International Journal of Computational Fluid Dynamics

Block-based adaptive mesh refinement scheme using numerical density of entropy production for three-dimensional two-fluid flows

F. Golay[a], M. Ersoy[a], L. Yushchenko[a] & D. Sous[b]

[a] Institut de mathématique de Toulon, University of Toulon, La Garde, France

[b] Mediterranean Institute of Oceanography, University of Toulon, CNRS/INSU,IRD, La Garde, France
Published online: 18 Feb 2015.

PLEASE SCROLL DOWN FOR ARTICLE

Block-based adaptive mesh refinement scheme using numerical density of entropy production for three-dimensional two-fluid flows

F. Golay[a,*], M. Ersoy[a], L. Yushchenko[a] and D. Sous[b]

[a]*Institut de mathématique de Toulon, University of Toulon, La Garde, France;* [b]*Mediterranean Institute of Oceanography, University of Toulon, CNRS/INSU,IRD, La Garde, France*

In this work, we present a fast and parallel finite volume scheme on unstructured meshes applied to complex fluid flow. The mathematical model is based on a three-dimensional compressible low Mach two-phase flows model, combined with a linearised 'artificial pressure' law. This hyperbolic system of conservation laws allows an explicit scheme, improved by a block-based adaptive mesh refinement scheme. Following a previous one-dimensional work, the useful numerical density of entropy production is used as mesh refinement criterion. Moreover, the computational time is preserved using a local time-stepping method. Finally, we show through several test cases the efficiency of the present scheme on two- and three-dimensional dam-break problems over an obstacle.

Keywords: low Mach model; dam-break problem; two-phase flow; artificial compressibility; finite volume; block-based mesh refinement; entropy production; local time stepping

1. Introduction

The understanding of wave hydrodynamics is of primary interest for countless ocean and naval engineering applications: dynamics of ships and floating structures, stability of offshore structures, mass, momentum and energy fluxes between ocean and atmosphere, coastal erosion and submersion processes, etc. Despite significant progress in numerical, experimental and theoretical works during the last decades, the wave dynamics remains a fairly open research field. Many difficulties arise when trying to describe the violent transformations observed during wave breaking and/or impact on rigid structures. The involved physical processes, such as splash-ups or gas pockets entrapment, are quite complex and can hardly be characterised by field or laboratory experiments or analytical approaches. Boosted by the continuous improvement of computer technology, the computational fluid dynamics simulations are expected to provide a unique insight, in terms of spatio-temporal resolution and controlled conditions, into such wave dynamics. Therefore, the numerical simulation of breaking and impacting waves is both an attractive research topic and a challenging task for coastal and environmental engineering.

The most comprehensive numerical approach, in terms of physical relevancy, would be the solving of the full set of Navier–Stokes (NS) equations for both air and water phases down to the Kolmogorov dissipation scale, together with non-linear dynamic and kinematic free surface boundary conditions and other boundary conditions to represent solid surfaces or open boundaries. Such an ambition is still out of reach for now in terms of computing power. Consequently, most of the studies based on the NS equations are generally performed in the two-dimensional case (see e.g. Hsu, Hsieh, and Hwang 2004; Hsiao and Lin 2010). Even if impressive results have been recently obtained using finite volume scheme (Lubin and Glockner 2013), Lattice-Boltzmann model (Banari et al. 2014), finite element model (Coupez and Hachem 2013; Hachem et al. 2013) or smoothed particle hydrodynamics model (Altomare et al. 2014), three-dimensional simulations still demand a significant effort in software development and mesh refinement technique with the use of powerful computers (see e.g. Kleefsman et al. 2005; Fuster et al. 2009; Abadie et al. 2010; Vincent et al. 2010). Turbulence models are often included in NS wave models whereas, despite of its importance in wave breaking and impact problems, the behaviour of the air-water micro-bubbles mixture is hardly taken into account (Plumerault et al. 2011).

Due to their much smaller requirements in cpu-time, simplified models based on potential theory or shallow water equations are also widely used in research and engineering applications (see e.g. Grilli, Guyenne, and Dias 2001; Dutykh, Poncet, and Dias 2011; Shi et al. 2012). Unfortunately, these models are not able to accurately describe the powerful rotational two-phase processes occurring during

---

*Corresponding author. Email: frederic.golay@univ-tln.fr

wave breaking or impact, such as splash-up, overturning and free surface reconnection.

The overall objective of our research work is to propose an 'intermediate' three-dimensional model aiming, on one hand, to be physically relevant in the context of highly dynamical and aerated flows associated with breaking and impacting waves and, on the other hand, to be less cpu-time consuming than the full three-dimensional NS equations. The presented model has been designed and continuously improved to this end (Helluy et al. 2005; Golay and Helluy 2007; Sambe et al. 2011), keeping in mind that an accurate capture of interface dynamics is a key issue for simulating wave transformation near breaking and after impacting. The assumption is made that, at least for the early stages, the physics of impacting flows is dominated by pressure forces and large-scale overturning motions rather than small-scale friction and dissipation processes. Viscous effects are thus neglected and the two-phase flows model can be based on the compressible Euler equations. An artificial linearised pressure law is used to compute low Mach flows (see for instance Chorin 1967) with a preconditioned physical pressure coefficient avoiding too constraining CFL (Courant–Friedrichs–Lewy) conditions. The obtained system is hyperbolic and, under some assumptions, the Riemann problem is well posed and can be solved using explicit parallel Godunov finite volume solver (as described in Golay and Helluy 2007). The physical relevancy of this compressible two-phase flow model has been successively tested on experimental and numerical test cases (Helluy et al. 2005), improved by the use of an isothermal model (Golay and Helluy 2007) and recently validated on breaking wave problem over a non-flat bottom with and without macro-roughness (Sambe et al. 2011).

The purpose of this paper is to present the implementation and to test the efficiency of a new block-based adaptive mesh refinement (BB-AMR) combined with an efficient parallel algorithm. Moreover, based on recent developments in one-dimensional case (Ersoy, Golay, and Yushchenko 2013), the use of the numerical density of entropy production as mesh refinement criterion is extended to our three-dimensional model. Following Ersoy, Golay, and Yushchenko (2013), the time-step constrain imposed by the mesh refinement is outweighed by the implementation of a local time-stepping method together with first- and second-order Adams–Bashforth time integration schemes.

The first section of the paper is dedicated to the presentation of the model governing equations (Section 2.1) and the finite volume approximation (Section 2.2). A particular attention is paid on the new BB-AMR scheme and the related adaptation of the space grid using the numerical density of entropy production. Combining both adaptive mesh refinement and local time-stepping, we construct a three-dimensional parallel solver on unstructured meshes. The model confrontation with experiments in both two-dimensional and three-dimensional dam-break problems is

presented in Section 3. Finally, concluding remarks and prospects are given in Section 4.

2. The mathematical and the numerical multi-fluid model

The hydrodynamics of breaking and impacting waves is extremely complex. As exposed in the introductory section, the most relevant approach would be to solve the full set of NS equations. Applied to three-dimensional cases and accounting for the time and space scales required by turbulence description, such computation is virtually inaccessible. The definition of a simplified mathematical and numerical model able to properly describe the physics of breaking and impacting waves at a decent cpu-time cost is a challenging problem which is the core of our research work. The purpose of this section is to briefly present the mathematical model with governing equations and finite volume approximation already described and validated in Helluy et al. (2005), Golay and Helluy (2007) and Sambe et al. (2011), and to detail the implementation of both the time integration and the three-dimensional extension of the adaptive multi-scale scheme introduced in Ersoy, Golay, and Yushchenko (2013) on unstructured meshed using a new BB-AMR.

2.1. Governing equation

We consider a compressible two-fluids flows problem in three space dimension. Viscosity, surface tension and heat conduction are neglected. The incompressibility condition is relaxed using a low Mach approach in order to lead to hyperbolic system of conservation laws. Thus, based on Golay and Helluy (2007), the following three-dimensional isothermal hyperbolic and compressible Euler equations system is applied to a mixture fluid of air and water:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \\ \dfrac{\partial (\rho \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = \rho \boldsymbol{g} \end{cases} \quad (1)$$

where the unknowns depend on spatial coordinates (x, y, z) and time t. The unknowns are the density ρ, the three components of the velocity $\boldsymbol{u} = (u, v, w)$, the pressure p. Here, $\boldsymbol{g}$ stands for gravitational acceleration.

Air and water fractions within the mixture are defined by the volume fraction function $\varphi \in [0, 1]$ ($\varphi = 0$ in the water, and $\varphi = 1$ in the air). With this definition of φ, the pressure of the two-phase flow problem is a function of the density ρ and the volume fraction φ, where φ solves the following non-conservative transport equation:

$$\frac{\partial \varphi}{\partial t} + \boldsymbol{u} \cdot \nabla \varphi = 0 . \quad (2)$$

It is usually admitted that a flow is incompressible if the Mach number $M = \|\boldsymbol{u}\| / c$ is lower than 0.1 (c is the sound speed), keeping in mind that the real (physical)

Mach number is generally much smaller (of the order of $1/400 \sim 1/1600$). In particular, this is constraining for explicit finite volume solver in which the time step Δt needs to satisfy a CFL condition. Also note that the numerical scheme efficiency is expected to decrease with the Mach number. Therefore, an artificial pressure law (isothermal equation of state) is used to close the system:

$$p = c_0^2(\rho - (\varphi\rho_A + (1 - \varphi)\rho_W)) + p_0. \qquad (3)$$

In this expression, ρ_A and ρ_W stand for air and water densities, respectively, c_0 is the artificial sound speed (defined below) and p_0 a reference pressure. For further details about the EOS choice, the reader is referred to Golay and Helluy (2007). The value of c_0 is chosen as a compromise between the limits of compressible effects, the rate of numerical diffusion and a reasonable CFL constraint. The optimal choice for the artificial sound speed is still an open problem and would require a large amount of comparative test cases in a wide range of hydrodynamical conditions. In the present context, i.e. for flow velocity of the order of 1 m/s, an optimised value $c_0 = 20$ m/s is used. Further improvements such as variable artificial sound speed can be considered. Finally, it is emphasised that in the boundary mixture region $0 < \varphi < 1$ related to numerical diffusion processes, the proposed pressure law has no physical meanings.

For the ease of reference, Equation (1) are written into a more compact form:

$$\frac{\partial \boldsymbol{w}(t)}{\partial t} + \nabla \cdot \boldsymbol{f}(t, \boldsymbol{w}) = \boldsymbol{G} \qquad (4)$$

where $\boldsymbol{w}$, $\boldsymbol{f}$, $\boldsymbol{G}$ stands respectively for conservative variables, flux and source:

$$\boldsymbol{w} = \begin{pmatrix} \rho \\ \rho\boldsymbol{u} \end{pmatrix}, \quad \boldsymbol{f} = \begin{pmatrix} \rho\boldsymbol{u} \\ \rho\boldsymbol{u} \otimes \boldsymbol{u} + pI \end{pmatrix} \quad \boldsymbol{G} = \begin{pmatrix} 0 \\ \rho\boldsymbol{g} \end{pmatrix}.$$

Here I stands for the 3×3 identity matrix.

2.2. Finite volume approximation

In this section, we recall the well-known semi-discrete finite volume approximation of Equation (4). The system is solved in $\mathbb{R}^3$ and the source term is omitted for the sake of simplicity. The computational domain $\Omega \subset \mathbb{R}^3$ is split into a set of control volumes, also referred as cells, $\Omega = \cup_k C_k$. We start by integrating Equation (4), by means of Green formula, on a cell C_k:

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

where $\boldsymbol{n}_{k/a}$ denotes the unit normal vector on the boundary $\partial C_{k/a}$. Then, setting

$$\boldsymbol{w}_k(t) \approx \frac{1}{|C_k|} \int_{C_k} \boldsymbol{w}(\boldsymbol{x}, t) \, d\Omega$$

where $|C_k|$ stands for the volume of the cell C_k, we approximate (5) by

$$\frac{\partial \boldsymbol{w}_k(t)}{\partial t} + \sum_a \boldsymbol{F}\left(\boldsymbol{w}_k(t), \boldsymbol{w}_a(t); \boldsymbol{n}_{k/a}\right) = 0, \qquad (6)$$

where the numerical flux

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

is defined by the Godunov solver. More precisely, the numerical flux $\boldsymbol{F}\left(\boldsymbol{w}_k(t), \boldsymbol{w}_a(t); \boldsymbol{n}_{k/a}\right)$, noted for the sake of simplicity $\boldsymbol{F}_{k/a}(t)$, is computed with the exact solution of the Riemann problem at the interface k/a (for further details, see for instance, Toro (1999) or Golay and Helluy (2007)). Pressure oscillations observed in the multi-fluid case are avoided using Abgrall's method (Abgrall 1996) leading to a non-conservative discretisation of Equation (2). It is computed through the contact discontinuity velocity of the Riemann problems u^* solved at the interface k/a. As a consequence, the numerical flux for Equation (2) simply reads $\min(u^*, 0)(\phi_a - \phi_k)$.

Alternatively, a full conservative scheme with another pressure law can be used (Allaire, Clerc, and Kokh 2002) but the less time consuming approach of Abgrall is retained here. The accurate interface capture is ensured by the use of a second-order approximation Monotonic Upstream-Centered Scheme for Conservation Laws (MUSCL) with Barth limiter. Further details can be found in Golay and Helluy (2007).

2.3. Time integration and local time-stepping method

This section is dedicated to the presentation of the principles of the Adams–Bashforth time integration and its implementation in our model. This approach has been shown to significantly decrease the computational cost compared to classical Runge–Kutta time integration which needs intermediate computations. The Runge–Kutta approach is deeply related to the size of the smallest control volume and therefore the time step is severely constrained when the mesh is refined. By contrast, even if the Adams–Bashforth is known to be less stable and less accurate, it can be easily handled in the framework of local time stepping (as described below) to save computational time (see for instance, Altmann et al. 2009 or Ersoy, Golay, and Yushchenko 2013).

Equation (6) is first integrated over the time interval $[0, T]$ subdivided such that $t_{n+1} = t_n + \delta t_n$ with δt_n the time step.

The Adams–Bashforth method of order m consists in replacing the numerical flux of Equation (6) by a Lagrange polynomial interpolation of the same order (Hairer, Nørsett, and Wanner 1993). Note that this explicit approximation is built with the fluxes previously computed and stored and

then the integration of Equation (4), in the case of interest, i.e. the second-order Adams–Bashforth, leads to

$$
\begin{aligned}
\boldsymbol{w}_k(t_{n+1}) = {}& \boldsymbol{w}_k(t_n) \\
& - \sum_a \left(\frac{\delta t_n}{h_k} \, \boldsymbol{F}_k(t_n) - \frac{\delta t_n^2}{2\delta t_{n-1} \, h_k} \left(\boldsymbol{F}_k(t_n) - \boldsymbol{F}_k(t_{n-1}) \right) \right),
\end{aligned}
\tag{7}
$$

where $h_k = \frac{|C_k|}{\sum_a |\partial C_{k/a}|}$. The main advantage of the Adams–Bashforth method is to avoid the computation of intermediate fluxes and thus facilitate a local time-stepping approach (see Ersoy, Golay, and Yushchenko 2013 for details).

2.4. Mesh refinement criterion and BB-AMR scheme

2.4.1. Entropy production as a mesh refinement parameter

As pointed out in Ersoy, Golay, and Yushchenko (2013), many works are based on a-posteriori error estimates, which are constructed from mathematical arguments for mesh refinement (Zhang, Trépanier, and Camarero 2000). But paradoxically, to our knowledge, very few works use a refinement criterion based on physical consideration, such as entropy (Golay 2009). Based on this latter as a mesh refinement parameter, Ersoy, Golay, and Yushchenko (2013) proposed an efficient adaptive numerical scheme. The aim of this work is to detail the extension of this scheme to three-dimensional cases and to demonstrate its efficiency.

It is well known that solving equations system (4) 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 system (4) with an entropy inequality of the form:

$$
\frac{\partial s(\boldsymbol{w})}{\partial t} + \nabla \cdot \psi(s(\boldsymbol{w})) \leq 0 \,,
\tag{8}
$$

where (s, ψ) stands for a convex entropy–entropy flux pair satisfying the relation

$$
(\nabla_w \boldsymbol{\psi}(s(\boldsymbol{w})))^t = (\nabla_w s(\boldsymbol{w}))^t \; D_{\boldsymbol{w}} f(\boldsymbol{w}).
$$

This inequality in Equation (8) 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. Therefore, the entropy production can be seen as a 'smoothness indicator' at the numerical level.

To this end and to be consistent with the discretisation of Equation (7), a discrete version of the entropy production, called *numerical density of entropy production*, is defined as follows:

$$
\begin{aligned}
S_k^n = {}& \frac{s_k^{n+1} - s_k^n}{\Delta t_n} + \sum_a \left(\frac{\delta t_n}{h_k} \boldsymbol{\psi}_{k/a}(t_n) \right. \\
& \left. + \frac{\delta t_n^2}{2\delta t_{n-1} h_k} \left(\boldsymbol{\psi}_{k/a}(t_n) - \boldsymbol{\psi}_{k/a}(t_{n-1}) \right) \right) \cdot \boldsymbol{n}_{k/a}. \quad
\end{aligned}
\tag{9}
$$

In this expression, $\boldsymbol{\psi}_{k/a}$ represents the entropy flux calculated from the resolution of the Riemann problem at the interface of cells k and a.

From a theoretical viewpoint, except during a shock, the numerical density of entropy production given by (9) is zero. But, at the numerical level, one can observe that it does not vanish and hence can be used as mesh refinement criterion. More precisely, Ersoy, Golay, and Yushchenko (2013) have shown that, in the case of one-dimensional gas dynamics equations for ideal gas, the support of relative error coincides with the support of numerical density of entropy production. As a consequence, the mesh is automatically refined only in regions where errors are identified.

The local entropy production must be compared to the total entropy $\bar{S} = \frac{1}{|\Omega|} \sum_k S_k^n$. Two coefficients $0 \leqslant \alpha_{\min} \leqslant \alpha_{\max} \leqslant 1$ 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 $S_k^n > \bar{S}\alpha_{\max}$, the mesh is refined and split and,
- if $S_k^n < \bar{S}\alpha_{\min}$ the mesh is coarsened whenever it is possible following the rule defined hereinafter.

Remark 1: The threshold parameters $\alpha_{\max}$ and $\alpha_{\min}$ are determined empirically for each problem to get a good compromise between computational cost and accuracy as done previously in the one-dimensional framework (Ersoy, Golay, and Yushchenko 2013). More precisely, $\alpha_{\min}$ and $\alpha_{\max}$ allow to set a percentage of mesh refinement and mesh coarsening with respect to the quantity $\bar{S}$. It is not surprising that these settings will deteriorate or improve the accuracy of the numerical solution. For instance, the more $\alpha_{\min}$ and $\alpha_{\max}$ are small, the more accurate are the results to the expense of the computational time.

Finally, in the present case, for the two-fluid model, the expression of entropy and entropy flux are

$$
s = \frac{1}{2}\rho\boldsymbol{u}^2 + c_0^2\rho\ln\rho - c_0^2(\rho_W - \rho_A)\varphi,
$$

$$
\boldsymbol{\psi} = \left(\frac{1}{2}\rho\boldsymbol{u}^2 + c_0^2\rho(\ln\rho + 1) \right)\boldsymbol{u}.
$$

2.4.2. BB-AMR scheme

Keeping in mind the following requirements: parallel treatment, fast computation and hierarchical grid (mesh level cannot exceed 2 for stability reasons as pointed out in Ersoy, Golay, and Yushchenko 2013), the main and challenging difficulty is to develop a suitable mesh refinement tool which is related on the design of software data structures and numerical scheme.

Moreover, as pointed out by several authors, the extension toward three-dimensional case is not an easy task. Interesting works have been presented for two-dimensional Cartesian grid or quad-tree (Berger and Colella 1989; Yiu et al. 1996; Min and Gibou 2007; Zhang and Wu 2011), octree for three-dimensional simulations (Losasso, Gibou, and Fedkiw 2004; Fuster et al. 2009) and anisotropic AMR (Coupez and Hachem 2013; Hachem et al. 2013).

This study aims to generalise the scheme proposed by Ersoy, Golay, and Yushchenko (2013) with a finite volume solver described in the previous section. This naturally leads to quad-tree meshing in two-dimensional problem and octree meshing in three-dimensional problem.

The presence of a complex moving air–water interface implies to re-mesh at each time step, which is clearly time consuming. Keeping in mind our willingness to find a relevant compromise between the contradictory aims of physical relevancy and numerical performance, we introduce a Cartesian block-based mesh approach (somehow like in Williamschen and Groth 2013; Zheng and Groth 2012). The computational domain is split in several blocks. In practice, each block corresponds to the initial unstructured mesh composed of hexahedral cells. According to the mesh refinement procedure (with respect to S_k^n, $\bar{S}$, the mesh refinement level N), each block is sub-divided in a Cartesian way ($2^{N-1}n_x$, $2^{N-1}n_y$, $2^{N-1}n_z$), where (n_x, n_y, n_z) stands for the initial block discretisation.

As done in Ersoy, Golay, and Yushchenko (2013), for each refined cells (or blocks), averaged values at time t^n are projected on each sub-cell and fluxes are computed as simply as possible to avoid heavy computation. Consequently, the level of two adjacent blocks cannot exceed 2 to avoid oscillations (see Ersoy, Golay, and Yushchenko 2013 for further details). Thus, the interface defining two blocks most of the time is a non-conforming one (as displayed in Figure 2(b)).

Next, in order to spare proportionally the balanced distribution of cpu-load, the cells of each block are redistributed in a fixed number of domains according to the Cuthill–McKee numbering. The number of domain being fixed, each domain are loaded in a given message passing interface (MPI) process. The procedure is certainly not optimal but the synchronisation time between each domain are almost the same, hence overall more efficient. Finally, the renumbering and re-meshing being expensive, the mesh is kept constant on a time interval, called AMR time step, given by *the smallest block* (and not by the smallest cell)

and the maximum velocity. This procedure provides a good compromise between computational time and accuracy.

Finally, let us illustrate the algorithm to define unstructured meshes using the BB-AMR scheme in the two-dimensional case (i.e. $n_z = 1$ always). During the refinement process, the number of blocks and domains are kept always constant. As displayed in Figure 1(a), we consider an initial mesh composed of nine blocks, assumed to be of level 1 (the coarsest level with $n_x = n_y = 1$), and decomposed on three domains according to a Cuthill–McKee numbering of the blocks ($3 + 3 + 3$ internal cells). For example, the first domain (blue) is composed of three internal cells and three adjacent 'ghost' cells in order to share to send/receive information between domains. After some computations, at the AMR-time T_1, let us suppose that blocks 1 and 4 have to be refined. As displayed on Figure 1(b), the blocks distribution into the domains has been modified ($5 + 5 + 5$ internal cells). One can observe that the new mesh is a non-conforming mesh. For example, the cell belonging to the block 2 is bordered by seven faces. If, at the AMR-time T_2, only block 1 exceeds the mesh refinement criterion, it has to be refined. But, as the level of two adjacent blocks cannot exceed 2 in order to avoid oscillations, block 2 must also be refined. As displayed in Figure 1(c), the blocks distribution has been modified ($16 + 7 + 7$ internal cells). The distribution is not well balanced because we distribute the blocks and not the cells for sake of simplicity. This drawback is reduced as the number of blocks is much bigger than the number of domains.

The three-dimensional case is dealt in the same way. For illustration purpose, Figure 2 shows an example of three-dimensional block-based mesh decomposition ($64 + 39 + 39$ internal cells) with three domains and 27 blocks, where the first block in the corner is of mesh refinement level 3.

3. Numerical results

The model is here tested with classical two- and three-dimensional test cases on dam-break problems over an obstacle. The two-dimensional case is used to validate the mesh decomposition strategy while the three-dimensional case quantitatively tests the overall performance of the presented model with and without BB-AMR scheme confronted to experimental and state-of-the-art numerical results.

3.1. A two-dimensional dam-break problem

This test case was initially introduced to study a particle method for simulating incompressible viscous flow with a particular focus on water dispersion and droplets after wave impact over an obstacle (Koshizuka, Tamako, and Oka 1995). The experimental data obtained by Koshizuka, Tamako, and Oka (1995) is here used to test the ability of

(a) Initial block and initial
3 domains

(b) AMR time step T_1:
block 1 and 4 exceed the
mesh refinement criterion

(c) AMR time step T_2 :
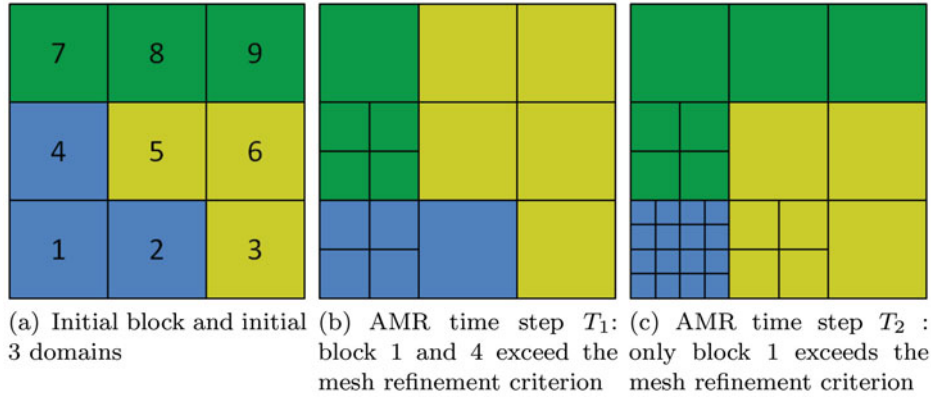only block 1 exceeds the
mesh refinement criterion

Figure 1.   Example of two-dimensional BB-AMR with three domains and nine blocks. add to legend (a): (domain 1: 1,2,4; domain 2: 3,5,6; domain 3: 7,8,9) add to legend (b): (domain 1: 1,2; domain 2: 3,5,6,8,9; domain 3: 4,7) add to legend (c): (domain 1: ; domain 2: 2,3,5,6; domain 3: 4,7,8,9)

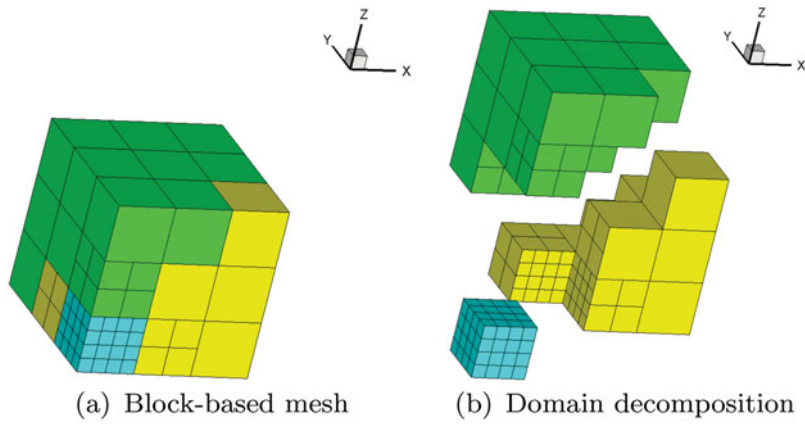(a) Block-based mesh

(b) Domain decomposition

Figure 2.   Example of three-dimensional block-based mesh with three domains and 27 blocks.

our adaptive strategy to represent the complex structure of the air–water interface.

The test case configuration is shown in Figure 3. The domain size is $4L \times 2L$ with $L = 14.6$ cm. A water column ($2L$ high and L wide) and a rigid obstacle ($2h$ high and h wide, with $h=2.4$ cm) are initially located at the
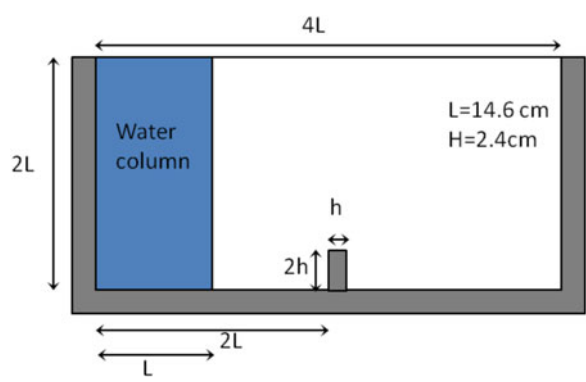

Figure 3.   Collapse of water column Koshizuka, Tamako, and Oka (1995).

left and centre of the domain, respectively. Air and water are at rest at the start of the experiment. The selected numerical parameters are CFL = 0.8, simulation time = 1.5 (s), 321 blocks, 120 cores, 321 domains or MPI process, 5 as the maximum level of mesh refinement, α_{max} (mesh refinement parameter) = 0.2, α_{min} (mesh coarsening parameter)= 0.02. The simulation time is divided in 300 AMR time intervals. Symmetry boundary conditions are imposed: zero normal velocity and zero normal gradients of all other variables at the symmetry plane.

Figure 4 represents the comparison between experimental and numerical snapshots during the dam-break process on the left and right columns, respectively. The selected times are $t = 0.2, 0.3, 0.4, 0.5$ and 1 s. At $t = 0.2$ s, the wave has already impacted the obstacle and a thick water jet is ejected. At $t = 0.3$ and 0.4 s, the jet extends and hits the right wall, inducing entrapment of a large air pocket, which remains significant during the whole backwash and retreat process observed at $t = 0.5$ and 1 s.

The macroscopic features of such complex hydrodynamics, such as impact times against the obstacle and wall
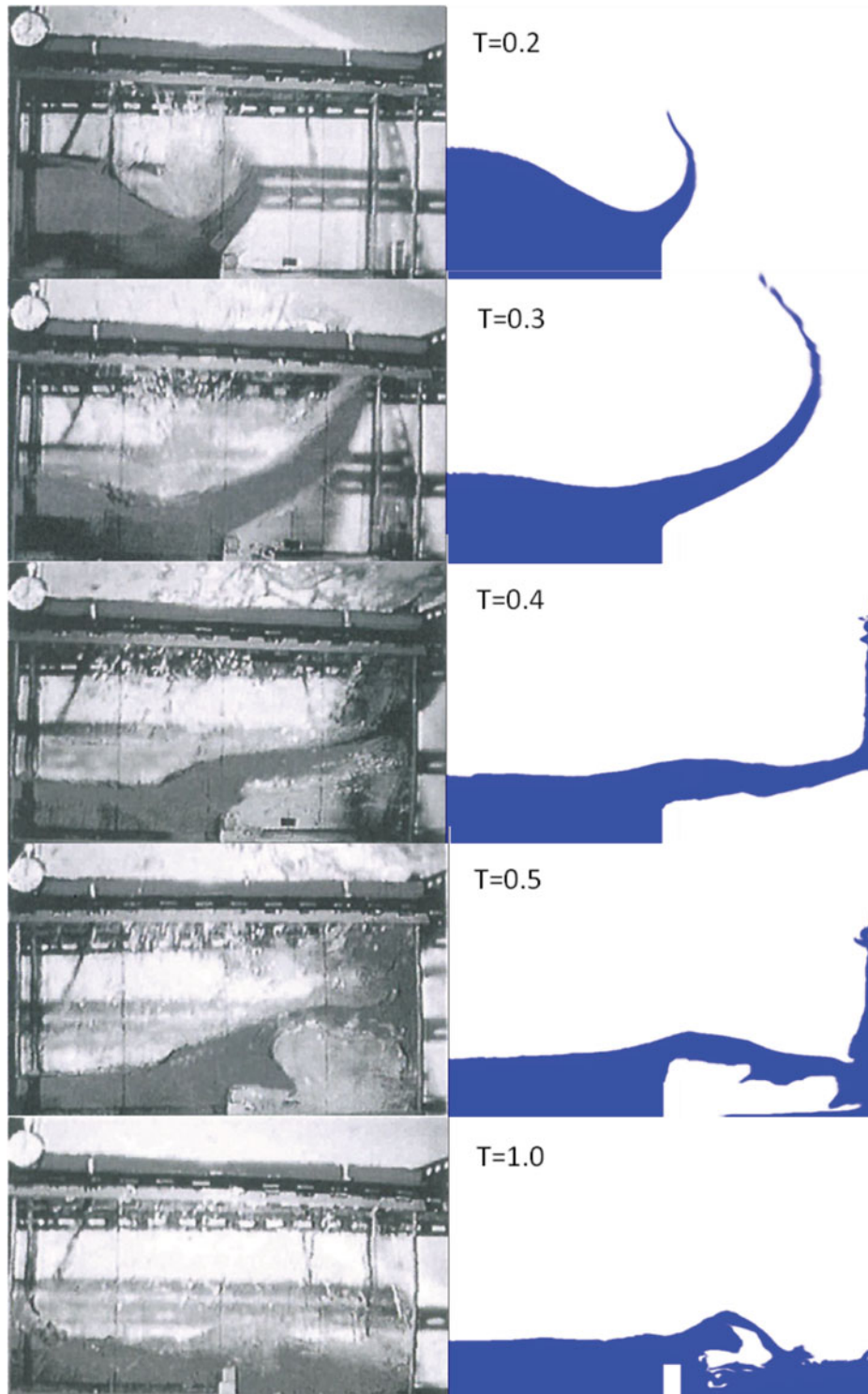
Figure 4.   Two-dimensional dam-break problem, confrontation with experiments: experimental simulation Koshizuka, Tamako, and Oka (1995) (left) – numerical simulation (right).

or jet shape and thickness, are rather well reproduced by the model. Moreover, even though our model was not designed to capture small-scale interface splitting and reconnection

processes, some of the droplets or packets of droplets are captured as displayed on the right column of Figure 4 at $t = 0.4, 0.5$ and 1 s. Increasing the level of refinement,
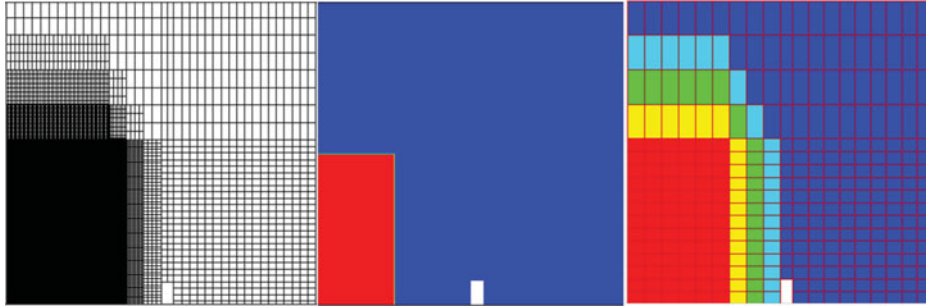
Figure 5. Initial configuration for two-dimensional dam-break problem: mesh (left), density (centre), mesh refinement level (1–5) per block (right).

leading to more cells, will improve the accuracy of the numerical description of such processes, but this is not the aim of this work.

The initial configuration for the numerical test case is shown in Figure 5. Five levels of mesh refinement are used (see Figure 5, right part). The mesh is initially denser in the areas of interest, i.e. the water phase and around the interface and then progressively coarsens in the air phase when moving away from the water column. Figures 6 and 7 represent the behaviour of the dynamic mesh driven by the numerical density of entropy production at time $t = 0.2$ and $0.4s$. The mesh is observed to remain and/or become the finest in the most active regions of the flow, i.e. interfacial, impact or sheared areas. This overall compared evolution of the collapsing water column with experiments demonstrates the relevancy of our refinement criterion, in particular when tracking the air–water interface. Note that the number of cells varies from 70,000 and 100,000 during all the simulation for an elapsed computing time about

5 hours. The present case shows the performance of the automatic mesh refinement strategy. Here, the numerical scheme is parallel and we have used more MPI process than cores based on a simple meshing strategy, one block is put in one domain. Such a simple procedure is of course rather time consuming as the load of the processors is not well balanced because of the required synchronisation at each time step. As explained previously and illustrated in the following three-dimensional cases, this synchronisation issue can be addressed by the use of BB-AMR scheme.

3.2. A three-dimensional dam-break problem without AMR block-based mesh decomposition

In order to test the efficiency of the model, we focus on the well-known three-dimensional dam-break experiments carried out by Kleefsman et al. (2005). This test case being
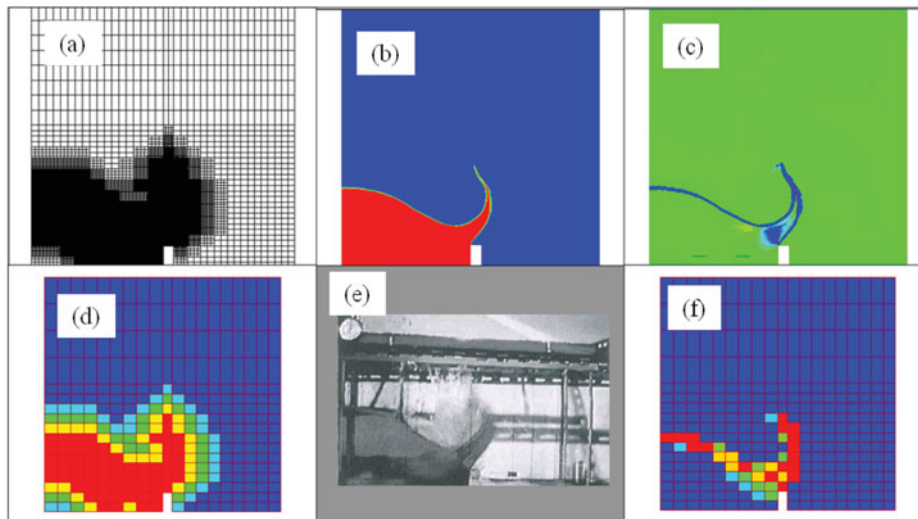

Figure 6. Two-dimensional dam-break problem at t=0.2 s. (a) Mesh; (b) density; (c) density of numerical entropy production; (d) mesh refinement level per block (1–5); (e) experiment Koshizuka, Tamako, and Oka (1995); (f) mesh refinement criterion per block.

Figure 7.   Two-dimensional dam-break problem at t=0.4 s. (a) Mesh; (b) density; (c) density of numerical entropy production; (d) mesh refinement level per block (1–5); (e) experiment Koshizuka, Tamako, and Oka (1995); (f) mesh refinement criterion per block.
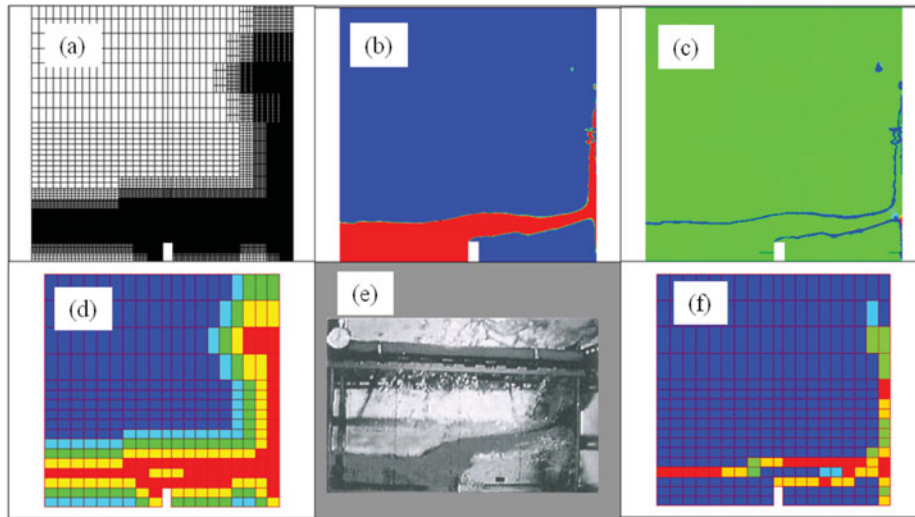
both highly dynamic and strongly three-dimensional as well as well documented by pressure sensors and wave gauges, it has been widely used to evaluate the performance of several numerical models such as volume of fluid (VOF) method (Kleefsman et al. 2005), Smoothed Particle Hydrodynamics (SPH) method (Lee et al. 2010) or Eulerian-Lagrangian method (Vincent et al. 2010).

The initial configuration of the problem as well as sensors location are shown in Figure 8. The setup consists in a hexahedral block of fluid collapsing in a three-dimensional pool and impacting a rectangular rigid obstacle on a flat bottom (see Kleefsman et al. 2005 for detailed description). The computational domain is 3.22 m long, 1 m high and 1 m width. The obstacle is located from $x = 0.6635$ m of the tank. The base-fixed mesh is splitted into 24 domains refined around the obstacle. Each domain is devoted to one process on the cluster. The numerical results are first qualitatively compared to Kleefsman's video recording in Figure 9 and quantitatively compared in Figure 10 to the experimental data and published numerical results mentioned hereinbefore.

Figure 9 shows a detailed qualitative comparison between Kleefsman's numerical results (left column), experimental video recording (central column) and our numerical results (right column) at times $t = 0.4, 0.6, 1, 1.8, 2$ and 4.8 s. The identification of the exact position of the experimental free surface is limited by the flow aeration observed in video recordings. One notes however the very satisfying overall agreement between both experimental and numerical Kleefsman's data and our results. The successive steps of the experiment are qualitatively well described by our numerical model, including the dambreak before the impact at $t = 0.4$ s, the first impact on the obstacle $t = 0.6$ s, the reflection on the wall $t = 1$ s, the formation of the first

reflected wave $t = 1$ s and the flow return after the second reflection on the right boundary $t = 4.8$ s. The comparison between numerical results reveals the fundamental difference between models. Kleesfman's model (Kleefsman et al. 2005) produces a variety of small-scale structures, such as centimetric secondary surface waves or deformations and water fragmentation in drops, while our model shows a much regular aspect of the free surface. At this stage of the qualitative analysis, and recalling that none of the models properly resolves the air–water mixture at the capillary scale, it appears difficult to identify which one of the numerical methods produces the better approximation of the real physical phenomenon.

The qualitative analysis showed that our model nicely compares with existing experimental and numerical results for a fully three-dimensional wave impact case. Our numerical results are now quantitatively compared to experimental data of Kleefsman et al. (2005) in terms of pressure (sensor $P1$) and free surface height (gauge $H2$). Similar comparisons with experiments performed on gauge $H4$ and sensor $P7$ data (not shown here) showed the same level of agreement. Figure 10 shows the temporal evolution of pressure and free surface elevation for a 6 s run comparing experimental data and numerical predictions provided by our model and a set of other published numerical results. Let us first focus on the comparison between the experiments (dotted line) and our model. At $t = 0$ s, the dambreaks but, as the $P1$ location is not reached by water, both pressures correspond to atmospheric pressure (Figure 10, left plot). Around $t = 0.45$ s, the water impacts the obstacle. One notes the simultaneous violent increase of pressure for both numerical and experimental data. The numerical pressure peak is less intense than the experimental one. As the water flows around and above the obstacle and then reflects on
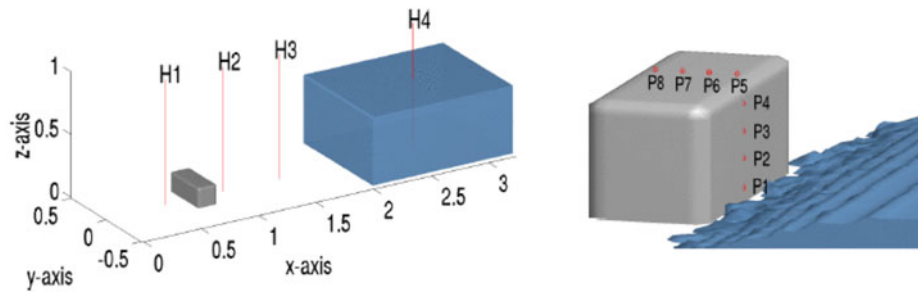
Figure 8.   Dam-break problem: domain geometry and sensors points from http://www.math.rug.nl/~veldman/comflow/dambreak.html.
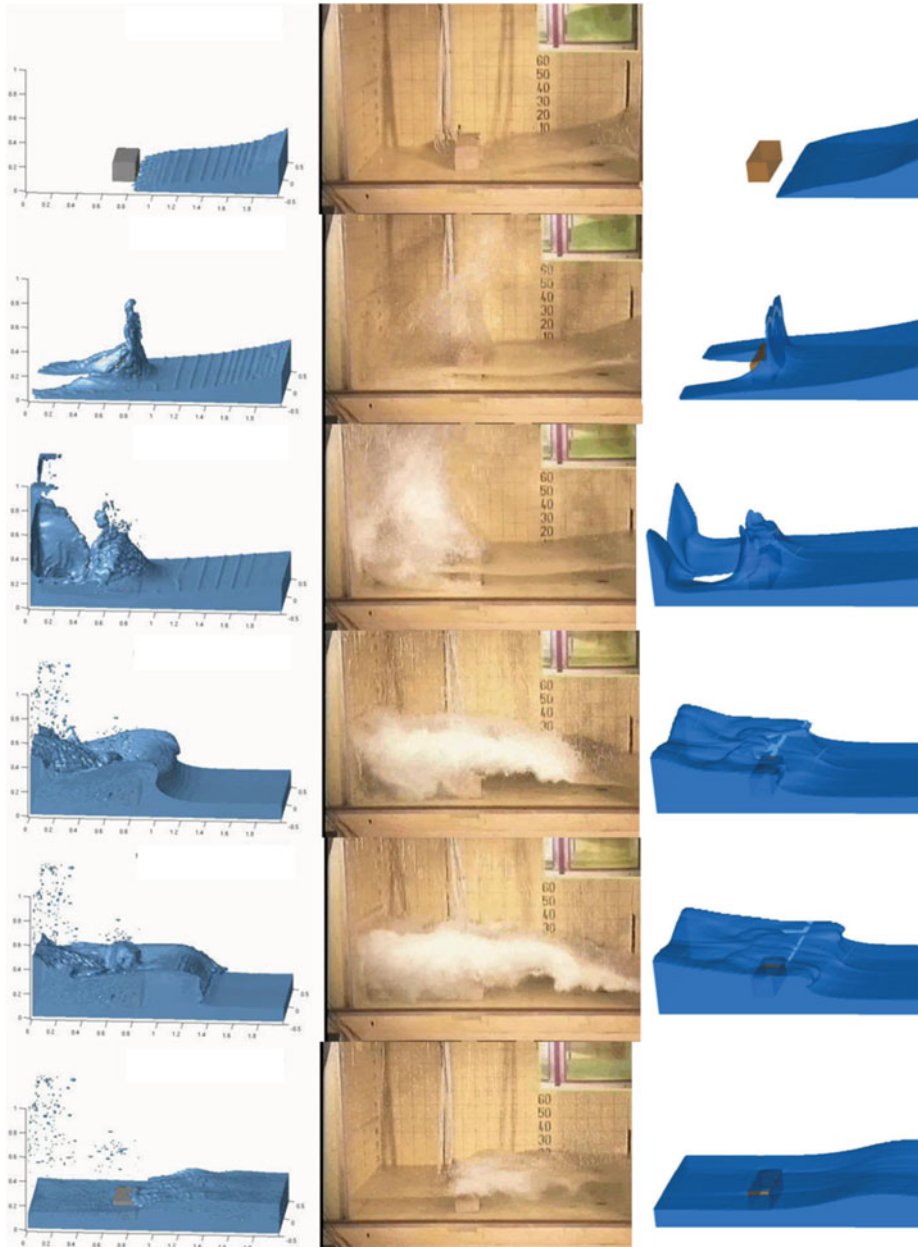

Figure 9.   Free surface elevation computed by Kleefsman (left), the experimentation (centre) and our simulation (right) at $t =$ 0.4, 0.6, 1, 1.8, 2, 4.8 s ($c = 20$ m.s^{-1}).
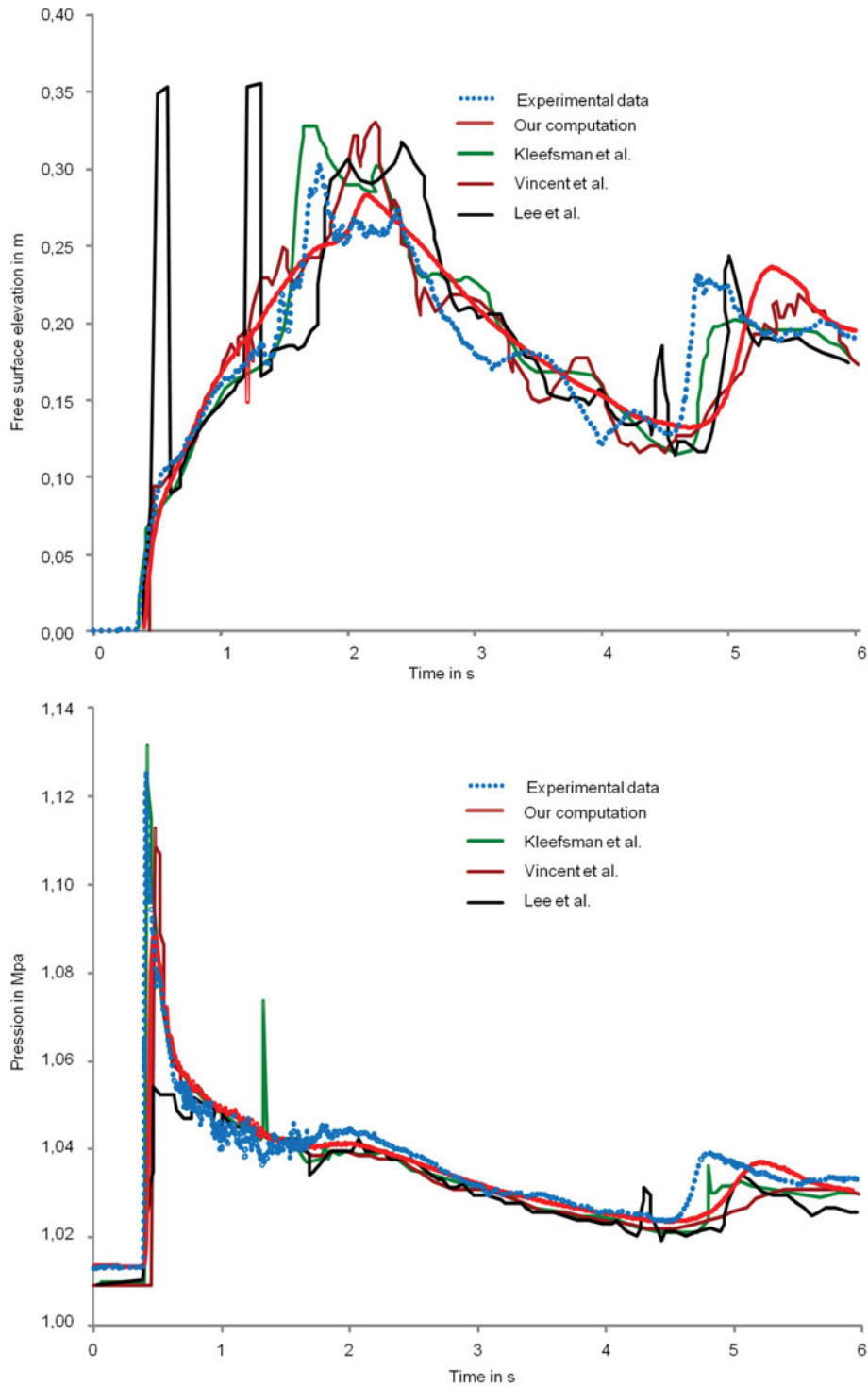
Figure 10. Comparison between three numerical models (Kleefsman et al. 2005; Vincent et al. 2010; Lee et al. 2010) and the present one. Top: free surface elevation (H2 gauge). Bottom: pressure at P1 sensor.

the left wall, the pressure at $P1$ continuously decreases. After $t = 1.8$ s, the obstacle is submerged by the reflecting wave, which induces a slight increase of pressure observed on both experimental and numerical results. One notes here

the appearance of a small delay between experiments and model. The pressure then still decreases until the arrival of the second reflected wave. The wave amplitude is rather well reproduced by the model but it comes with a delay
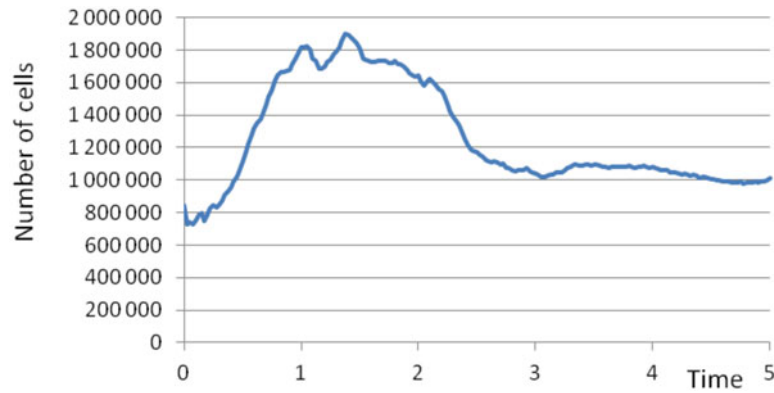
Figure 11.   Number of cells during the computation.

around 0.3 s and a smoother shape. Further comparisons are performed with experimental wave height for gauge $H2$ (Figure 10, left plot). The agreement between numerical and experimental wave heights is quite satisfactory. Both incoming and reflected waves are nicely reproduced by the model. Some differences are observed, in particular for the return wave which arrives with a slight delay around 0.3 s and a loss of amplitude.

Despite some discrepancies between model and experimental results, Figure 10 demonstrates the ability of the present numerical in describing the whole dam-break process. Kleesfman's experiments have also recently been used as a benchmark test for several other numerical models. A comparison here is proposed between the presented model, the truly incompressible smoothed particle hydro-dynamics of Lee et al. (2010), the NS–volume of fluid sub-mesh method (VOF-SM), model with hybrid Eulerian–Lagrangian resolution of the transport equations by Vincent et al. (2010) and Kleefsman's NS–VOF model (Kleefsman et al. 2005). Figure 10 shows a quantitative comparison between the four models and the experimental data for water height at $H2$ gauge (left graph) and pressure at $P1$ sensor (right graph). One notes first that both water level and pressure evolution are macroscopically well described by all numerical models, except the pressure peak during the first impact which is poorly described by the SPH method. Some differences are observed between models but their detailed description is not the aim of this work. The main conclusion of the proposed comparison is to highlight that our mathematical and numerical model nicely compares with state-of-the-art heavier numerical methods in a quantitative point of view.

3.3.   *A three-dimensional dam-break problem with AMR block-based mesh decomposition*

The previous section demonstrated that our simple two-fluid model is, at the least, competitive with heavier numerical models. The BB-AMR method described in this paper is now introduced and tested against the same Kleefsman's experiment (Kleefsman et al. 2005). Numerical parameters are CFL = 0.8, 3628 blocks, 48 cores, 48 domains or MPI process, 4 levels of mesh refinement, $\alpha_{max} = 0.2$ and $\alpha_{min} = 0.02$ as mesh refinement and coarsening parameters, respectively. Symmetry boundary conditions are imposed. The simulation time (4.8 s) is divided in 240 AMR time intervals.

The model accuracy in describing flow dynamics being unchanged, we restrict our comments to the numerical aspects of the computation. Figure 11 shows the evolution of the total number of cells. It is observed to evolve from 800,000 cells up to about 1 500,000 cells. By contrast with Coupez and Hachem (2013) and Hachem et al. (2013), the number of cells is not prescribed, but prescribing the maximum of mesh refinement level and the parameters α_{min} and α_{max} is sufficient to prevent a number of cells out of control. As seen in Figure 12 (left), the distribution of cells on the 48 domains follows the flow evolution (Figure 12, right plot) as explained in Section 2.4.2 and maintains a well-balanced processors load. Finally, the present simulation, performed with the standard Runge–Kutta time integration on 48 Intel X5675 cores, is achieved in 10 hours. The same test case simulated without BB-AMR but with a faster Adams–Bashforth time integration (3 levels) on 12 Itanium II cores is nearly five times longer.

4.   Concluding remarks and perspectives

This paper reports on numerical simulation of two-phase dam-break flows over an obstacle. The presented mathematical model is based on previous works of Helluy et al. (2005), Golay and Helluy (2007), and Sambe et al. (2011). The model has been specifically developed to simulate violent three-dimensional air and water flows, such as breaking or impacting waves. A set of assumptions is made to establish the best compromise between computing costs and physical relevancy in the considered physical framework. As such, this can be considered as an
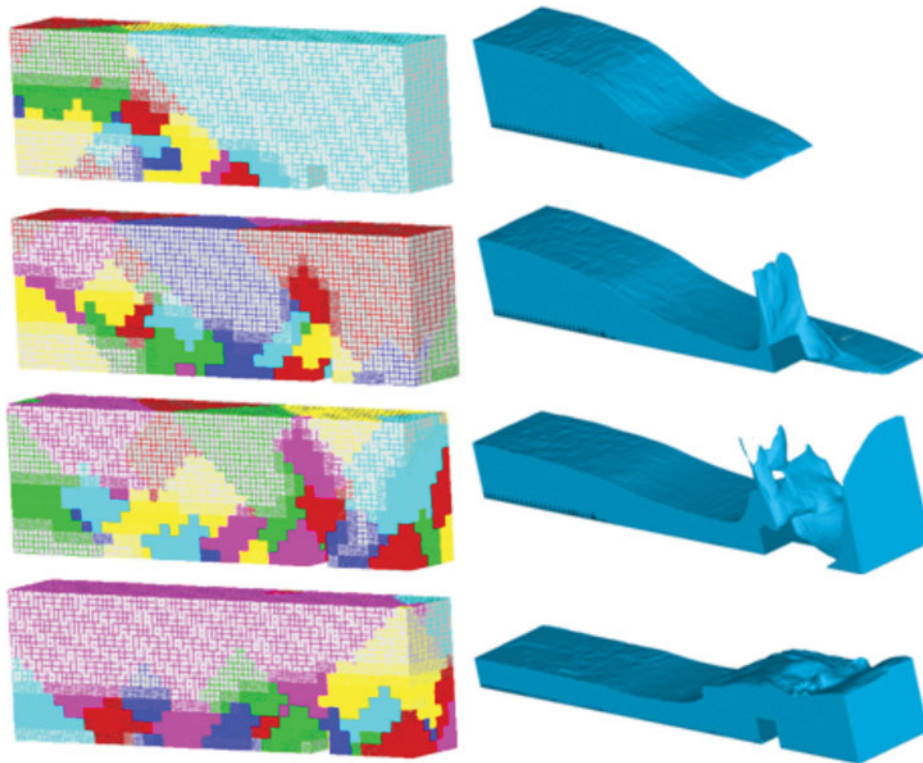
Figure 12. Domains due to the BB-AMR scheme (left) and air-water interface (right) at time 0.4, 0.6, 1.0, 2 s.

alternative 'intermediate' approach between full NS solvers and simplified shallow water models. It is expected that such approach, which provides access to fully three-dimensional simulations at a rather reasonable computing cost, can bring a great insight into a variety of coastal and marine engineering problems.

The model is based on an artificial linearised pressure law for low Mach regime. This artificial pressure law induces several interesting effects from mathematical and numerical viewpoints described above and in previous related papers. The novelty of the present work is to extend the multi-scale adaptive scheme proposed by Ersoy, Golay, and Yushchenko (2013) to our three-dimensional hyperbolic equations. The automatic mesh refinement process is performed thanks to an original criterion: the numerical density of entropy production. The increase of computational costs caused by the re-meshing procedure is outweighed by the introduction of an efficient BB-AMR allowing an easy and fast parallel implementation and computation. Through several two- and three-dimensional wave impact test cases, the overall numerical approach is validated. In particular, the entropy production appears to be a relevant criterion for automatic mesh refinement process, which leads to better describe the regions of interest (interfaces and impact zones) and to drastically reduce computational time.

On-going developments of the model focus on the optimisation of the artificial sound celerity, with the prospect of introducing a spatially and temporally variable value. As this approach can virtually be applied to any hyperbolic system, longer term extensions of the method will concern the application to shallow water models and fluid-structure interaction issues.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Abadie, S., D. Morichon, S. Grilli, and S. Glockner. 2010. "Numerical Simulation of Waves Generated by Landslides Using a Multiple-Fluid NavierStokes Model." *Coastal Engineering* 57 (9): 779–794. http://www.sciencedirect.com/science/article/pii/S0378383910000396.

Abgrall, R. 1996. "How to Prevent Pressure Oscillations in Multicomponent Flow Calculations: A Quasi-Conservative Approach." *Journal of Computational Physics* 125: 150–160. http://www.sciencedirect.com/science/article/pii/S0021999196900856.

Allaire, G., S. Clerc, and S. Kokh. 2002. "A Five-Equation Model for the Simulation of Interfaces Between Compressible Fluids." *Journal of Computational Physics* 181: 577–616. http://www.sciencedirect.com/science/article/pii/S0021999102971433.

Altmann, C., T. Belat, M. Gutnic, P. Helluy, H. Mathis, E. Sonnendrücker, W. Angulo, and J.M. Hérard. 2009. "A Local Time-Stepping Discontinuous Galerkin Algorithm for the MHD System." *ESAIM* 28: 33–54. http://dx.doi.org/10.1051/proc/2009038.

Altomare, C., A.J.C. Crespo, B.D. Rogers, J.M. Dominguez, X. Gironella, and M. Gomez-Gesteira. 2014. "Numerical Modelling of Armour Block Sea Breakwater with Smoothed Particle Hydrodynamics." *Computers and Structures* 130: 34–45. http://www.sciencedirect.com/science/article/pii/S0045794913002824.

Banari, A., C. Janßen, S.T. Grilli, and M. Krafczyk. 2014. "Efficient GPGPU Implementation of a Lattice Boltzmann Model for Multiphase Flows with High Density Ratios." *Computers & Fluids* 93 (0): 1–17. http://www.sciencedirect.com/science/article/pii/S0045793014000115.

Berger, M.J., and P. Colella. 1989. "Local Adaptive Mesh Refinement for Shock Hydrodynamics." *Journal of Computational Physics* 82 (1): 64–84. http://www.sciencedirect.com/science/article/pii/0021999189900351.

Chorin, A.J. 1967. "A Numerical Method for Solving Incompressible Viscous Flow Problems." *Journal of Computational Physics* 2 (1): 12–26. http://www.sciencedirect.com/science/article/pii/S0021999197957168.

Coupez, T., and E. Hachem. 2013. "Solution of High-Reynolds Incompressible Flow with Stabilized Finite Element and Adaptive Anisotropic Meshing." *Computer Methods in Applied Mechanics and Engineering* 267: 65–85. http://www.sciencedirect.com/science/article/pii/S0045782513002077.

Dutykh, D., R. Poncet, and F. Dias. 2011. "The VOLNA Code for the Numerical Modeling of Tsunami Waves: Generation, Propagation and Inundation." *European Journal of Mechanics - B/Fluids* 30 (6): 598–615. http://www.sciencedirect.com/science/article/pii/S0997754611000574.

Ersoy, M., F. Golay, and L. Yushchenko. 2013. "Adaptive Multiscale Scheme Based on Numerical Density of Entropy Production for Conservation Laws." *Central European Journal of Mathematics* 11 (8): 1392–1415. http://dx.doi.org/10.2478/s11533-013-0252-6.

Fuster, D., G. Agbaglah, C. Josserand, S. Popinet, and S. Zaleski. 2009. "Numerical Simulation of Droplets, Bubbles and Waves: State of the Art." *Fluid Dynamics Research* 41 (6): 065001. http://stacks.iop.org/1873-7005/41/i=6/a=065001.

Golay, F. 2009. "Numerical Entropy Production and Error Indicator for Compressible flows." *Comptes Rendus Mécanique* 337 (4): 233–237. http://www.sciencedirect.com/science/article/pii/S1631072109000564.

Golay, F., and P. Helluy. 2007. "Numerical Schemes for Low Mach Wave Breaking." *International Journal of Computational Fluid Dynamics* 21 (2): 69–86. http://www.tandfonline.com/doi/citedby/10.1080/10618560701343382#.VED5sDMcS9I.

Grilli, S., P. Guyenne, and F. Dias. 2001. "A Fully Non-Linear Model for Three-Dimensionnal Overturning Waves Over an Arbitrary Bottom." *International Journal for Numerical Methods in Fluids* 35: 829–867.

Hachem, E., S. Feghali, R. Codina, and T. Coupez. 2013. "Immersed Stress Method for Fluid Structure Interaction Using Anisotropic Mesh Adaptation." *International Journal for Numerical Methods in Engineering* 94 (9): 805–825. http://dx.doi.org/10.1002/nme.4481.

Helluy, P., F. Golay, J.P. Caltagirone, P. Lubin, S. Vincent, D. Drevrard, R. Marcer, et al. 2005. "Numerical Simulation of Wave Breaking." *M2AN* 39 (3): 591–607.

Hsiao, S.C., and T.C. Lin. 2010. "Tsunami-Like Solitary Waves impinging and Overtopping an Impermeable Seawall: Experiment and RANS Modeling." *Coastal Engineering* 57 (1): 1–18. http://www.sciencedirect.com/science/article/pii/S0378383909001288.

Hsu, T.W., C.M. Hsieh, and R. R. Hwang. 2004. "Using RANS to Simulate Vortex Generation and Dissipation Around Impermeable Submerged Double Breakwaters." *Coastal Engineering* 51 (7): 557–579. http://www.sciencedirect.com/science/article/pii/S0378383904000626.

Kleefsman, K.M.T., G. Fekken, A.E.P. Veldman, B. Iwanowski, and B. Buchner. 2005. "A Volume-of-Fluid Based Simulation Method for Wave Impact Problems." *Journal of Computational Physics* 206 (1): 363–393. http://www.sciencedirect.com/science/article/pii/S0021999104005170.

Koshizuka, S., H. Tamako, and Y. Oka. 1995. "A Particle Method for Incompressible Viscous Flow with Fluid Fragmentations." *Computational Fluid Dynamics Journal* 4 (1): 29–46.

Lee, E.S., D. Violeau, R. Issa, and S. Ploix. 2010. "Application of Weakly Compressible and Truly Incompressible SPH to 3-D Water Collapse in Waterworks." *Journal of Hydraulic Research* 48 (sup1): 50–60. http://www.tandfonline.com/doi/abs/10.1080/00221686.2010.9641245.

Losasso, F., F. Gibou, and R. Fedkiw. 2004. "Simulating Water and Smoke with an Octree Data Structure." *ACM Transactions on Graphics* 23 (3): 457–462. http://doi.acm.org/10.1145/1015706.1015745.

Lubin, P., & S. Glockner (2013, June). Detailed numerical investigation of the three-dimensional flow structures under breaking waves. In Proc. 7th International Conference on Coastal Dynamics Conference (pp. 1127–1136).

Min, C., and F. Gibou. 2007. "A Second Order Accurate Level Set Method on Non-Graded Adaptive Cartesian Grids." *Journal of Computational Physics* 225 (1): 300–321. http://www.sciencedirect.com/science/article/pii/S0021999106005912.

Plumerault, L.-R., D. Astruc, P. Villedieu, and P. Maron. 2011. "A Numerical Model for Aerated-Water Wave Breaking." *International Journal for Numerical Methods in Fluids*. http://dx.doi.org/10.1002/fld.2667.

Sambe, A.N., D. Sous, F. Golay, P. Fraunié, and R. Marcer. 2011. "Numerical Wave Breaking with Macro-Roughness." *European Journal of Mechanics - B/Fluids* 30 (6): 577–588. http://www.sciencedirect.com/science/article/pii/S0997754611000215.

Shi, F., J-.T. Kirby, J-.H. Harris, J-.D. Geiman, and S-.T. Grilli. 2012. "A High-Order Adaptive Time-Stepping TVD Solver for Boussinesq Modeling of Breaking Waves and Coastal Inundation." *Ocean Modelling* 4344: 36–51. http://www.sciencedirect.com/science/article/pii/S1463500311002010.

Toro, E.F. 1999. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. 2nded. New York, NY: Springer.

Vincent, S., G. Balmigère, J.-P. Caltagirone, and E. Meillot. 2010. "Eulerian–Lagrangian Multiscale Methods for Solving Scalar Equations – Application to Incompressible Two-Phase Flows." *Journal of Computational Physics* 229 (1): 73–106. http://www.sciencedirect.com/science/article/pii/S0021999109004926.

Wanner, G., & Hairer, E. (1991). Solving ordinary differential equations II (Vol. 1). Springer-Verlag, Berlin.

Williamschen, M.J., and C.P.T. Groth. 2013. "Parallel Anisotropic Block-Based Adaptive Mesh Refinement Algorithm For Three-Dimensional Flows." In *21st AIAA Computational*

Fluid Dynamics Conference, 1–22. San Diego, CA. http://arc.aiaa.org/doi/abs/10.2514/6.2013-2442

Yiu, K.F.C., D.M. Greaves, S. Cruz, A. Saalehi, and A.G.L. Borthwick. 1996. "Quadtree Grid Generation: Information Handling, Boundary Fitting and CFD Applications." *Computers & Fluids* 25 (8): 759–769. http://www.sciencedirect.com/science/article/pii/S0045793096000291.

Zhang, M., and W.M. Wu. 2011. "A Two Dimensional Hydrodynamic and Sediment Transport Model for Dam Break Based on Finite Volume Method with Quadtree Grid." *Applied Ocean Research* 33 (4): 297–308. http://www.sciencedirect.com/science/article/pii/S0141118711000563.

Zhang, X.D., J.Y. Trépanier, and R. Camarero. 2000. "A Posteriori Error Estimation for Finite-Volume Solutions of Hyperbolic Conservation Laws." *Computer Methods in Applied Mechanics and Engineering* 185: 1–19. http://www.sciencedirect.com/science/article/pii/S0045782599000997.

Zheng, Z.X.J., and C.P.T. Groth. 2012. "Block-Based Adaptive Mesh Refinement Finite-Volume Scheme for Hybrid Multi-Block Meshes." Proc In *7st conference on Computational Fluid Dynamics (ICCFD7-2504)*, 1–19. HI. http://www.iccfd.org/iccfd7/assets/pdf/papers/ICCFD7-2504_paper.pdf