Adaptive Mesh Refinement Method Applied to Shallow Water Model: A Mass Conservative Projection

Kévin Pons, Frederic Golay, R. Marcer

ADAPTIVE MESH REFINEMENT METHOD APPLIED TO SHALLOW WATER MODEL: A MASS CONSERVATIVE PROJECTION

K. Pons [1,2], **F. Golay** [2], **R. Marcer** [1]

[1] **Principia S.A.S., Zone Athélia 1, 215 voie Ariane, 13705 La Ciotat cedex, France**
[2] **IMATH, Université de Toulon, CS 60584, 83041 Toulon cedex 9, France**

Abstract

In the tsunami waves context, efficient numerical methods are necessary to simulate multi scales events. One way to reduce the computational cost is to use an adaptive mesh refinement method on unstructured meshes. This approach is used in this paper with a finite volume scheme solving the multi-dimensional Saint-Venant system. The adaptive mesh refinement method follows a block-based decomposition (called BB-AMR), which allows quick meshing and easy parallelization. One step of the AMR method is the so-called projection step where the new mesh values have to be defined from the old ones. For vertically integrated model the bathymetry variation during the projection step plays a crucial role on the mass conservation. To avoid large deficit or excess of mass a special attention is given to the projection step. Finally a 3D test case simulation is compared to experimental results to illustrated the quasi conservation of the mass with an appropriated projection method.

Keywords: Finite volume, Adaptive mesh refinement, Projection, Tsunami propagation, Saint-Venant model, Shallow water

1   Introduction

The most common tsunamis are a consequence of very large water volume displacement resulting in gravity waves. Depending of the tsunami source, the order of magnitude of a tsunami wave length is about several tens kilometers. As its horizontal length is much larger than the water depth, vertically integrated model are generally used for tsunami propagation. When dispersive effects are important, the Boussinesq or the Green Naghdi model are well adapted, but generally a quite simple model as the Saint-Venant model is used [7, 10]. It leads to an hyperbolic system of equations, where the numerical production of entropy can be used as mesh refinement indicator [6, 1]. One of the difficulty of AMR method is the projection step where the variable are defined from an old mesh to a new one. For vertically integrated model a special attention should be given to the bathymetry level to be mass conservative. In this paper we propose to describe these difficulties and to fix it with a quite simple method. Firstly the Saint-Venant model and the multi dimensional finite volume scheme are recalled. Then, the block-base adaptive mesh refinement procedure is presented in the framework of Saint-Venant model. A particular attention is given to the consideration of the bathymetry for the projection step. Finally, numerical simulations of a 3D test case involving a solitary wave propagation over a complex bathymetry with flooding and drying phenomena are compared with experimental data. The main goal of this application is to show the good mass conservation of the projection method used.

2   Saint-Venant model

The Saint-Venant model can be used to model several tsunami problems from the wave generation/propagation to the wave breaking. The Saint-Venant (or non linear shallow water) equations are derived from depth integrated Navier-Stokes equations (see e.g.[10]). This model assumes that horizontal length scale are much greater than vertical one which at first order involves constant velocity in the water height. After integration and simplification, the mass and momentum conservation lead to:

$$\frac{\partial h}{\partial t} + \operatorname{div}(h\boldsymbol{u}) = 0 \tag{1}$$

$$\frac{\partial h\boldsymbol{u}}{\partial t} + \operatorname{div}(h\boldsymbol{u} \otimes \boldsymbol{u}) + \frac{g}{2}\boldsymbol{\nabla}(h^2) = -gh\boldsymbol{\nabla}z \tag{2}$$

where $h(\boldsymbol{x}, t)$ stands for the water elevation above the topography $z(\boldsymbol{x})$ (Fig. 1), $\boldsymbol{u}(\boldsymbol{x}, t)$ represents the depth-averaged velocity vector, $\boldsymbol{x} \in \mathbb{R}^2$ is the space coordinate, $t > 0$ is the time, g is the gravitational constant $g \approx 9.81 \, m/s^2$.
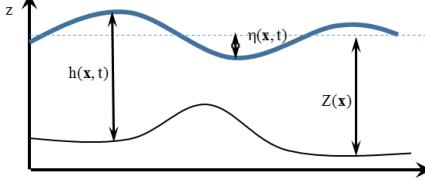

Figure 1: Notations

The Saint-Venant equations are a non-linear hyperbolic system of partial differential equations for which whose solutions may contain shock waves. For this model, the characteristic propagation speed in any direction, is $c = \sqrt{gh(\boldsymbol{x}, t)}$. As for any hyperbolic system, there exist an entropy and flux of entropy satisfying the Lax condition (or entropy inequality). For the Saint-Venant system, the entropy s and the flux ψ are defined by:

$$s = h \frac{\|\boldsymbol{u}\|^2}{2} + g \frac{h^2}{2} + ghz, \quad \psi = \left(s + g \frac{h^2}{2} \right) \boldsymbol{u} \tag{3}$$

3 Finite volume formulation

For the sake of completeness, we recall here the finite volume formulation of the Saint-Venant system. The source term is discretized using an upwind scheme at the cell interface through an hydrostatic reconstruction [2]. Using the conservative variable $w(\boldsymbol{x}, t) = \langle h, h\boldsymbol{u} \rangle^T$, an appropriate tensor f, the system (1) without source term can be written in the usual form:

$$\frac{\partial w}{\partial t} + \mathrm{div} \left(f(w) \right) = 0 \tag{4}$$

The computational domain $\Omega \subset \mathbb{R}^d$ is split into a set of control volumes, also referred as cells, $\Omega = \cup_k C_k$ of mesh size $|C_k|$. On a given cell C_k, approximating the unknown by it mean value $w_k(t) \simeq \int_{C_k} w(t, \boldsymbol{x}) \, dx \, / |C_k|$ at time t, and integrating (4) over each cell, we obtain:

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

where $\boldsymbol{n}_{k/a}$ denotes the unit normal vector on the boundary $\partial C_{k/a}$ between cells k and a.

Next, the flux is approximated such that

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

where $F \left(w_k(t), w_a(t), \boldsymbol{n}_{k/a} \right)$ is defined via the Godunov solver, i.e. it is computed with the exact solution of the 1D Riemann problem at the interface k/a with the states $w_k(t)$ and $w_a(t)$ (for further details see, for instance [14]).

The source term is discretized using an upwind scheme at the cell interface following the hydrostatic reconstruction [2]. Noting $\Delta Z_{k/a}$ the jump of topography across the cells k and a, firstly the numerical flux $F \left(w_k^*(t), w_a^*(t), \boldsymbol{n}_{k/a} \right)$ is evaluated using modified water depths: $h_k^* = \max(0, h_k - \max(0, \Delta Z_{k/a}))$ and $h_a^* = \max(0, h_a - \max(0, -\Delta Z_{k/a}))$. Secondly the term $\frac{g}{2}(h_k^2 - (h_k^*)^2) \boldsymbol{n}_{k/a}$ is added to the flux momentum at left hand side of the interface k/a and the term $\frac{g}{2}(h_a^2 - (h_a^*)^2) \boldsymbol{n}_{k/a}$ is added to the flux momentum at the right hand side of the interface k/a. The scheme is therefore well-balanced by construction, i.e., the still water steady states are exactly satisfied. The time

integration of the equation (4) can be achieved in a classical way by a Runge-Kutta scheme. By integrating the equation (4) 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 first order Euler's scheme is obtained:

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

In order to increase the accuracy, a second order Runge-Kutta method can be used as follows

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

where

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

4 Block Based Adaptive Mesh Refinement procedure (BB-AMR)

A dynamic block based adaptive mesh refinement method (BB-AMR) has been developed to balance fast computations and accurate simulations [1]. At first an efficient mesh refinement criterion has to be defined. Following previous work, we can notice that for all hyperbolic systems of conservation laws, as for the one we use here, a mathematical entropy can be derived (3), verifying the Lax's entropy condition.

$$C = \frac{\partial s(w)}{\partial t} + \nabla \cdot \psi(w) \leq 0 \ , \tag{5}$$

It has been numerically observed (and from theoretical considerations) that the production of the numerical density of entropy C is almost zero for smooth solution and non positive when the solution develops discontinuities. The equation (5) is a conservative expression that can be integrated into the system of conservation laws. Thus it uses the same discretization in space and time from which a numerical production of entropy can be derived. This quantity is used as a refinement criterion. Locally, the mesh is coarser or finer according to the value of its numerical production of entropy relative to the numerical production of entropy of the whole domain. More details can be found in [6]. As the choice of this threshold value is important, an improvement has been recently done in order to determine automatically this value [11].

Next a mesh refinement procedure has to be developed. For the Saint-Venant model, the quadtree meshing is widely used (see *e.g.* [16]). Very scalable and powerful schemes can be used, see *e.g.*[15]. But, the presence of a complex moving interface (composed of rarefaction, shocks and/or contacts) implies to re-mesh at each time step, which is obviously a costly process. Guided by the need to reach a relevant compromise between solution accuracy and computing speed, a Cartesian block-based mesh approach is introduced. If the mesh is not refined at each time step, the patch where the grid should be refined 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 hexahedral elements, called blocks. For simplicity raison, the Fig. 2(a) represents a regular quadrilateral mesh, but in fact it could be any unstructured hexahedral mesh (with 1 cell in z-direction). 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, the level of mesh refinement is automatically adapted if necessary (as for example for the blocks B5 and B2 Fig 2(b)). Thirdly, in order to balance the CPUs load, the cells of each block are re-distributed in a fixed number of domains according to the Cuthill-McKee numbering, see Fig. 2(c). This step will be improved later, using space-filling curve [3]. We do not focus on domain partitioning in this paper, CMK is simpler and easy to implement. Finally, as the mesh discretization is well defined, the unstructured mesh is built for each domain. Considering the old mesh variable values, the new ones are determined by an appropriated projection (see section 5 where this step is widely described). At the interface between two domains, ghost cells are created

9	10	11	12
5	6	7	8
1	2	3	4

(a) Initial grid domain

B9 1,1,0	B10 1,1,0	B11 1,1,0	B12 1,1,0
B5 1,1,0	B6 1,1,0	B7 1,1,0	B8 1,1,0
B1 1,1,2	B2 1,1,0	B3 1,1,0	B4 1,1,0

(b) Definition of the mesh n_x, n_y, l_r

(c) Domain decomposition

(d) Mesh generation
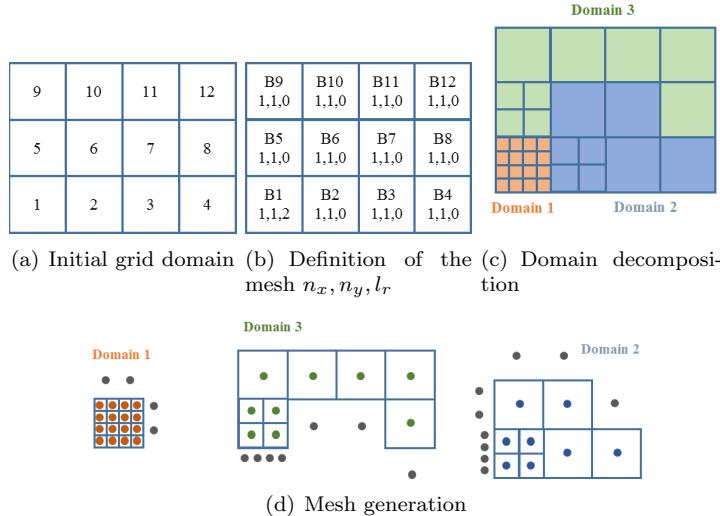
Figure 2: BB-AMR method

in order to transfer the data necessary 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 finally kept constant on a time interval, called AMR time-step, given by the smallest block (rather by the smallest cell) and the maximum velocity.

5   Projection methods

Each time the mesh is modified, the variables have to be defined on the new cells. This AMR step is called the projection step because knowing the variable field on a given mesh we have to determine the new field on a new mesh. The projection have to be mass, momentum and energetically conservative to respect the physic laws. Usually, for each cells split the values of the parent-cells are directly affected on each sub-cells. On the other hand, if some cells are coarsened, the child-cell (which concatenates the parent-cells) is defined by the mean values of the parent-cells (see [4] for more details). A more complex and accurate strategy to define new refined cells from coarse parents, is to interpolate them with the old gradient as done in [5]. However for vertically integrated models, the topography variation set some difficulties for these classical projection methods. Indeed the bathymetry does not act on the flow as a boundary condition (as for any 3D model) but acts directly in the model as a source term (see the equation (1)). Therefore if the bathymetry level is update on the new mesh, the projection have to consider not only the conservative variables but also the bathymetry level with a special attention. To use the more complex method quoted previously ([5]) in this context, Leveque et al. [8] propose to interpolate the free surface η instead of the conservative variable h. Far from the shore, the method is satisfying because mass and free surface are conserved. In this paper, the usual projection method is preferred for its simplicity and its reduced computational cost (no gradients need to be computed). The projection difficulties associated to vertically integrated models are detailed bellow and a simple solution is presented to assure a consistent projection for this method.

5.1   Difficulties with vertically integrated free surface model

In the special case of vertically integrated free surface model, the free surface level conservation is required to avoid the creation of spirituous waves. When the mesh changes the bathymetry level can evolve because a finer mesh allows to define a finer bathymetry (and vice versa). In the refinement case, as η is assumed constant to not produce spirituous waves, this bathymetry level

changing involves a water height changing $(h = \eta - z_{bottom})$, and therefore a possible deficit or excess of mass, see the Fig. 3 and for more details [13].


$$m_0 \neq m_1 + m_2$$
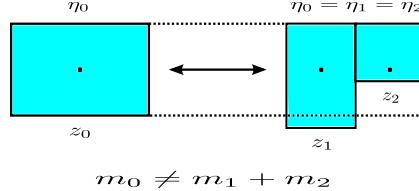
Figure 3: Illustration of mass conservation difficulty in the projection method for different grid levels. Here there is no direct link between the several bathymetry level. Each level comes from a proper interpolation/extrapolation of the initial bathymetry data file.

It is possible to neglect the mass loss/creation for some applications as done in [13] in the tsunami propagation context. However without much more additional difficulties it is possible to insure also the mass conservation if the bathymetry projection respect one requirement between each grid level. To formally expose this constraint in the framework of BB-AMR methods, let d denotes the dimension of the domain (1 or 2), 2^d the number of cells C_{0i} of surface $|C_{0i}|$ and z_{0i} the bathymetry which correspond on the coarser grid to one cell C_0 of surface $|C_0|$. Then, the mass conservation requirement, with constant density ρ, leads to:

$$\int_{oldmesh} \rho dV = \int_{newmesh} \rho dV \iff \rho |C_0| h_0 = \sum_{i=1}^{2^d} \rho |C_{0i}| h_{0i} \underset{h_i \geq 0}{\iff} |C_0|(\eta_0 - z_0) = \sum_{i=1}^{2^d} |C_{0i}|(\eta_{0i} - z_{0i})$$

$$\underset{\eta=cste}{\iff} z_0 = \frac{\sum_{i=1}^{2^d} |C_{0i}| z_{0i}}{|C_0|} \tag{6}$$

In the case of uniform one dimensional cartesian mesh we have therefore:

$$z_0 = \frac{z_{01} + z_{02}}{2} \tag{7}$$

This simple result shows that any cell refinement/coarsening under the constraint of free surface level and mass conservation should respect that grid bathymetry of rank n is the average of the bathymetry levels of rank $n + 1$. This function $(z_{01}, z_{02}) \longrightarrow \frac{z_{01}+z_{02}}{2}$ is not bijective. Therefore to respect this property in the refinement case, the finer possible bathymetry grid have to be the master and all other coarser grids have to be deduce from it. In this way the injective function will not be restrictive for the bathymetry accuracy.

5.2   Bathymetry discretization

In practice as the bathymetry data has a finite accuracy, it is often not necessary to define it on the smallest mesh grid. We choose a relevant mesh (with a close accuracy) and we interpolate the data on each cell center of the master grid as illustrated in Fig. 4(a),4(b). We are therefore able to define the bathymetry level on any coarser grid consistently in respecting the equation (6) from this master grid. To define the bathymetry on a more refined grid than the bathymetry accuracy (assumed relevant), we make the trivial choice $z_{0i} = z_0$ to extrapolate the data which obviously satisfied the equation (6).

(a) Interpolation of the bathymetry data on the master grid (b) Interpolation/Extrapolation of the sub-grid from the master grid values
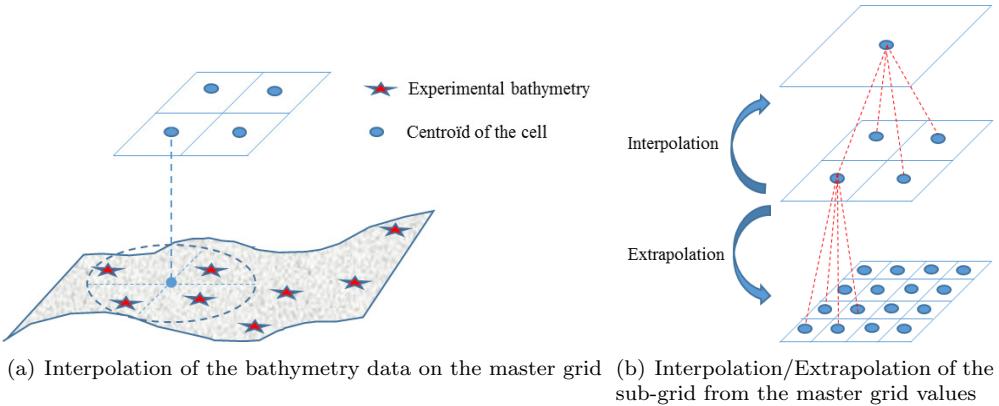
Figure 4: Building process of cell center bathymetry values

Remark 1 *It is important to quote that the interpolation procedure from a bathymetry data to an unstructured mesh can be an computational expensive process. Indeed, for each cell center of the grid the closer bathymetry data have to be found. In the tsunami wave context, the bathymetry file and the mesh can easily reach several millions of values. Fortunately, this process is only done one time (as a preprocessing task) because we only need to know the bathymetry level on the master grid (each other grid value being determined from this master level). The bathymetry values of the master grid are stored in order to be used each time the mesh is modified. In the block based AMR method, an efficient sorting process is to create a direct link between each block and the associated bathymetry values on the master level. Once is done, any sub-grid (coarser or finer) of the master one can be directly defined by the block ordered data with quick access.*

5.3 Refinement case

Once the bathymetry is known on the new mesh, the water height of refined cell is determined using its definition: $h_i = max(0, \eta_i - z_i)$ and the free surface conservation $\eta = cste$. It leads to :

$$h_{0i} = max(0, h_0 + z_0 - z_{0i}) \tag{8}$$

The "*max*" operator is very important when the mesh is adapted close to the shoreline for dry cells (see in section 5.5 the example on the Fig. 5 to update h_i from cell 0 to 2 for instance).

Once h_i is known, the momentum (for conservative variable) or the velocity (for non conservative variable) can be update. With an analogous reasoning of section 5.1 the momentum conservation leads to :

$$h_0 u_0 = \frac{\sum_{i=1}^{2^d} |C_{0i}| h_{0i} u_{0i}}{|C_0|} \tag{9}$$

When the mesh is refined, we have to determine u_{0i} from u_0. Therefore, as h_{0i} are computed by 8, a simple choice satisfying the equation (9) is

$$u_{0i} = u_0 \tag{10}$$

5.4 Coarsening case

To define the water height of coarse cell from its sub-cells, as the free surface η is not necessarily constant on the sub-cells, the mean value is taken. Considering the mass conservation and the equation (6) it involves that h is averaged.

$$h_0 = \frac{\sum_{i=1}^{2^d} |C_{0i}| h_{0i}}{|C_0|} \tag{11}$$

For the momentum, inversely to the refinement case, we have to determine u_0 from u_0i. The combination of the equations (9) and 11 leads to the following velocity:

$$u_0 = \frac{\sum_{i=1}^{2^d} |C_{0i}| h_{0i} u_{0i}}{|C_0| h_0} \tag{12}$$

5.5 Near shore projection

The previous reasoning is general but do not consider if a cell pass from a dry to a wet region or inversely. These cases can spoil the previous method because the shoreline accuracy can be modified by the mesh position.

A first particular case appears when dry and wet cell near the shoreline are coarsened by a cell with $z_0 > 0$, as illustrated in the Fig. 5. Then, using the equation (11), h_0 becomes positive which will create spirituous waves. In fact when a cell is dry, the definition of $\eta = h + z$ is not well-chosen. To get a better result without to modify the η definition, an other choice is to set :

$$z_{0i} > z_0 > 0 \; and \; h_{0i} = 0 \Longrightarrow \; h_0 = 0 \tag{13}$$

A second particular case appears when a dry cell is refined and when the bathymetry level passes from a positive value to a negative one, as illustrated in the Fig. 5. In this case, we can get a wrong water height value using the equation (8) which will result in a non conservation of the lake of rest (creation of spirituous wave). A trivial solution to fix it is to set:

$$z_0 > 0 \; and \; z_{0i} < 0 \; and \; h_0 = 0 \Longrightarrow \; h_{0i} = -z_{0i} \tag{14}$$
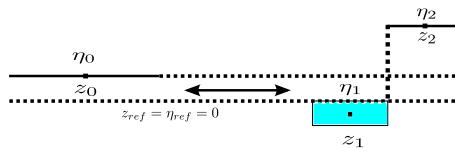

Figure 5: Illustration of the projection difficulties for shoreline cells

In practice the lost of mass conservation is quite rare and very localized which make the previous method satisfying in the long wave context. To be exactly conservative an easy solution is to force the mesh to not move on the shoreline. In this case, the particular cases previously presented are not realizable which make the mass conservation totally exact.

To finish, the new momentum value have also to be defined in the special cases close to the shoreline. For the first problematic case (see the equation (13)), as the water height is none, the velocity is obviously also none. For the second case (see the equation (14)), the volume of water of the new wet cell was not visible on the previous mesh. Therefore this water mass could not act on the system which ensures that the flow is stationary. So again in this case the velocity is set to zero.

6 Numerical application

In this section, a 3D experiment involving a solitary wave propagation over a complex bathymetry is simulated with the Saint-Venant model solved by the previously presented BB-AMR method. In a first time the mesh convergence of the AMR method is presented and compared to the experimental data and uniform mesh computation on one gauge. Then the converged AMR results are compared to several experimental results. Finally, the "quasi" conservative projection method presented in section 5 is illustrated for this test case.

In this experiment, a solitary wave of height of 39 cm at $x = 5$ m is produced with a piston-type wave-maker. For the numerical simulations, a first order Boussinesq solitary wave is used to initialize the flow (see for instance [12] for more details). In order to easily verify the mass

conservation of the simulations, all boundary conditions are assumed to be reflective. Several water height gauges (WG) are available as illustrated in the Fig. 6. For further information on this experiment, interested readers can found more details in [9].
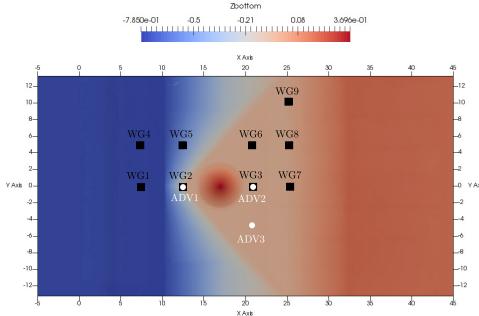

Figure 6: Experimental setup and bathymetry level representation. There are 9 water height gauges ($WG1-9$) and 3 velocity gauges ($ADV1-3$)

Before analyzing the numerical results, the mesh convergence is illustrated for one gauge (WG2) in the Fig. 7(a). Three AMR computations are performed with different refinement level leading to a mesh size variation $\triangle = [1, \{0.2, 0.1, 0.05, 0.025\}]$ m. An uniform mesh result (with a cell size equal to the minimum cell size of the converged AMR results $\triangle = 0.1$ m) is also plotted to show the relevancy of the AMR method. For equivalent results, the CPU gain for this test case is close to 2. For all computation, the CFL is set to 0.5, the time and space integration are of first order. For AMR simulation the re-meshing time step is set to 0.5 s (but it can be automatically chosen by a block-CFL, see for instance [1]). With a minimum cell size of $0.1m$ we obtain quasi converged results for AMR computation. Therefore in the following, only the AMR simulation with $\triangle_{min} = 0.1$ m is compared to other experimental results. The Fig. 7 shows the relevancy of the present computation with the Saint-Venant model. Some shifts in time and space can be observed on the Fig. 7(a) and 7(c) which are in part the consequence of the wave dispersion that is not taking into account by the present model. To fix these differences a more complex model like the Green-Naghdi (weakly dispersive shallow water) model should be used. Globally the results obtained with the simple Saint-Venant model for this test case are satisfying.

The Fig. 8 represents the mesh and the bathymetry adaptation during the conical island submersion. We can qualitatively see that important flow zones like steep gradient or flooding are well refined by the AMR method. The quantitative justification is obviously justified in the Fig. 7. Also we can see that even if the bathymetry is bad defined on the conical island in the beginning of the computation (see the Fig. 8(a)), when the wave comes up to this zone, the right bathymetry is well reconstructed from the master grid (see the Fig. 8(b)). This adaptive reconstruction is important for large scale simulation because there is no need to well define the coast (here the conical island and the reef) as soon the wave is not close to it. With the method presented in this paper the bathymetry is well defined only when the wave pass on it which allows to save an important number of cells.

The Fig. 7(d) represents the mass conservation of the projection method for the three finer AMR simulations. Despite many flooding phenomena in this test case, one can see a quasi constant water mass in the closed domain simulated and a good convergence rate for the several meshes.
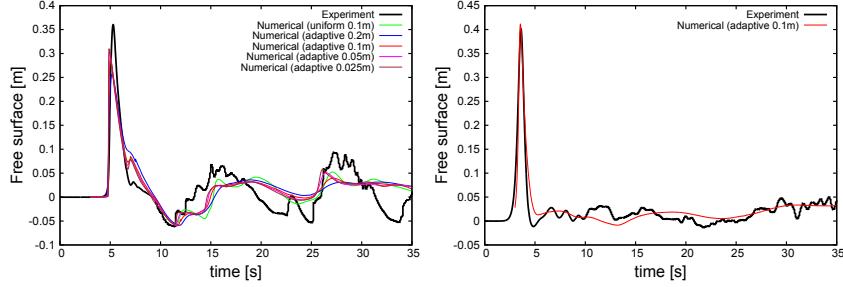
(a) Mesh convergence illustration for WG2
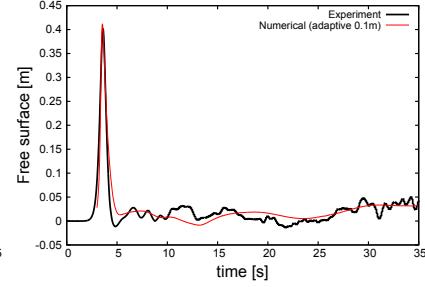
(b) Comparison between converged AMR results with experimental data for WG4
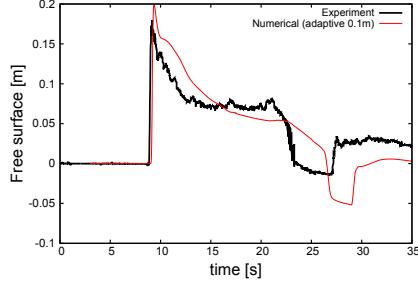
(c) Comparison between converged AMR results with experimental data for WG7
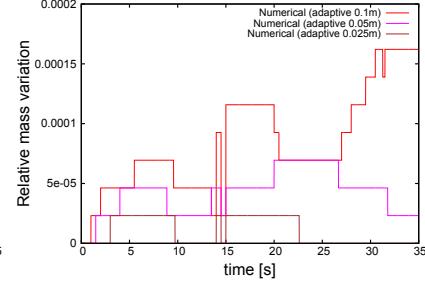
(d) Relative mass variation during the AMR simulation

Figure 7: Numerical vs experimental free surface level results and mass conservation
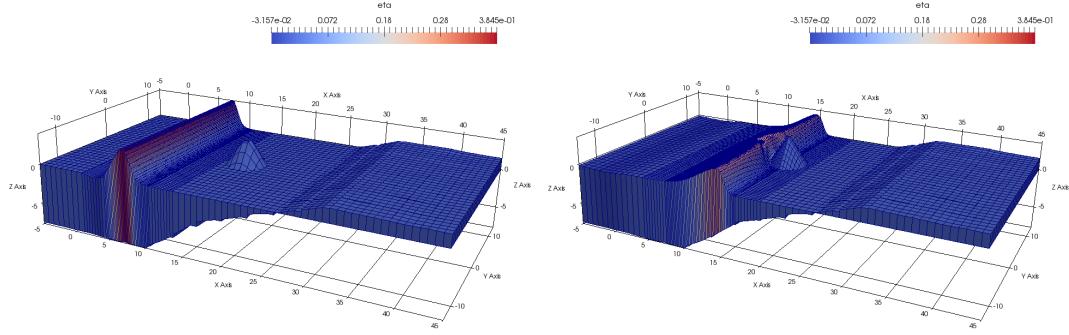

(a) $t = 3.5$ s.

(b) $t = 5.5$ s.

Figure 8: Illustration of the good mesh and bathymetry adaptation during the conical island submersion

7 Conclusion

We have presented a parallel finite volume scheme on adaptive unstructured mesh for the multi-dimensional Saint-Venant system. Because of the model formulation, a particular attention has been given to the projection step during mesh adaptation in order to be "quasi" mass conservative for any bathymetry. Finally, the simulation of a solitary wave propagation around a conical island validates this projection approach and shows the global efficiency of the AMR method.

Acknowledgment

References

[1] T. Altazin, M. Ersoy, F. Golay, D. Sous, and L. Yushchenko. Numerical investigation of bb-amr scheme using entropy production as refinement criterion. *International Journal of Computational Fluid Dynamics*, 30(3):256–271, 2016.

[2] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065, 2004.

[3] M. Bader. *Space-Filling Curves - An Introduction with Applications in Scientific Computing*, volume 9 of *Texts in Computational Science and Engineering*. Springer-Verlag, 2013.

[4] M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82(1):64 – 84, 1989.

[5] M. J. Berger and R. J. LeVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM Journal on Numerical Analysis*, 35(6):2298–2316, 1998.

[6] M. Ersoy, F. Golay, and L. Yushchenko. Adaptive multiscale scheme based on numerical density of entropy production for conservation laws. *Central European Journal of Mathematics*, 11(8):1392–1415, 2013.

[7] J.-F. Gerbeau and B. Perthame. Derivation of viscous Saint-Venant system for laminar shallow water; numerical validation. *Discrete Cont. Dyn. Syst. Ser. B*, 1(1):89–102, 2001.

[8] R. J. LeVeque, D. L. George, and M. J. Berger. Tsunami modelling with adaptively refined finite volume methods. *Acta Numerica*, 20:211–289, 2011.

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

[10] F. Marche. Derivation of a new two-dimensional viscous shallow water model with varying topography, bottom friction and capillary effects. *European Journal of Mechanic. B, Fluids*, 26(1):49–63, 2007.

[11] K. Pons and M. Ersoy. Adaptive mesh refinement method. Part 1: Automatic thresholding based on a distribution function. submitted, June 2016.

[12] K. Pons, M. Ersoy, F. Golay, and R. Marcer. Adaptive mesh refinement method. Part 2: Application to tsunamis propagation. submitted, July 2016.

[13] S. Popinet. Quadtree-adaptive tsunami modelling. *Ocean Dynamics*, 61(9):1261–1285, 2011.

[14] E. Toro. *Riemann solvers and numerical methods for fluid dynamics*. 2nd ed., Springer: New York, 1999.

[15] T. Weinzierl, M. Bader, K. Unterweger, and R. Wittmann. Block fusion on dynamically adaptive spacetree grids for shallow water waves. *Parallel Processing Letters*, 24(03):1441006, 2014.

[16] M. Zhang and W. Wu. 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.