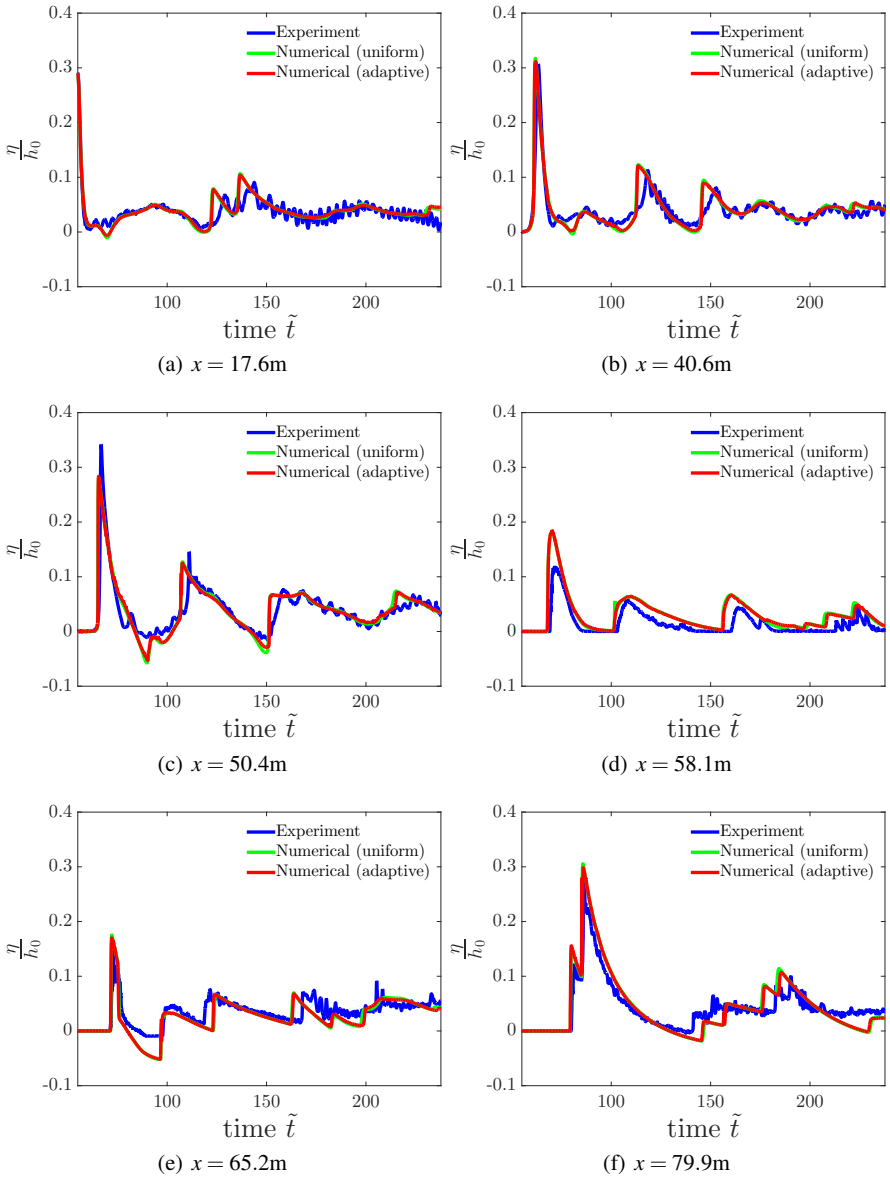
We observe that the results on adaptive grid are similar to the uniform grid ones and computed about 4 times faster. We see also in Figs. 8(a)-8(f) that for each time step, the threshold parameter is automatically well-set to capture efficiently the region to refine (see also Fig. 10(a)). During the first 50 non-dimensional time, we have already pointed out that the solitary wave propagates and splash-up. This phenomenon is characterized by a steep gradient flow followed by a discontinuous flow for the Saint-Venant system, see Fig 8. In view of the remarks in Sect. 3.1, we observe that for almost all $\tilde{t} \in (55, 100)$, the threshold is set to $\alpha_{\mathrm{PE}} < S_m$. For $\tilde{t} > 100$, the flow is almost smooth and therefore $\alpha_{\mathrm{PE}} \approx S_m$. It is also interesting to highlight that only the region of interest are automatically refined as shown in Fig. 8(e). Therefore, we recover the desired behavior predicted by Pons and Ersoy [29, Section 4]. Moreover, in view of the confrontation with the computation on uniform grid, this test shows the reliability/efficiency of the automatic selection of the threshold for multiple scale flows.

4.2 Solitary wave propagation over an irregular three-dimensional shallow shelf

In this test case, the numerical simulation of a solitary wave propagation over a complex three dimensional bathymetry is performed. This test case was experimentally introduced in [25] to understand the turbulence and kinematic properties associated with a breaking solitary wave. In this experiment, the propagation, run-up, drying and flooding phenomena are involved.

4.2.1 Experimental settings

A laboratory experiment was conducted in a large wave basin which was 48.8 m long and 26.5 m wide at Oregon State University for which free surface elevations

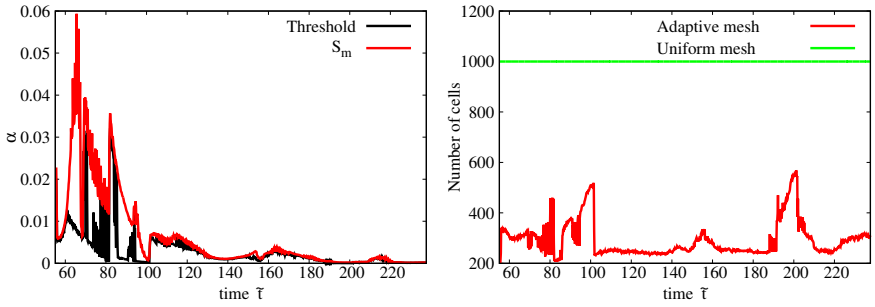


(a) $x = 17.6$m (b) $x = 40.6$m

(c) $x = 50.4$m (d) $x = 58.1$m

(e) $x = 65.2$m (f) $x = 79.9$m

Fig. 9 Surface profiles of solitary wave propagation in time at wave gauges 1 to 6. Confrontation of experimental data (solid blue line) to numerical data computed on a uniform grid (solid green line) and on an adaptive grid (solid red line)



(a) Threshold (b) Number of cells

Fig. 10 Time evolution of the mesh refinement threshold and the number of cells

and fluid velocities were recorded at several locations by wave gauges (WG) and velocity captors (ADV). A single solitary wave of height of 39 cm at $x = 5$ m is produced with a piston-type wave-maker. The bathymetry begins with a flat part allowing to generate a solitary wave. Then, starting at $x = 10.2$ m of the wave-maker and extending to $x = 25.5$ m, a complex three dimensional shelf is built as illustrated in Fig. 11. The associated slope to this zone is variable but less and less steep. Beyond that, a very small one dimensional slope finishing on a flat onshore zone is built. The experimental basin is closed by walls. Water height and velocities are recorded during 45 s. Finally, the material used for the walls and the bathymetry is made of smooth concrete so that one can assume almost a frictionless flow. For further informations on this experiment, interested readers can found more details in [25].

4.2.2 Numerical vs experimental results

The domain is numerically extended to $x = -5$ m with a water depth of 78 cm to impose a solitary wave instead of reproducing the wave maker movement. The simulated solitary wave is a first order solution of the Boussinesq equation (see equations (13) and (14)) with $x_0 = 5$ m and $\frac{A}{h_0} = 0.5$. For computational purpose, we have considered 128 initial blocks composed of 7 500 (varying up to 25 000) cells for the initialization of the adaptive computation and almost 33 000 cells for uniform mesh computation. The simulation time is 30s. Reflecting boundary conditions are prescribed to walls.

To illustrate the main propagation phenomena, we have displayed the numerical solutions calculated at time $t = 0.5$ s (see Fig. 12), $t = 2.5$ s (see Fig. 13), $t = 5.75$ s (see Fig. 14) and $t = 23.75$ s (see Fig. 15). The Fig. 12 represents the profile of the solitary wave which propagates towards the coast. Figures 13 and 14 show respectively the wave hitting the conic island and then the beginning of the flooding.

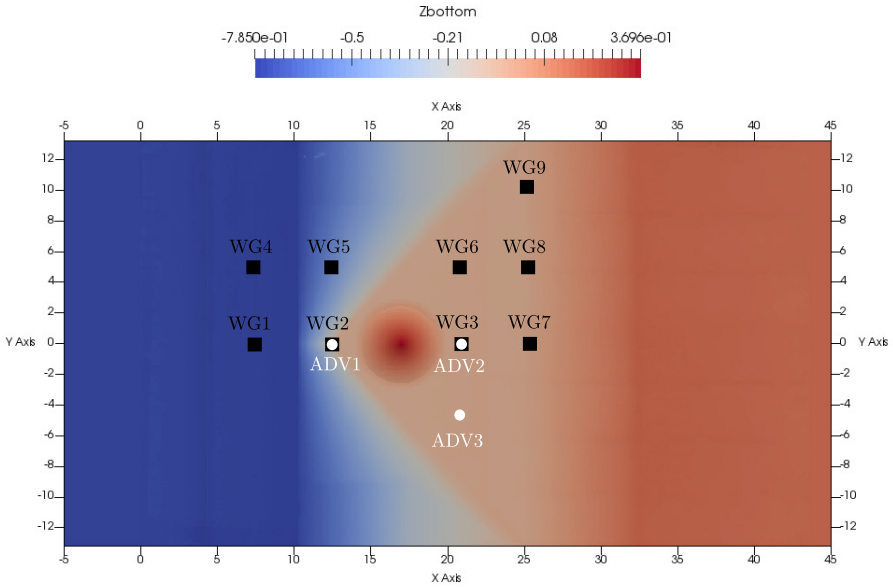


Fig. 11 Experimental settings

Figure 15 shows the flow after the wave run-up and its reflection on the right side of the domain.

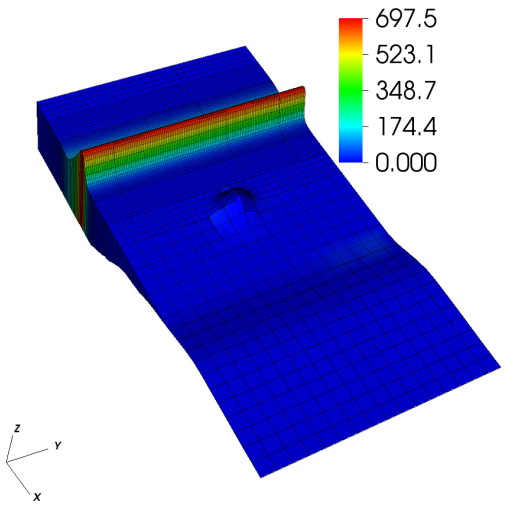


Fig. 12 Numerical result for the water height at time $t = 0.5s$. The colours correspond to the magnitude of the kinetic energy

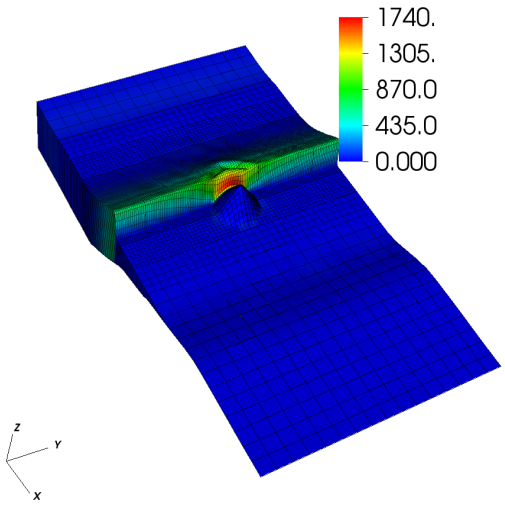


Fig. 13 Numerical result for the water height at time $t = 2.5s$. The colours correspond to the magnitude of the kinetic energy

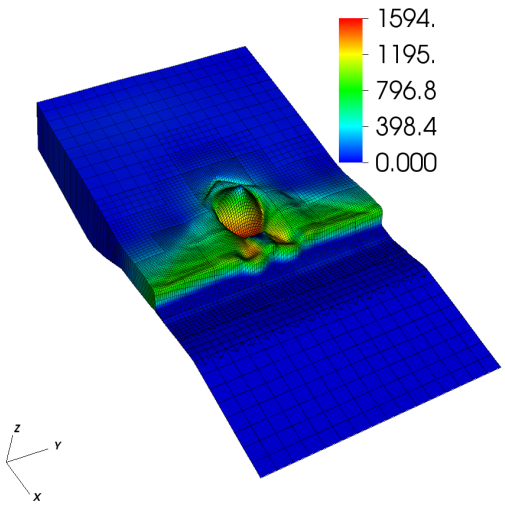


Fig. 14 Numerical result for the water height at time $t = 5.75s$. The colours correspond to the magnitude of the kinetic energy

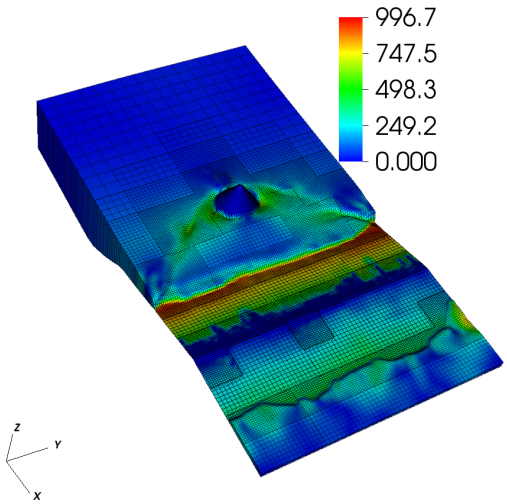
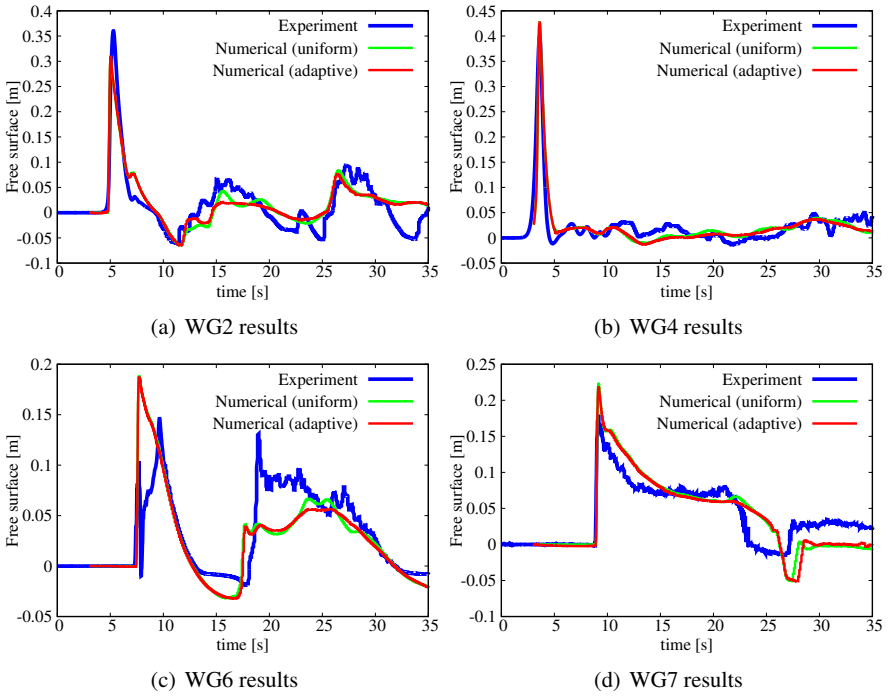


Fig. 15 Numerical result for the water height at time $t = 23.75s$. The colours correspond to the magnitude of the kinetic energy

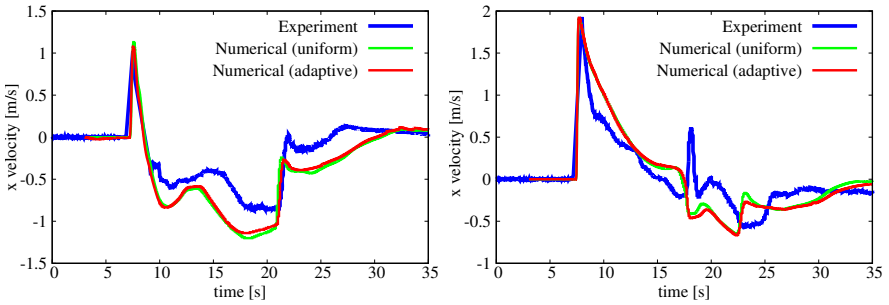
The global numerical results are now confronted to the experimental data recorded at the waves gauges 2, 4, 6 and 7 (WG) for the water height, see Fig. 11 and 16, and to the velocity captors 2 and 3 (ADV) for the first u component of the velocity vector, see Fig. 11 and 17. For the wave gauge 2 to 7 (see Figs. 16(a)–16(d)), the global shape of the free surface is in a good agreement with the experimental data but some differences are observed. The numerical solution computed with the Saint-Venant system (2) slightly underestimates the water level at the wave gauge 2 (see Fig. 16(a)) and overestimates at the wave gauge 4 (see Fig. 16(b)). Moreover, the numerical solution is rather accurate up to time $t = 15$ s at the wave gauges 6 and 7 (see Fig. 16(c) and 16(d)). For $t \geqslant 15$ s , the dispersive effects, not reproduced with the Saint-Venant system, begin significant and induce slightly different results between experimental and numerical data. Similar conclusion can be drawn up for the first component of the vector velocity on the captor 2 (see Fig. 17(a)) and 3 (see Fig. 17(b)). Globally, the numerical results on this test case are in a good agreement and can be improved with a dispersive model.



(a) WG2 results (b) WG4 results

(c) WG6 results (d) WG7 results

Fig. 16 Free surface results at different positions : experimental data versus numerical simulation with and without mesh adaptivity



(a) ADV2 u results (b) ADV3 u results

Fig. 17 Velocity results at different positions : experimental data versus numerical simulation with and without mesh adaptivity

4.2.3 AMR performance

To highlight the efficiency of the mesh refinement parameter and the automatic selection of the threshold, the numerical density of entropy production (without absolute value) and the associated mesh are displayed at several times in Figs. 18–21, (time $t = 0.5$ s in Fig. 18, for $t = 2.5$ s in Fig. 19, for $t = 5.75$ s in Fig. 20 and for $t = 23.75$ s in Fig. 21). The time evolution of the mesh refinement threshold selecting the region to be refined is shown in Fig. 22(a). It can be noticed that according to the complex flow zones the numerical density of entropy production indicates which areas need to be refined. Secondly, the mesh density is well distributed on these zones which illustrate the efficiency of the automatic thresholding method.

For instance, in Fig. 14, the region which needs to be refined is clearly the zone around the ”conic island”. According to Fig. 20, the numerical density of entropy production in this area shows that the local maxima (lowest value without the absolute value) are of almost of order 0.005 and clearly follows the wave front. We see in Fig. 22(a), approximately at time $t = 5.75$ s, that the threshold is automatically fixed to almost 0.001 and therefore allows to perform a suitable refinement in the region of interests as observed in Fig. 14 or 20.

Compared to the computation on a uniform grid (see Figs. 16 and 17), the adaptive mesh refinement scheme requires only an average of 13000 cells against 33000 cells (see Fig. 22(b) for the number of cells for all the simulation). For the same accuracy, the adaptive scheme allows to save almost 60% of cells with respect to the uniform mesh. During the adaptive simulation, the maximum refined area is reached around the time $t = 12$ s which corresponds to the time where the free surface oscillates behind the island and the wave run-up on the ”coast”. This result makes sense because the mesh has to follow a lot of oscillation and not only a solitary wave. Globally, all these cells saved allow to speed up the computation by 2.5 time for this test case.

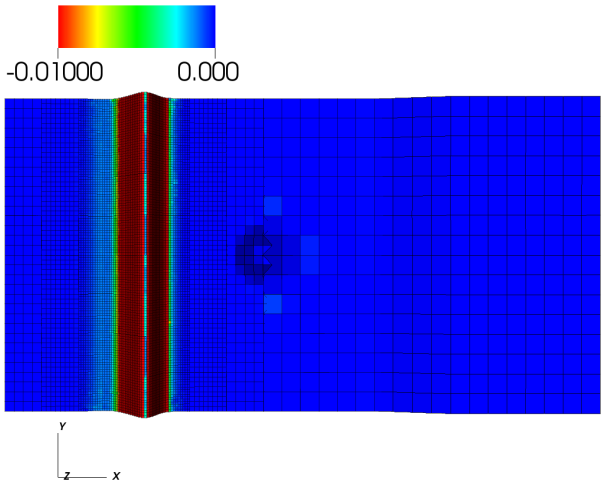


Fig. 18 Time evolution of the adaptive mesh : $t = 0.5$s. The colours correspond to the numerical density of entropy production $\mathscr{S}$

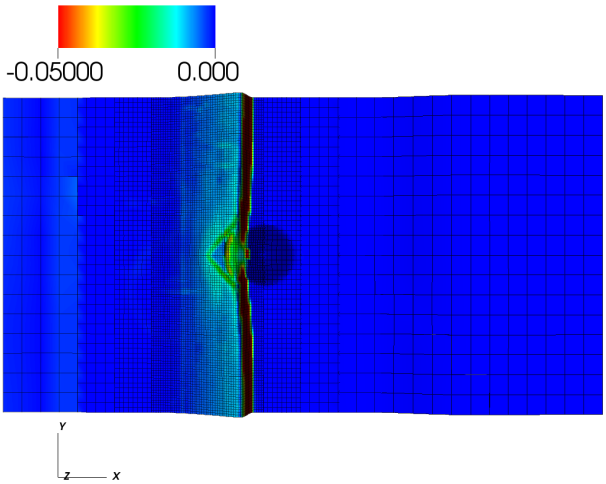


Fig. 19 Time evolution of the adaptive mesh : $t = 2.5$s. The colours correspond to the numerical density of entropy production $\mathscr{S}$

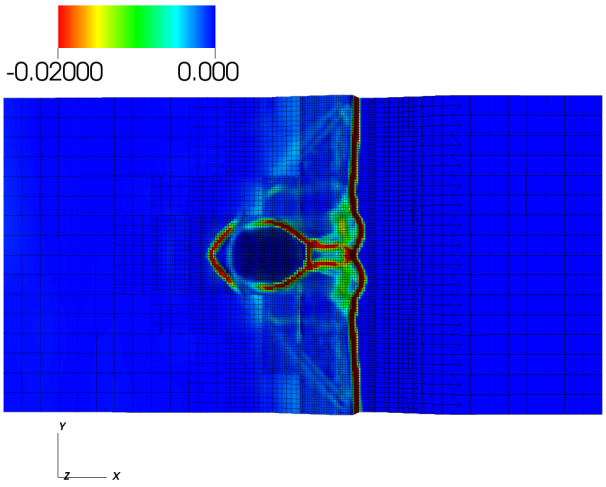


Fig. 20 Time evolution of the adaptive mesh : $t = 5.75$s. The colours correspond to the numerical density of entropy production $\mathscr{S}$

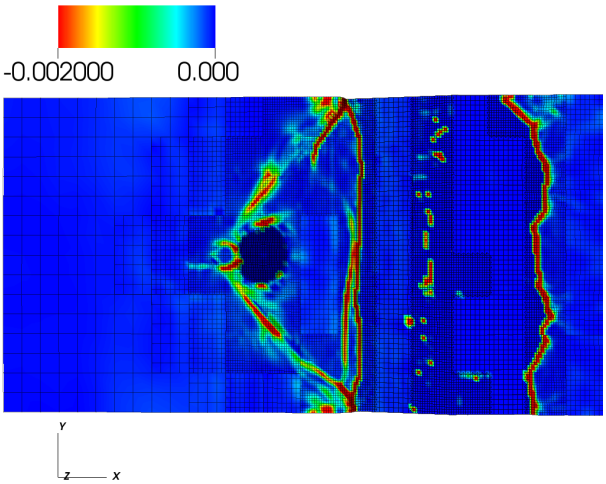
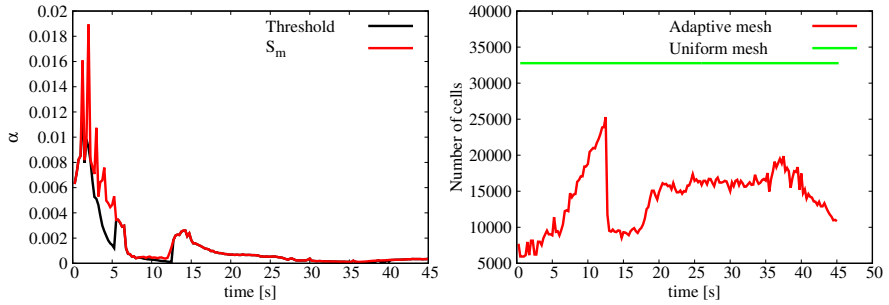


Fig. 21 Time evolution of the adaptive mesh : $t = 23.75$s. The colours correspond to the numerical density of entropy production $\mathscr{S}$



(a) Threshold (b) Number of cells

Fig. 22 Time evolution of the mesh refinement threshold and the number of cells

4.3 Tsunami runup onto a complex three dimensional Monai-Walley beach

This test case concerns the Hokkaido-Nansei-Oki tsunami, in 1993, that struck the Okushiri Island in Japan. The tsunami run-up records was about 30 m height and the currents speed of order 10-18 m/s for which relevant high-quality data were saved. [4] The largest value run-up (32 m) was recorded near the Monai-Walley beach.

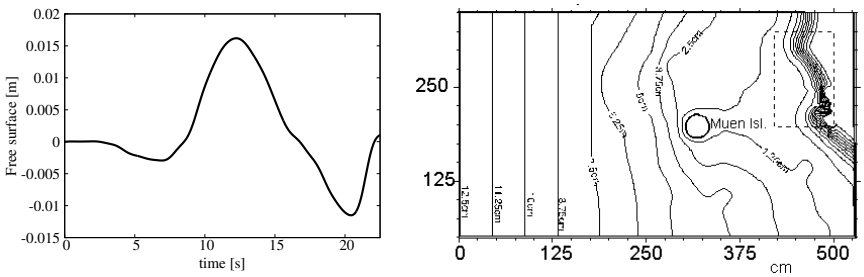
4.3.1 Experimental settings

To understand this complex run-up, a 1/400 scale laboratory model of Monai was realized in a large-scale tank of 205 m long, 6 m deep, and 3.5 m wide at the Central Research Institute for Electric Power Industry (CRIEPI) in Abiko (Japan). The offshore incident wave on a water depth $d = 13.5$ cm is prescribed. There are reflective side-walls at $y = 0$ and 3.5 m as for the x boundaries. The bathymetry as well as the coastal topography reproduced in the laboratory experiment are represented[5] in Fig. 23(b) and 23(c) .

The input wave at $x = 0$ m is a leading-depression height of -2.5 mm with a crest of 1.6 cm, as displayed in Fig. 23(a). In the experiment, the waves are measured at thirteen locations and complete time histories are given at three locations, (x,y) = (4.521, 1.196), (4.521, 1.696), and (4.521, 2.196) in meters, see also [23]. In contrast with the previous experimental test cases, the dispersive phenomena can be fully neglected here.

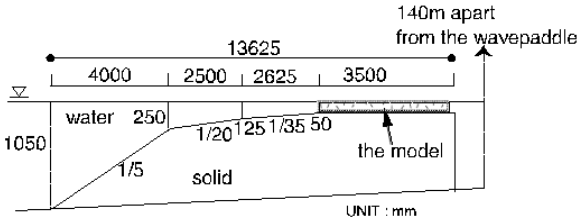
[4] Several sources and data can be found, see for instance `http://nctr.pmel.noaa.gov/benchmark/Laboratory/Laboratory\_MonaiValley/` or `http://isec.nacse.org/workshop/2004_cornell/bmark2.html`

[5] sources and pictures are available at `http://isec.nacse.org/workshop/2004\_cornell/bmark2.html`



(a) Initial wave profile

(b) Top view



(c) Side view

Fig. 23 Experimental settings

4.3.2 Numerical vs experimental results.

The initial condition for this test case is a lake at rest and an imposed water height on the left side wall, see Fig. 23(a). Wave gauges data [6] experimental results are given up to time $t = 100$ s. However, the information concerning the input wave profile (see Fig. 23(a)) at $x = 0$ m is only available up to time $t = 22$ s. For numerical purpose, we prescribe for $t > 22$ s, a constant free surface level $\eta = 0$ m on the left wall. In practice, due to this assumption, some artificial reflecting waves can occur on the wave-maker beyond $t > 22$ s and can reach the free surface gauges around $t = 30$ s. Finally, to avoid any wrong comparisons due to these artificial reflections, the comparisons between the experimental gauges and the numerical results will be only done over the first 30 s of the experiment.

For computational purpose, we have considered 240 initial blocks composed of 8 000 (varying up to 40 000) cells for the initialisation of the adaptive simulation and 62 000 cells for uniform mesh simulation. The time simulation is 30s. Reflecting boundary conditions are prescribed to walls.

To illustrate this test case we show in Figs. 24–27, the propagation of the wave at time $t = 11.25$ s (Fig. 24), $t = 13.25$ s (Fig. 25), $t = 16$ s (Fig. 26) and $t = 17.5$ s (Fig. 27). The Fig. 24 represents the tsunami wave entering inside the computational

[6] Wave gauges data are available at the address `http://nctr.pmel.noaa.gov/benchmark/Laboratory/Laboratory_MonaiValley/`

domain. The Fig. 25 shows the submersion of the conical island. Figures 26 and 27 show respectively the large flooding and run-up of the wave and its reflection on the cliff.

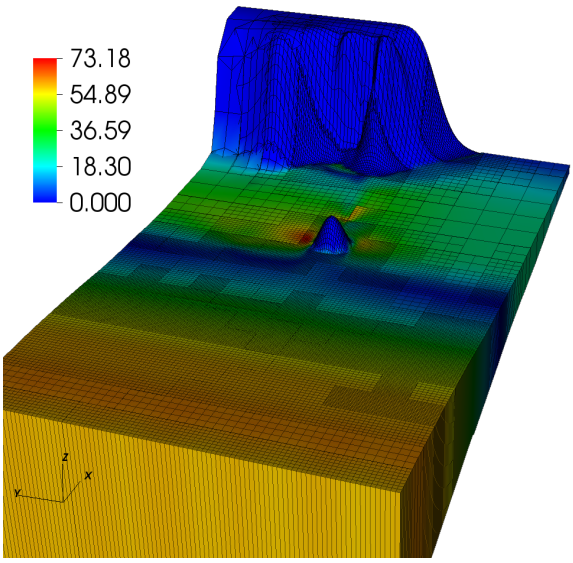


Fig. 24 Numerical result for the water height at time $t = 11.25$s. The colours correspond to the magnitude of the kinetic energy

The numerical solutions are now quantitatively confronted to the experimental measures. The comparisons are shown in Fig. 28 at the three waves gauges WG 1 (see Fig. 28(a)), 2 (see Fig. 28(b)) and 3 (see Fig. 28(c)). These figures show that the wave amplitude and the wave shape are accurately predicted at three different location in the computed domain. As emphasized before, dispersive effects being negligible, the numerical results are accurate. However, it can be noticed in Fig. 28 a small amplitude errors occurring at the beginning of the simulations. This error can be attributed to the lack of accuracy of the prescribed initial condition and boundary condition at $x = 0$. This problem was already encountered by Popinet [31].

4.3.3 AMR performance

To highlight the efficiency of the mesh refinement parameter and the automatic selection of the threshold, the numerical density of entropy production (without absolute value) and the associated mesh are displayed at several times in Figs. 29–32, (time $t = 11.25$ s in Fig. 29, for $t = 13.25$ s in Fig. 30, for $t = 16$ s in Fig. 31 and for $t = 17.5$ s in Fig. 32). We also display in Fig. 33(a) the evolution of the threshold parameter which selects the region to be refined.

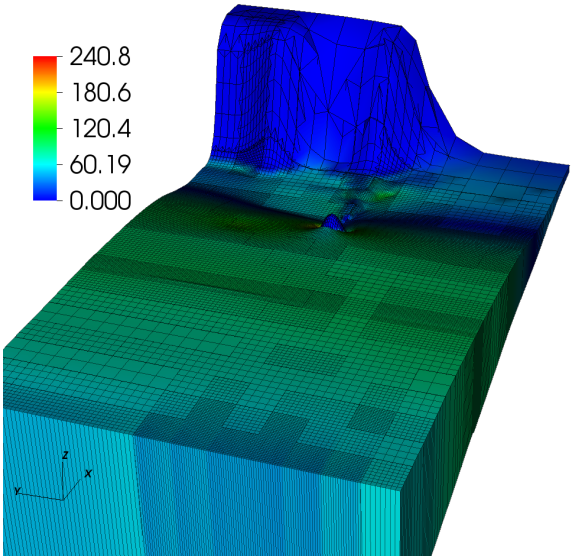


Fig. 25 Numerical result for the water height at time $t = 13.25$s. The colours correspond to the magnitude of the kinetic energy

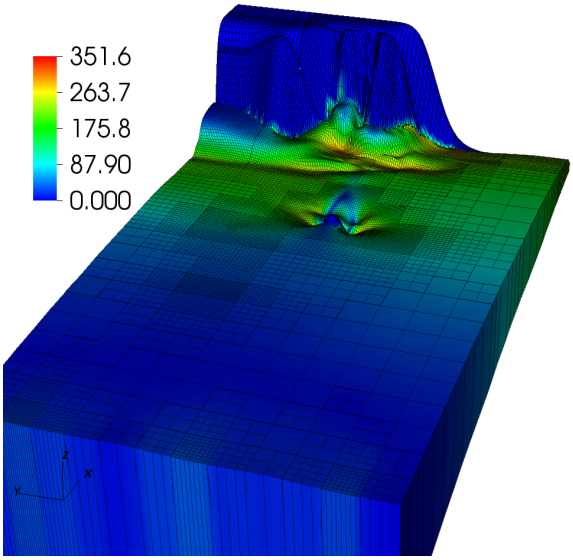


Fig. 26 Numerical result for the water height at time $t = 16$s. The colours correspond to the magnitude of the kinetic energy

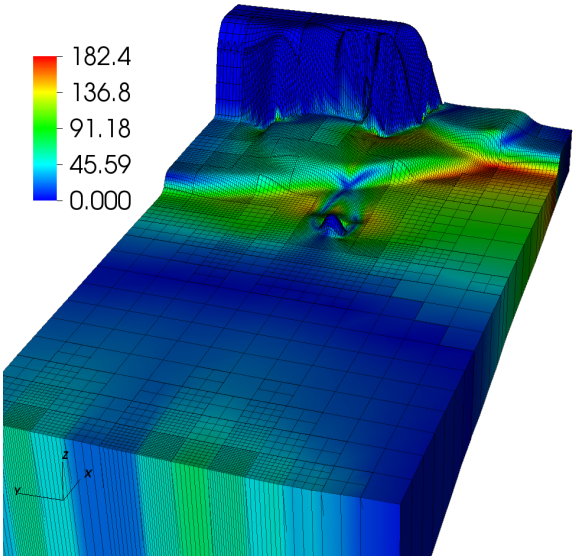
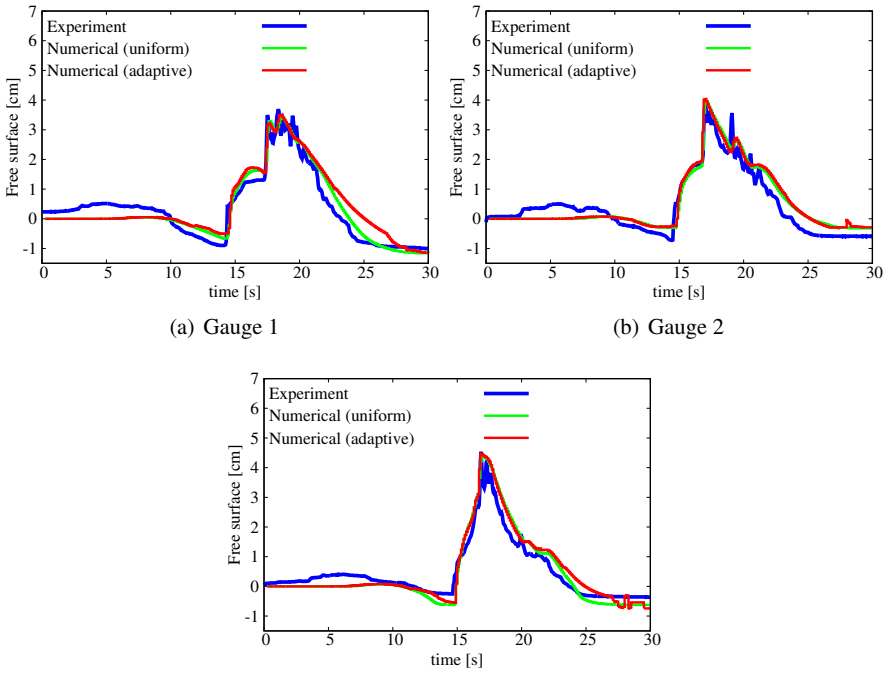


Fig. 27 Numerical result for the water height at time $t = 17.5$s. The colours correspond to the magnitude of the kinetic energy

We observe mainly that the wave front and subsequent waves are very well-captured by the criterion and the mesh refinement threshold is well chosen to refine in those areas (see for instance Fig. 27 and 32). Between 3 and 7 seconds, the tsunami propagates in the domain involving a grow-up of the number of cells. Then the wave passes through the island and reaches the reef around 15 seconds. After this, the wave is reflected and goes back to the wave maker with a second interaction with the island. The maximum number of cells is reached when the wave is passing on the island and begins to run-up the coast (see Figs. 30 and 33(a)), and when the wave goes back to the wave maker. These results are coherent with the physical process involved since it corresponds to the both times where multiple active flow regions are observed. The overall scales of the tsunami propagation with its reflections are very well captured as displayed in Fig. 29, Fig. 31 and Fig. 32.

Finally, the time evolution of the number of cells is represented in Fig. 33(b). Compared to the computation on the uniform grid (see Fig. 28), the adaptive mesh refinement method requires an average of almost 25000 cells against 62000 cells. For the same accuracy, the AMR method allows to save almost 60% of cells with respect to the uniform simulation. Thus, the AMR method allows to speed up the computation almost 3 times.



(a) Gauge 1 (b) Gauge 2

(c) Gauge 3

Fig. 28 Free surface results at different positions : experimental data versus numerical simulation with and without mesh adaptivity

5 Conclusion

In this paper, we have considered the one and two dimensional non linear non dispersive Saint-Venant system for tsunamis propagation problems. Solving accurately this model over very large ranges in spatial scale leads to heavy computational time. Therefore, we have considered a well-balanced finite volume scheme in adaptive mesh framework. The BB-AMR method is improved using an Automatic Mesh Refinement Threshold based on the decreasing rearrangement of the mesh refinement criterion function [29] yielding to a robust AMR method.

To test the efficiency of the overall method, and in particular of the automatic threshold technique, we have considered three classical large scale tsunami propagation test cases involving run-up, reflection and/or flooding and drying phenomena. As expected for the non dispersive test cases, we have obtained results in a very good agreement. In the case of weakly dispersive flows, we have obtained some small errors as pointed out before by several authors. However, the presented results can be improved using depth-averaged dispersive model such as the Boussinesq (weakly non linear) or the Green-Naghdi (fully non linear) one for instance.

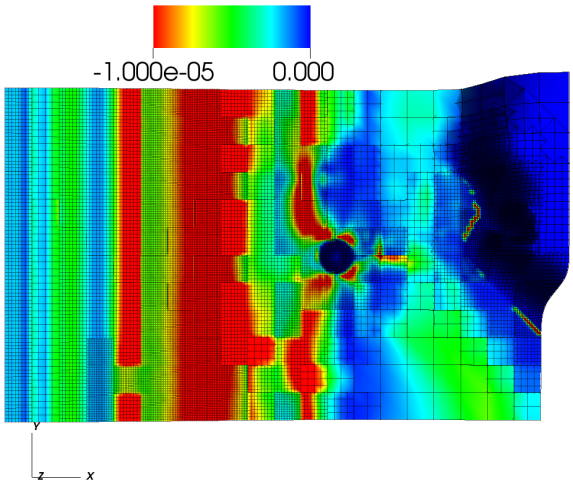


Fig. 29 Adaptive mesh : time $t = 11.25$s. The colours correspond to the numerical density of entropy production $\mathscr{S}$

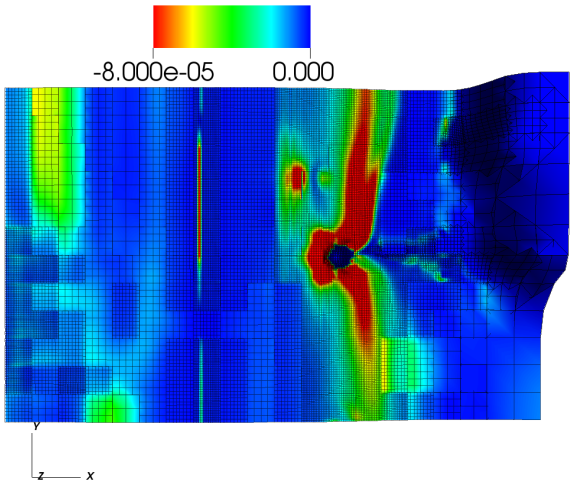


Fig. 30 Adaptive mesh : time $t = 13.25$s. The colours correspond to the numerical density of entropy production $\mathscr{S}$

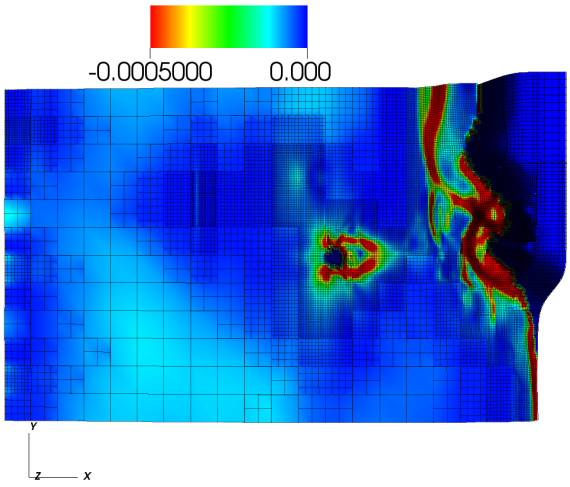


Fig. 31 Adaptive mesh : time $t = 16$s. The colours correspond to the numerical density of entropy production $\mathscr{S}$

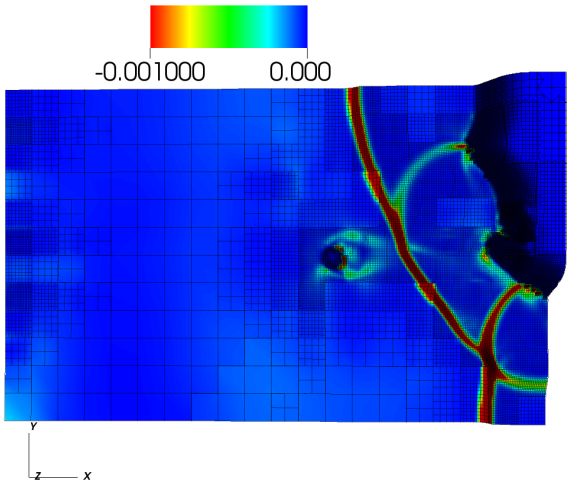
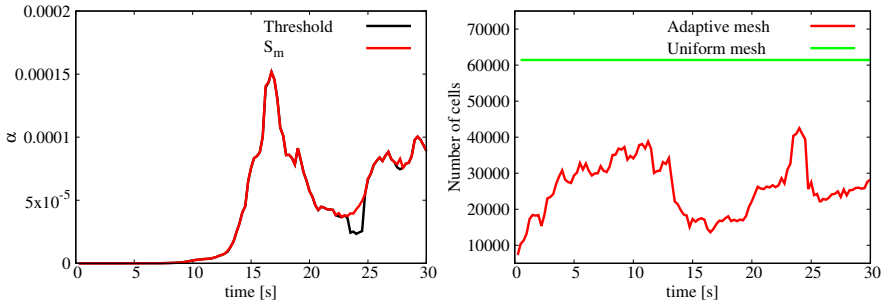


Fig. 32 Adaptive mesh : time $t = 17.5$s. The colours correspond to the numerical density of entropy production $\mathscr{S}$



(a) Threshold (b) Number of cells

Fig. 33 Time evolution of the mesh refinement threshold and the number of cells

As emphasized in [29], the use of the automatic threshold allows to localise almost all regions to be refined balancing the computational cost and the accuracy. The Automatic Mesh Refinement Threshold technique makes the AMR method almost *parameterless* and robust while the existing AMR methods are based on a fixed threshold and therefore are test case-dependent.

The overall efficiency and performance of the method has been shown through these three different test cases. Finally, let us highlight that the automatic thresholding method is independent of the numerical method used and the equations solved. Therefore it can be applied for a wide range of numerical applications.

Acknowledgments

This work is partially supported the Project MTM2011-29306-C01-01 from the MICINN (Spain) and the French national research project TANDEM (Tsunamis in the Atlantic and the English channel : definition of the effects through numerical modeling), the French government (Projets Investissement d'Avenir, agreement reference number ANR-11-RSNR-0023-01). Moreover, the authors wish to thank the referees for their remarks and the careful reading of the numerical method presented in this paper.

References

1. Altazin, T., Ersoy, M., Golay, F., Sous, D., Yushchenko, L.: Numerical investigation of BB-AMR scheme using entropy production as refinement criterion. International Journal of Computational Fluid Dynamics (2016). DOI 10.1080/10618562.2016.1194977. URL `https://hal.archives-ouvertes.fr/hal-01330654`

2. Arpaia, L., Ricchiuto, M.: r- adaptation for shallow water flows: conservation, well balancedness, efficiency. Computers & Fluids **160**, 175–203 (2018)
3. Audusse, E., Bouchut, F., Bristeau, M.O., Klein, R., Perthame, B.: A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. SIAM Journal on Scientific Computing **25**(6), 2050–2065 (2004)
4. Benzoni-Gavage, S., Serre, D.: Multi-dimensional hyperbolic partial differential equations: First-order Systems and Applications. Oxford University Press on Demand (2007)
5. Berger, M., Colella, P.: Local adaptive mesh refinement for shock hydrodynamics. Journal of Computational Physics **82**(1), 64 – 84 (1989). DOI http://dx.doi.org/10.1016/0021-9991(89)90035-1. URL http://www.sciencedirect.com/science/article/pii/0021999189900351
6. Berger, M., Oliger, J.: Adaptive mesh refinement for hyperbolic partial differential equations. J. Comp. Phys. **53**(3), 484–512 (1984)
7. Cockburn, B., Gremaud, P.: A priori error estimates for numerical methods for scalar conservation laws. Part II: Flux-splitting monotone schemes on irregular Cartesian grids. Math. Comp. **66**(218), 547–572 (1997)
8. Coupez, T., Hachem, E.: Solution of high-reynolds incompressible flow with stabilized finite element and adaptive anisotropic meshing. Computer Methods in Applied Mechanics and Engineering **267**(0), 65 – 85 (2013). DOI http://dx.doi.org/10.1016/j.cma.2013.08.004. URL http://www.sciencedirect.com/science/article/pii/S0045782513002077
9. Donat, R., Martí, M.C., Martínez-Gavara, A., Mulet, P.: Well-balanced adaptive mesh refinement for shallow water flows. Journal of Computational Physics **257**, 937–953 (2014)
10. Duran, A., Marche, F.: A discontinuous galerkin method for a new class of green–naghdi equations on simplicial unstructured meshes. Applied Mathematical Modelling **45**, 840–864 (2017)
11. 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). DOI 10.2478/s11533-013-0252-6. URL http://dx.doi.org/10.2478/s11533-013-0252-6
12. Ersoy, M., Golay, F., Yushchenko, L.: Adaptive multiscale scheme based on numerical density of entropy production for conservation laws. Central European Journal of Mathematics **11**(8), 1392–1415 (2013). DOI 10.2478/s11533-013-0252-6. URL http://dx.doi.org/10.2478/s11533-013-0252-6
13. Eymard, R., T., G., Herbin, R.: Finite volume methods. In: Handbook of numerical analysis, Vol. VII, Handb. Numer. Anal., VII, pp. 713–1020. North-Holland, Amsterdam (2000)
14. Fuster, D., Agbaglah, G., Josserand, C., Popinet, S., Zaleski, S.: Numerical simulation of droplets, bubbles and waves: state of the art. Fluid Dynamics Research **41**(6), 065,001 (2009)
15. Godlewski, E., Raviart, P.: Numerical approximation of hyperbolic systems of conservation laws, *Applied Mathematical Sciences*, vol. 118. Springer-Verlag, New York (1996)
16. Golay, F., Ersoy, M., Yushchenko, L., Sous, D.: Block-based adaptive mesh refinement scheme using numerical density of entropy production for three-dimensional two-fluid flows. International Journal of Computational Fluid Dynamics **29**(1), 67–81 (2015)
17. Hachem, E., Feghali, S., Codina, R., Coupez, T.: Immersed stress method for fluid structure interaction using anisotropic mesh adaptation. International Journal for Numerical Methods in Engineering **94**(9), 805–825 (2013). DOI 10.1002/nme.4481. URL http://dx.doi.org/10.1002/nme.4481
18. Houston, P., Mackenzie, J., Süli, E., Warnecke, G.: A posteriori error analysis for numerical approximations of Friedrichs systems. Numer. Math. **82**(3), 433–470 (1999)
19. Karni, S., Kurganov, A.: Local error analysis for approximate solutions of hyperbolic conservation laws. Adv. Comput. Math. **22**(1), 79–99 (2005)
20. Karni, S., Kurganov, A., Petrova, G.: A smoothness indicator for adaptive algorithms for hyperbolic systems. J. Comp. Phys. **178**(2), 323–341 (2002)

21. LeVeque, R.J., George, D.L.: High-resolution finite volume methods for the shallow water equations with bathymetry and dry states. In: Advanced numerical models for simulating tsunami waves and runup, pp. 43–73. World Scientific (2008)
22. LeVeque, R.J., George, D.L., Berger, M.J.: Tsunami modelling with adaptively refined finite volume methods. Acta Numerica **20**, 211–289 (2011)
23. Liu, P.L., Synolakis, C.: Advanced numerical models for simulating tsunami waves and runup, *Advances in Coastal and Ocean Engineering*, vol. 10, liu, philip lf and synolakis, costas edn. School of Civil and Environmental Engineering (2008)
24. Losasso, F., Gibou, F., Fedkiw, R.: Simulating water and smoke with an octree data structure. ACM Trans. Graph. **23**(3), 457–462 (2004). DOI 10.1145/1015706.1015745. URL http://doi.acm.org/10.1145/1015706.1015745
25. 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)
26. Ma, G., Shi, F., Kirby, J.T.: Shock-capturing non-hydrostatic model for fully dispersive surface wave processes. Ocean Modelling **43**, 22–35 (2012)
27. Min, C., Gibou, F.: A second order accurate level set method on non-graded adaptive cartesian grids. Journal of Computational Physics **225**(1), 300 – 321 (2007). DOI http://dx.doi.org/10.1016/j.jcp.2006.11.034. URL http://www.sciencedirect.com/science/article/pii/S0021999106005912
28. Osher, S., Sanders, R.: Numerical approximations to nonlinear conservation laws with locally varying time and space grids. Math. Comp. **41**(164), 321–336 (1983)
29. Pons, K., Ersoy, M.: Adaptive mesh refinement method. Part 1: Automatic thresholding based on a distribution function. submitted (2016)
30. Pons, K., Golay, F., Marcer, R.: Adaptive Mesh Refinement Method Applied to Shallow Water Model: A Mass Conservative Projection. In: Topical Problems of Fluid Mechanics 2017. Institute of Thermomechanics AS CR v.v.i., Prague, Czech Republic (2017). DOI 10.14311/TPFM.2017.032. URL https://hal.inria.fr/hal-01805238
31. Popinet, S.: Quadtree-adaptive tsunami modelling. Ocean Dynamics **61**(9), 1261–1285 (2011)
32. Popinet, S.: Adaptive modelling of long-distance wave propagation and fine-scale flooding during the tohoku tsunami. Nat. Hazards Earth Syst. Sci **12**(4), 1213–1227 (2012)
33. Popinet, S.: A quadtree-adaptive multigrid solver for the serre–green–naghdi equations. Journal of Computational Physics **302**, 336–358 (2015)
34. Puppo, G., Semplice, M.: Numerical entropy and adaptivity for finite volume schemes. Commun. Comput. Phys. **10**(5), 1132–1160 (2011). DOI 10.4208/cicp.250909.210111a. URL http://dx.doi.org/10.4208/cicp.250909.210111a
35. Roeber, V., Cheung, K.F.: Boussinesq-type model for energetic breaking waves in fringing reef environments. Coastal Engineering **70**, 1–20 (2012)
36. Roeber, V., Cheung, K.F., Kobayashi, M.H.: Shock-capturing boussinesq-type model for nearshore wave processes. Coastal Engineering **57**(4), 407–423 (2010)
37. Serre, D.: Systems of Conservation Laws 1: Hyperbolicity, entropies, shock waves. Cambridge University Press (1999)
38. Shi, F., Kirby, J.T., Harris, J.H., Geiman, J.D., Grilli, S.T.: A high-order adaptive time-stepping tvd solver for boussinesq modeling of breaking waves and coastal inundation. Ocean Modelling **43–44**(0), 36 – 51 (2012). DOI http://dx.doi.org/10.1016/j.ocemod.2011.12.004. URL http://www.sciencedirect.com/science/article/pii/S1463500311002010
39. Simeoni, C.: Remarks on the consistency of upwind source at interface schemes on nonuniform grids. SIAM J. Sci. Comput. **48**(1), 333–338 (2011)
40. 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)
41. Toro, E.F.: Riemann solvers and numerical methods for fluid dynamics, third edn. Springer-Verlag, Berlin (2009). DOI 10.1007/b79761. URL http://dx.doi.org/10.1007/b79761

42. Williamschen, M.J., Groth, C.: Parallel anisotropic block-based adaptive mesh refinement algorithm for three-dimensional flows. In: 21st AIAA Computational Fluid Dynamics Conference, June 24-27, San Diego, CA, pp. 1–22 (2013)
43. Yiu, K., Greaves, D., Cruz, S., Saalehi, A., Borthwick, A.: Quadtree grid generation: Information handling, boundary fitting and cfd applications. Computers & Fluids **25**(8), 759 – 769 (1996). DOI http://dx.doi.org/10.1016/S0045-7930(96)00029-1. URL http://www.sciencedirect.com/science/article/pii/S0045793096000291
44. Zhang, M., Wu, W.: 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 (2011). DOI http://dx.doi.org/10.1016/j.apor.2011.07.004. URL http://www.sciencedirect.com/science/article/pii/S0141118711000563
45. Zheng, Z., Groth, C.: Block-based adaptive mesh refinement finite-volume scheme for hybrid multi-block meshes. In: 7st conference on Computational Fluid Dynamics (ICCFD7), July 9-13, Hawaii, pp. 1–19 (2012)
46. Zijlema, M., Stelling, G., Smit, P.: Swash: An operational public domain code for simulating wave fields and rapidly varied flows in coastal waters. Coastal Engineering **58**(10), 992–1012 (2011)