Available online at www.sciencedirect.com
SCIENCE DIRECT®
Comput. Methods Appl. Mech. Engrg. 195 (2006) 5597–5620
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

On the simulation of flows with violent free surface motion

Rainald Löhner [a,*], Chi Yang [a], Eugenio Oñate [b]

[a] School of Computational Science and Informatics, M.S. 4C7, George Mason University, Fairfax, VA, USA
[b] International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnica de Cataluña, Barcelona, Spain

Received 21 March 2005; received in revised form 2 November 2005; accepted 3 November 2005

Abstract

A volume of fluid (VOF) technique has been developed and coupled with an incompressible Euler/Navier–Stokes solver operating on adaptive, unstructured grids to simulate the interactions of extreme waves and three-dimensional structures. The present implementation follows the classic VOF implementation for the liquid–gas system, considering only the liquid phase. Extrapolation algorithms to obtain velocities and pressure in the gas region near the free surface have been implemented. The VOF technique is validated against the classic dam-break problem, as well as series of 2D sloshing experiments and results from smoothed particle hydrodynamics (SPH) calculations. These and a series of other examples demonstrate that the present CFD method is capable of simulating violent free surface flows with strong nonlinear behavior.

Keywords: Marine engineering; Computational techniques; Incompressible flow; Projection schemes; VOF; Level set; FEM; CFD

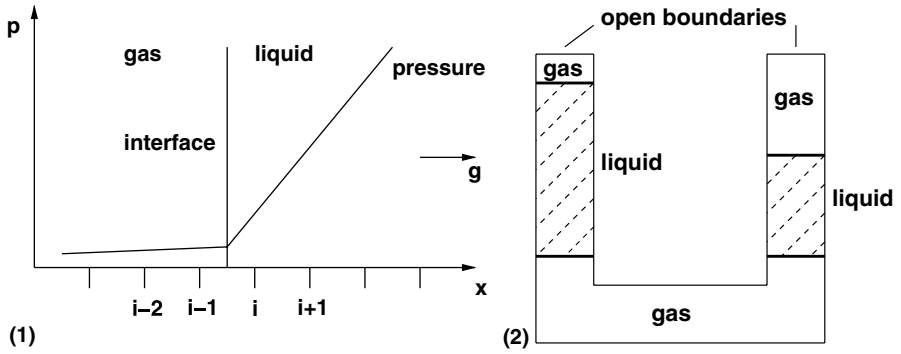
1. Introduction

High sea states, waves breaking near shores and moving ships, the interaction of extreme waves with floating structures, green water on deck and sloshing (e.g., in liquid natural gas (LNG) tankers) are but a few examples of flows with violent free surface motion. Many of these flows have a profound impact on marine engineering [4].

The computation of highly nonlinear free surface flows is difficult because neither the shape nor the position of the interface between air and water is known a priori; on the contrary, it often involves unsteady fragmentation and merging processes. There are basically two approaches to compute flows with free surface: interface-tracking and interface-capturing methods. The former computes the liquid flow only, using a numerical grid that adapts itself to the shape and position of the free surface. The free surface is represented and tracked explicitly either by marking it with special marker points, or by attaching it to a mesh surface. Various surface fitting methods for attaching the interface to a mesh surface were developed during the past decades [15,1,40]. In the interface-tracking methods, the free surface is treated as a boundary of the computational domain, where the kinematic and dynamic boundary conditions are applied. These methods cannot be used if the interface topology changes significantly, as is contemplated here for overturning or breaking waves. The second possible approach is given by the so-called interface-capturing methods [45,26,59,57,60,49,8,17,10,5,28,51,16,11]. These consider both fluids as a single effective fluid with variable properties; the interface is captured as a region of sudden change in fluid properties. The main problem of complex free surface flows is that the density ρ jumps by three orders of magnitude between the gaseous and liquid phase. Moreover, this surface can move, bend and reconnect in arbitrary ways. The difficulties that can arise if one treats either the complete system or just the liquid phase can be illustrated on

* Corresponding author. Tel.: +1 703 993 1990.
E-mail address: rlohner@gmu.edu (R. Löhner).

0045-7825/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2005.11.010



Figs. 1,2. Hydrostatic pressure distribution and communicating tubes.

two small examples. The first one considers hydrostatic flow, where the exact solution is $\mathbf{v} = 0$, $p = -\mathbf{g} \cdot (\mathbf{x} - \mathbf{x}_0)$, where $\mathbf{x}_0$ denotes the position of the free surface (see Fig. 1). Unless the free surface coincides with the faces of elements, there is no way for typical finite element shape functions to capture the discontinuity in the gradient of the pressure. This implies that one has to either increase the number of Gauss-points [10] or modify (e.g., enrich) the shape function space [11]. Using the standard linear element procedure leads to spurious velocity jumps at the interface, as any small pressure gradient that 'pollutes over' from the water to the air region will accelerate the air considerably. This in turn will lead to loss of divergence, causing more spurious pressures. The whole cycle may, in fact, lead to a complete divergence of the solution.

Faced with this dilemma, most flows with free surfaces have been solved neglecting the air. The shortcomings of this approach can be illustrated in the second example, sketched in Fig. 2. A tube is filled with a column of fluid which is interrupted by a pocket of air. Neglecting the air would never allow the higher column to push the lower column up. In the present case, we have followed this approach, fully aware of the limitations.

The remainder of the paper is organized as follows: Section 2 summarizes the basic elements of the present incompressible flow solver; Sections 3 and 4 describe the temporal and spatial discretization; Section 5 describes the volume of fluid extensions; some examples are shown in Section 6; finally, some conclusions are given in Section 7.

2. Basic elements of the solver

In order to fix the notation, the equations describing incompressible, Newtonian flows in an arbitrary Lagrangian–Eulerian (ALE) frame are written as:

$$\rho \mathbf{v}_{,t} + \rho \mathbf{v}_a \nabla \mathbf{v} + \nabla p = \nabla \mu \nabla \mathbf{v} + \rho \mathbf{g}, \tag{1}$$

$$\nabla \cdot \mathbf{v} = 0. \tag{2}$$

Here ρ denotes the density, $\mathbf{v}$ the velocity vector, p the pressure, μ the viscosity and $\mathbf{g}$ the gravity vector. The advective velocity is given by $\mathbf{v}_a = \mathbf{v} - \mathbf{w}$, where $\mathbf{w}$ is the mesh velocity. We remark that both the gaseous and liquid phases are considered incompressible, thus Eq. (2). The liquid–gas interface is described by a scalar equation of the form:

$$\Phi_{,t} + \mathbf{v}_a \cdot \nabla \Phi = 0. \tag{3}$$

For the classic volume of fluid (VOF) technique, Φ represents the total density of the material in a cell/element or control volume (see [45,26,49,8,17,5,28]). For pseudo-concentration techniques, Φ represents the percentage of liquid in a cell/element or control volume. For the level set (LS) approach Φ represents the signed distance to the interface [51,16]. For a combination of these approaches, see [52].

Since over a decade [37,43,48,40] the numerical schemes chosen by the authors to solve the incompressible Navier–Stokes equations given by Eqs. (1) and (2) have been based on the following criteria:

- Spatial discretization using **unstructured grids** (in order to allow for arbitrary geometries and adaptive refinement).
- Spatial approximation of unknowns with **simple finite elements** (in order to have a simple input/output and code structure).
- Temporal approximation using **implicit integration of viscous terms and pressure** (the interesting scales are the ones associated with advection).
- Temporal approximation using **explicit integration of advective terms**.
- **Low-storage, iterative solvers** for the resulting systems of equations (in order to solve large 3D problems).
- Steady results that are **independent from the timestep** chosen (in order to have confidence in convergence studies).

3. Temporal discretization

For most of the applications listed above, the important physical phenomena propagate with the **advective** timescales. We will therefore assume that the advective terms require an explicit time integration. Diffusive phenomena typically occur at a much faster rate, and can/should therefore be integrated implicitly. Given that the pressure establishes itself immediately through the pressure-Poisson equation, an implicit integration of pressure is also required. The hyperbolic character of the advection operator and the elliptic character of the pressure-Poisson equation have led to a number of so-called projection schemes. The key idea is to predict first a velocity field from the current flow variables without taking the divergence constraint into account. In a second step, the divergence constraint is enforced by solving a pressure-Poisson equation. The velocity increment can therefore be separated into an advective–diffusive and pressure increment:

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta\mathbf{v}^a + \Delta\mathbf{v}^p = \mathbf{v}^* + \Delta\mathbf{v}^p. \tag{4}$$

For an explicit (forward Euler) integration of the advective terms, with implicit integration of the viscous terms, one complete timestep is given by:

- *Advective–diffusive prediction*: $\mathbf{v}^n \rightarrow \mathbf{v}^*$

$$\left[\frac{\rho}{\Delta t} - \theta\nabla\mu\nabla\right](\mathbf{v}^* - \mathbf{v}^n) + \mathbf{v}^n_{\mathrm{a}} \cdot \nabla\mathbf{v}^n + \nabla p^n = \nabla\mu\nabla\mathbf{v}^n + \rho\mathbf{g}. \tag{5}$$

- *Pressure correction*: $p^n \rightarrow p^{n+1}$

$$\nabla \cdot \mathbf{v}^{n+1} = 0, \tag{6}$$

$$\rho\frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} + \nabla(p^{n+1} - p^n) = 0, \tag{7}$$

which results in

$$\nabla \cdot \frac{1}{\rho}\nabla(p^{n+1} - p^n) = \frac{\nabla \cdot \mathbf{v}^*}{\Delta t}. \tag{8}$$

- *Velocity correction*: $\mathbf{v}^* \rightarrow \mathbf{v}^{n+1}$

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \frac{\Delta t}{\rho}\nabla(p^{n+1} - p^n). \tag{9}$$

At steady-state, $\mathbf{v}^* = \mathbf{v}^n = \mathbf{v}^{n+1}$ and the residuals of the pressure correction vanish, implying that the result does not depend on the timestep Δt. θ denotes the implicitness-factor for the viscous terms ($\theta = 1$: first order, fully implicit; $\theta = 0.5$: second order, Crank–Nicholson). One can replace the one-step explicit advective–diffusive predictor by a multistage Runge–Kutta scheme [42], allowing for higher accuracy in the advection-dominated regions and larger timesteps without a noticeable increment in CPU cost. A k-step, time-accurate Runge–Kutta scheme or order k for the advective parts may be written as:

$$\rho\mathbf{v}^i = \rho\mathbf{v}^n + \alpha^i\gamma\Delta t(-\rho\mathbf{v}^{i-1}_{\mathrm{a}} \cdot \nabla\mathbf{v}^{i-1} - \nabla p^n + \nabla\mu\nabla\mathbf{v}^{i-1}), \quad i = 1, k-1, \tag{10}$$

$$\left[\frac{\rho}{\Delta t} - \theta\nabla\mu\nabla\right](\mathbf{v}^k - \mathbf{v}^n) + \rho\mathbf{v}^{k-1}_{\mathrm{a}} \cdot \nabla\mathbf{v}^{k-1} + \nabla p^n = \nabla\mu\nabla\mathbf{v}^{k-1}. \tag{11}$$

Here, the α^i are the standard Runge–Kutta coefficients $\alpha^i = 1/(k + 1 - i)$. As compared to the original scheme given by Eq. (5), the $k - 1$ stages of Eq. (10) may be seen as a predictor (or replacement) of $\mathbf{v}^n$ by $\mathbf{v}^{k-1}$. The original right-hand side has not been modified, so that at steady-state $\mathbf{v}^n = \mathbf{v}^{k-1}$, preserving the requirement that the steady-state be independent of the timestep Δt. The factor γ denotes the local ratio of the stability limit for explicit timestepping for the viscous terms versus the timestep chosen. Given that the advective and viscous timestep limits are proportional to:

$$\Delta t_{\mathrm{a}} \approx \frac{h}{|\mathbf{v}|}; \quad \Delta t_{\mathrm{v}} \approx \frac{\rho h^2}{\mu}, \tag{12}$$

we immediately obtain

$$\gamma = \frac{\Delta t_{\mathrm{v}}}{\Delta t_{\mathrm{a}}} \approx \frac{\rho|\mathbf{v}|h}{\mu} \approx Re_h, \tag{13}$$

or, in its final form:

$$\gamma = \min(1, Re_h). \tag{14}$$

In regions away from boundary layers, this factor is O(1), implying that a high-order Runge–Kutta scheme is recovered. Conversely, for regions where $Re_h = O(0)$, the scheme reverts back to the original one (Eq. (5)). Projection schemes of this kind (explicit advection with a variety of schemes, implicit diffusion, pressure-Poisson equation for either the pressure or pressure increments) have been widely used in conjunction with spatial discretizations based on finite differences [33,2,3,1], finite volumes [29], and finite elements [21,12,23,37,43,48,40,53,13,30,9,36,31,42,7].

One **complete timestep** is then comprised of the following substeps:

- predict velocity (advective–diffusive predictor, Eqs. (5), (10), and (11));
- extrapolate the pressure (imposition of boundary conditions);
- update the pressure (Eq. (8));
- correct the velocity field (Eq. (9));
- extrapolate the velocity field;
- update the scalar interface indicator.

4. Spatial discretization

As stated before, we desire a spatial discretization with unstructured grids in order to:

- approximate arbitrary domains, and
- perform adaptive refinement in a straightforward manner, i.e., without changes to the solver.

From a numerical point of view, the difficulties in solving Eqs. (1)–(3) are the usual ones. First-order derivatives are problematic (overshoots, oscillations, instabilities), while second-order derivatives can be discretized by a straightforward Galerkin approximation. We will first treat the advection operator and then proceed to the divergence operator. Given that for tetrahedral grids solvers based on edge data structures incur a much lower indirect addressing and CPU overhead than those based on element data structures [41], only these will be considered.

4.1. The advection operator

It is well known that a straightforward Galerkin approximation of the advection terms will lead to an unstable scheme (recall that on a 1-D mesh of elements with constant size, the Galerkin approximation is simply a central difference scheme). Three ways have emerged to modify (or stabilize) the Galerkin discretization of the advection terms:

- integration along characteristics [27,22];
- Taylor–Galerkin (or streamline diffusion) [32,6,12]; and
- edge-based upwinding [40].

Of these, we only consider the third option here. The Galerkin approximation for the advection terms yields a right-hand side (RHS) of the form:

$$r^i = D^{ij}\mathscr{F}_{ij} = D^{ij}(\mathbf{f}_i + \mathbf{f}_j), \tag{15}$$

where the $\mathbf{f}_i$ are the 'fluxes along edges':

$$\mathbf{f}_i = S_k^{ij}\mathbf{F}_i^k, \quad S_k^{ij} = \frac{d_k^{ij}}{D^{ij}}, \quad D^{ij} = \sqrt{d_k^{ij}d_k^{ij}}, \tag{16}$$

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j, \quad \mathbf{f}_i = (S_k^{ij}v_i^k)\mathbf{v}_i, \quad \mathbf{f}_j = (S_k^{ij}v_j^k)\mathbf{v}_j \tag{17}$$

and the edge-coefficients are based on the shape-functions N^i as follows:

$$d_k^{ij} = \frac{1}{2}\int_\Omega (N^i{}_{,k}N^j - N^j{}_{,k}N^i)\,\mathrm{d}\Omega. \tag{18}$$

A consistent numerical flux is given by

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |v^{ij}|(\mathbf{v}_i - \mathbf{v}_j), \quad v^{ij} = \frac{1}{2}S_k^{ij}(v_i^k + v_j^k). \tag{19}$$

As with all other edge-based upwind fluxes, this first-order scheme can be improved by reducing the difference $\mathbf{v}_i - \mathbf{v}_j$ through (limited) extrapolation to the edge center [41]. The same scheme is used for the transport equation that describes the propagation of the VOF fraction, pseudo-concentration or distance to the free surface given by Eq. (3).

4.2. The divergence operator

A persistent difficulty with incompressible flow solvers has been the derivation of a stable scheme for the divergence constraint (2). The stability criterion for the divergence constraint is also known as the Ladyzenskaya–Babuska–Brezzi or LBB condition [24]. The classic way to satisfy the LBB condition has been to use different functional spaces for the velocity and pressure discretization [18]. Typically, the velocity space has to be richer, containing more degrees of freedom than the pressure space. Elements belonging to this class are the p1/p1 + bubble mini-element [50], the p1/iso-p1 element [56], and the p1/p2 element [54]. An alternative way to satisfy the LBB condition is through the use of artificial viscosities [37], 'stabilization' [19,55,20] or a 'consistent numerical flux' (more elegant terms for the same thing). The equivalency of these approaches has been repeatedly demonstrated (e.g. [50,37,41]). The approach taken here is based on consistent numerical fluxes, as it fits naturally into the edge-based framework. For the divergence constraint, the Galerkin approximation along edge i,j is given by

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j, \quad \mathbf{f}_i = S_k^{ij} v_i^k, \quad \mathbf{f}_j = S_k^{ij} v_j^k. \tag{20}$$

A consistent numerical flux may be constructed by adding pressure terms of the form:

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\lambda^{ij}|(p_i - p_j), \tag{21}$$

where the eigenvalue λ^{ij} is given by the ratio of the characteristic advective timestep of the edge Δt and the characteristic advective length of the edge l:

$$\lambda^{ij} = \frac{\Delta t^{ij}}{l^{ij}}. \tag{22}$$

Higher order schemes can be derived by reconstruction and limiting, or by substituting the first-order differences of the pressure with third-order differences:

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\lambda^{ij}|\left(p_i - p_j + \frac{l^{ij}}{2}(\nabla p_i + \nabla p_j)\right) \tag{23}$$

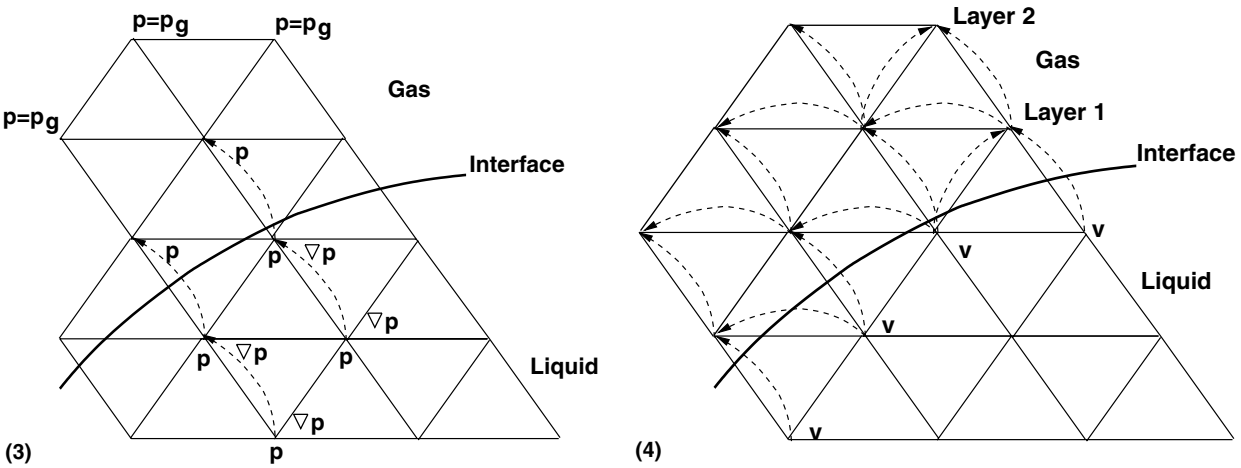
This results in a stable, low-diffusion, fourth-order damping for the divergence constraint.

5. Volume of fluid extensions

The extension of a solver for the incompressible Navier–Stokes equations to handle free surface flows via the VOF or LS techniques requires a series of extensions which are the subject of the present section. Before going on, we remark that both the VOF and LS approaches were implemented as part of this effort. Experience indicates that both work well. For VOF, the profiles can be sharp fronts that propagate through the domain. This places a higher demand on the quality of the advection schemes used to integrate Eq. (3). In particular, it is important to have a monotonicity preserving scheme for Φ. The profiles advected for LS are smoother, and do not place such demands on advection schemes. On the other hand, for LS, it is important to balance the cost and accuracy loss of reinitializations vis-a-vis propagation. Given that the advection solvers used are all monotonicity preserving, and that the VOF option is less CPU-demanding than LS, only the VOF technique is considered in the following.

5.1. Extrapolation of the pressure

The pressure in the gas region needs to be extrapolated properly in order to obtain the proper velocities in the region of the free surface. This extrapolation is performed using a three step procedure. In the first step, the pressures for all point in the gas region are set to (constant) values, either the atmospheric pressure or, in the case of bubbles, the pressure of the particular bubble. In a second step, the gradient of the pressure for the points in the liquid that are close to the liquid–gas interface are extrapolated from the points inside the liquid region (see Fig. 3). This step is required as the pressure gradient for these points cannot be computed properly from the data given. Using this information (i.e., pressure and gradient of pressure), the pressure for the points in the gas that are close to the liquid–gas interface are computed.



Figs. 3,4. Extrapolation of the pressure and velocity.

5.2. Extrapolation of the velocity

The velocity in the gas region needs to be extrapolated properly in order to propagate accurately the free surface. This extrapolation is started by initializing all velocities in the gas region to $\mathbf{v} = 0$. Then, for each subsequent layer of points in the gas region where velocities have not been extrapolated (unknown values), an average of the velocities of the surrounding points with known values is taken (see Fig. 4).

5.3. Imposition of constant mass

Experience indicates that the amount of liquid mass (as measured by the region where the VOF indicator is larger than a cut-off value) does not remain constant for typical runs. The reasons for this loss or gain of mass are manifold: loss of

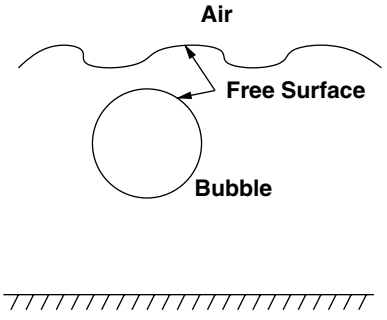


Fig. 5. Bubble in water.

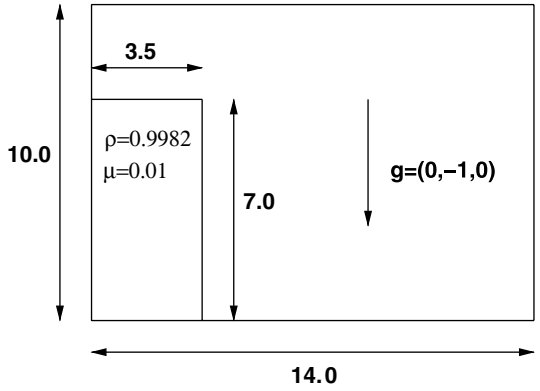


Fig. 6a. Breaking dam: problem definition.

steepness in the interface region, inexact divergence of the velocity field, boundary velocities, etc. This lack of exact conservation of liquid mass has been reported repeatedly in the literature [51,52,16]. The recourse taken here is the classic one: add/remove mass in the interface region in order to obtain an exact conservation of mass. At the end of every timestep, the total amount of fluid mass is compared to the expected value. The expected value is determined from the mass at the previous timestep, plus the mass-flux across all boundaries during the timestep. The differences in expected and actual mass are typically very small, so that quick convergence is achieved by simply adding and removing mass appropriately. The amount of mass taken/added is made proportional to the absolute value of the normal velocity of the interface:

$$v_n = \left| \mathbf{v} \cdot \frac{\nabla \Phi}{|\nabla \Phi|} \right|. \tag{24}$$

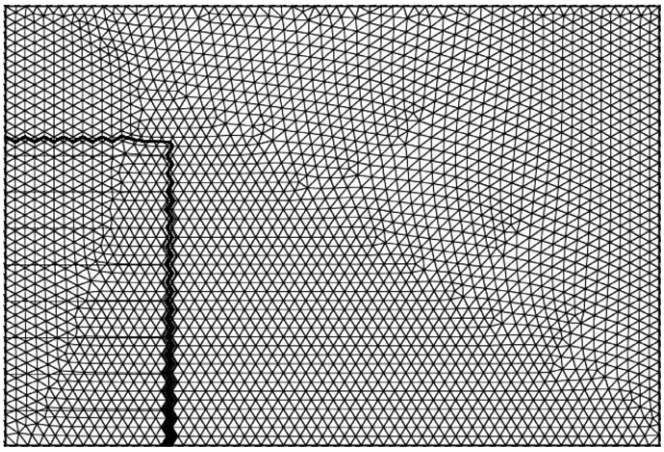
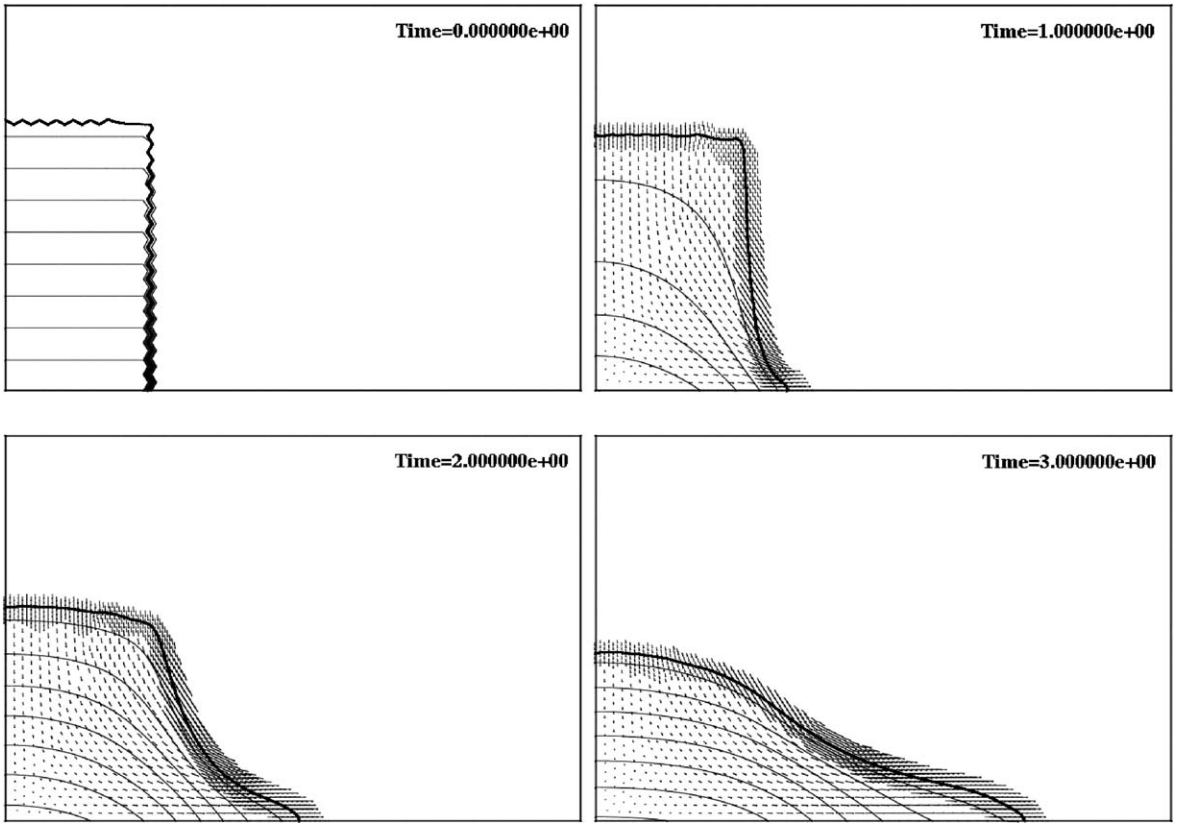


Fig. 6b. Breaking dam: surface discretization for the coarse mesh.



Figs. 6c–f. Breaking dam: flowfield at different times.

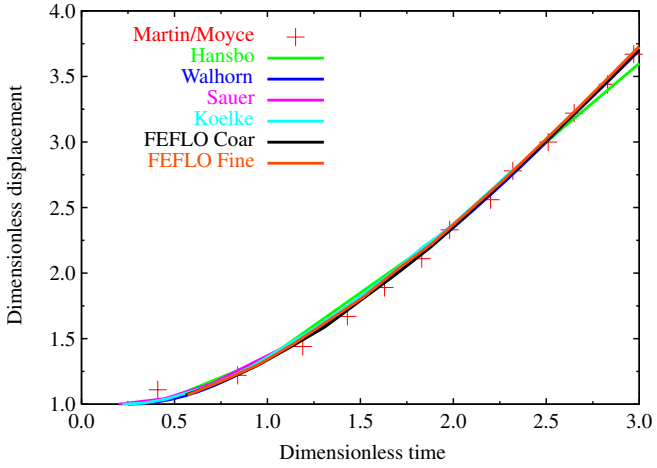


Fig. 6g. Breaking dam: horizontal displacement.

In this way the regions with no movement of the interface remain unaffected by the changes made to the interface in order to impose strict conservation of mass.

5.4. Deactivation of air region

Given that the air region is not treated/updated, any CPU spent on it may be considered wasted. Most of the flow solver work is spent in loops over the edges (upwind solvers, limiters, gradients, etc.). Given that edges have to be grouped in order to avoid memory contention/allow vectorization when forming right-hand sides [38,39], this opens a natural way of avoiding unnecessary work: form relatively small edge-groups that still allow for efficient vectorization, and deactivate groups instead of individual edges [41]. In this way, the basic loops over edges do not require any changes. The `if`-test whether an edge group is active or deactive is placed outside the inner loops over edges, leaving them unaffected. On scalar processors, edges-groups as small as `negrp` = 8 are used. Furthermore, if points and edges are grouped together in such a way that proximity in memory mirrors spatial proximity, most of the edges in air will not incur any CPU penalty.

5.5. Treatment of bubbles

The treatment of bubbles follows the classic assumption that the timescales associated with speed of sound in the bubble are much faster than the timescales of the surrounding fluid. This implies that at each instance the pressure in the bubble is

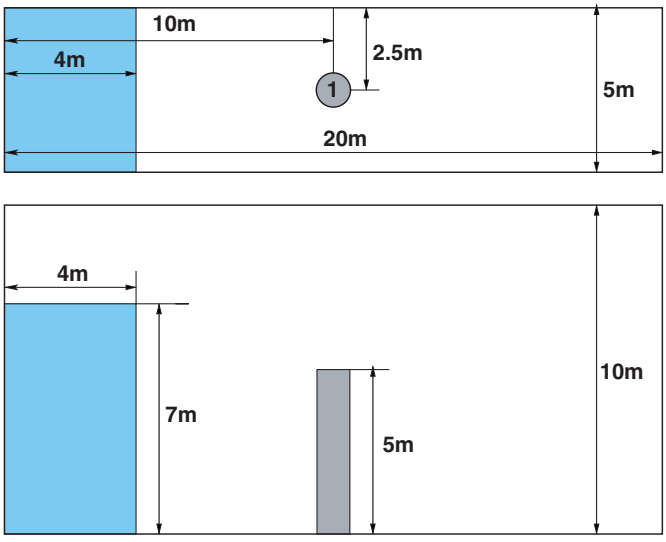


Fig. 7a. 3D dam-break wave with circular cylinder: problem definition.

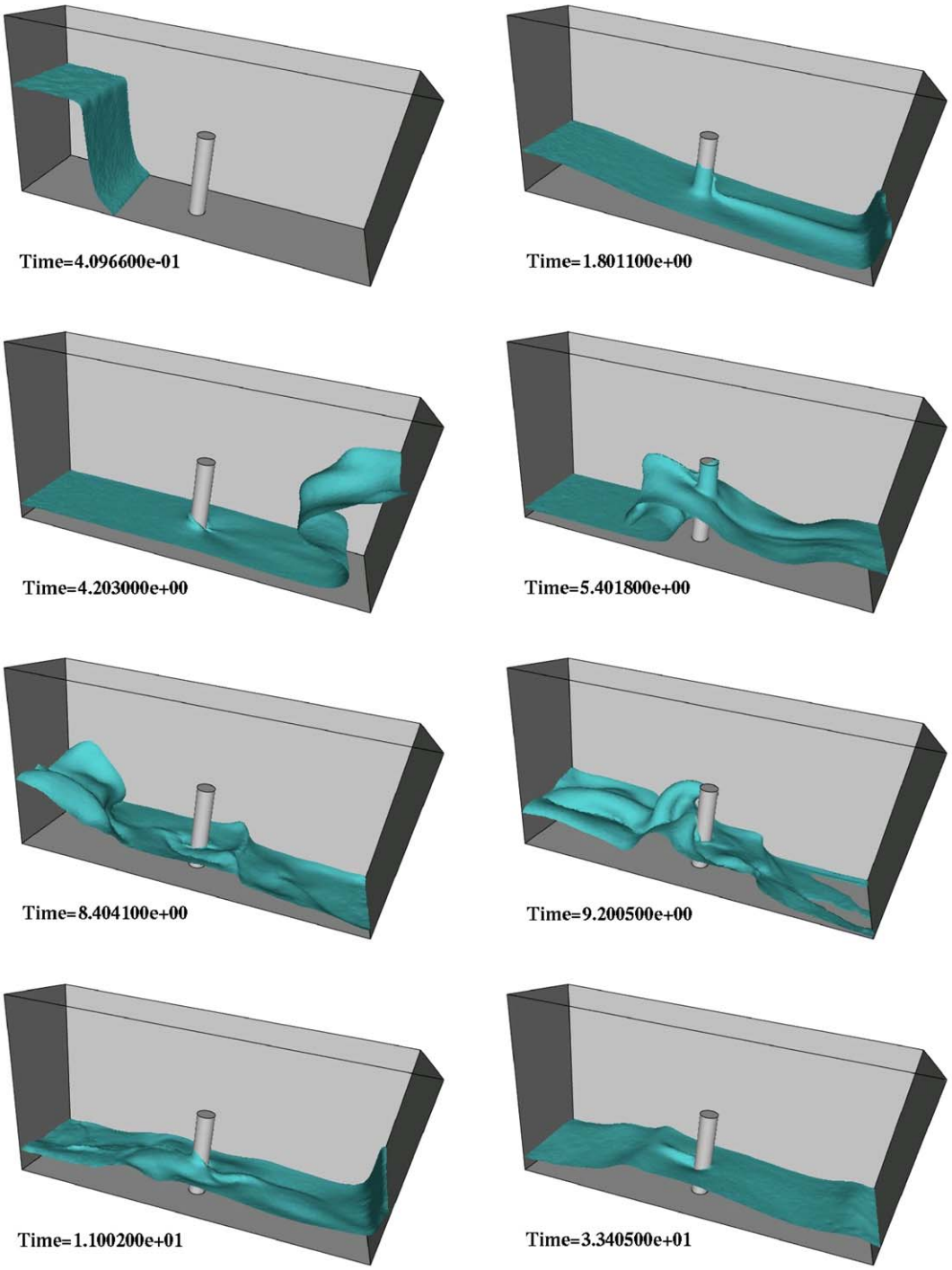


Fig. 7b. Free surface wave elevation.

(spatially) constant. As long as the bubble is not in contact with the atmospheric air (see Fig. 5), the pressure can be obtained from the isentropic relation:

$$\frac{p_{\mathrm{b}}}{p_{\mathrm{b0}}} = \left(\frac{\rho_{\mathrm{b}}}{\rho_{\mathrm{b0}}}\right)^{\gamma}, \tag{25}$$

where $p_{\mathrm{b}}, \rho_{\mathrm{b}}$ denote the pressure and density in the bubble and p_{b0},ρ_{b0} the reference values (e.g., those at the beginning of the simulation). The gas in the bubble is marked by solving a scalar advection equation of the form given by Eq. (3):

$$b_{,t} + \mathbf{v}_{\mathrm{a}} \cdot \nabla b = 0. \tag{26}$$

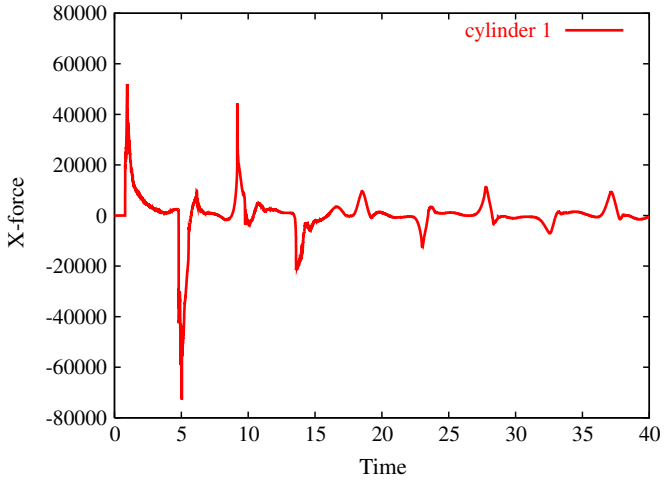


Fig. 7c. Wave impact force on cylinder.

At the beginning of every timestep the total volume occupied by gas is added. From this volume the density is inferred, and the pressure is computed from Eq. (25).

At the end of every timestep, a check is performed to see if the bubble has reached contact with the air. Should this be the case, the pressure in the bubble is set to atmospheric pressure. One then typically observes a rather quick collapse of the bubble.

6. Examples

6.1. Breaking dam problem

This is a classic test case for free surface flows.

The problem definition is shown in Fig. 6a. This case was run on a coarse mesh with `nelem = 16,562` elements, a fine mesh with `nelem = 135,869` and an adaptively refined mesh (where the coarse mesh was the base mesh) with approximately `nelem = 30,000` elements. The refinement indicator for the latter was the free surface, and the mesh was adapted every 5 time steps. Fig. 6b shows the discretization for the coarse mesh, and Figs. 6c–f the development of the flowfield and

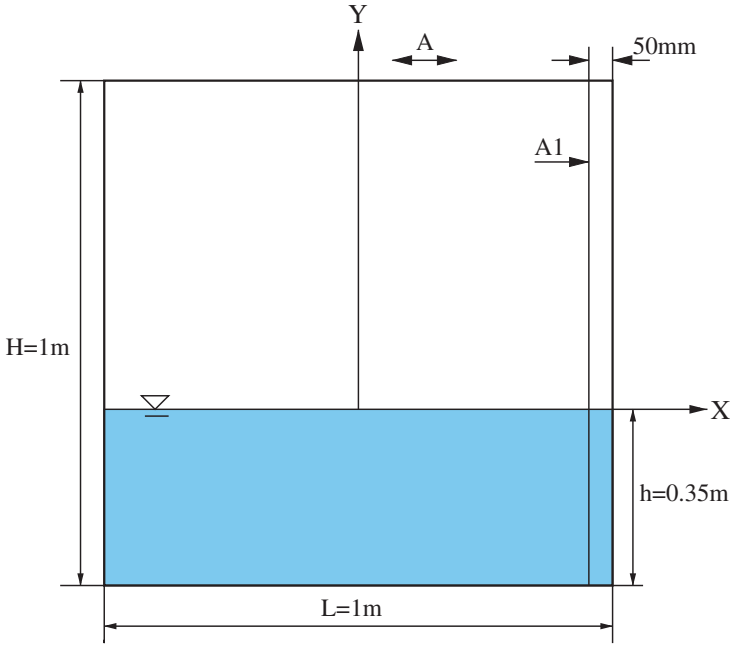


Fig. 8a. 2D tank: problem definition.

the free surface until the column of water hits the right wall. Note the mesh adaptation in time. The initialization was performed using a step-function (i.e., a discontinuous function). For this reason, when displayed, the free surface indicator will look like a 'sawtooth' oscillation in 2D and a 'heavy sea' in 3D (see also Fig. 11 below). The results obtained for the horizontal location of the free surface along the bottom wall are compared to the experimental values of Martin and Moyse [44], as well as the numerical results obtained by Hansbo [25], Kölke [34] and Walhorn [58] in Fig. 6g. The dimensionless time and displacement are given by $\tau = t\sqrt{2g/a}$ and $\delta = x/a$, where a is the initial width of the water column. As one can see, the agreement is very good, even for the coarse mesh. The difference between the adaptively refined mesh and the fine mesh was almost indistinguishable, and therefore only the results for the fine mesh are shown in the graph. CPU times were of the order of 1–5 min on a Dell PC (3.2 GHz IP4, Linux OS, 2Gbyte RAM, Intel compiler).

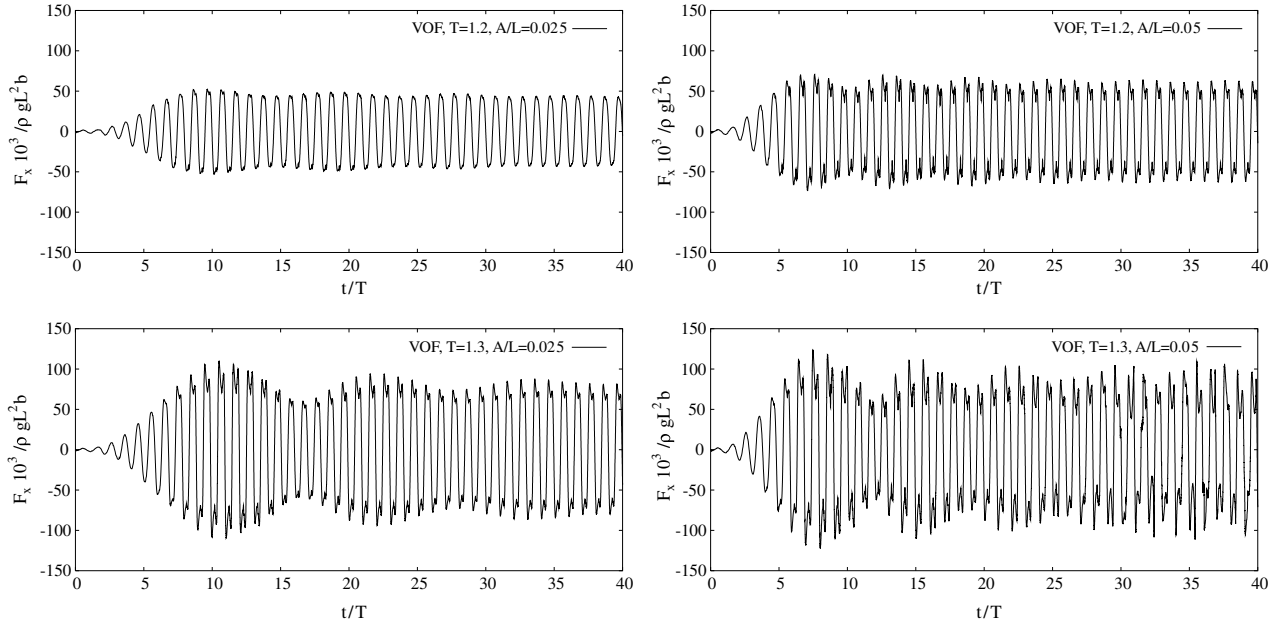


Fig. 8b. 2D tank: time history of lateral force F_x.

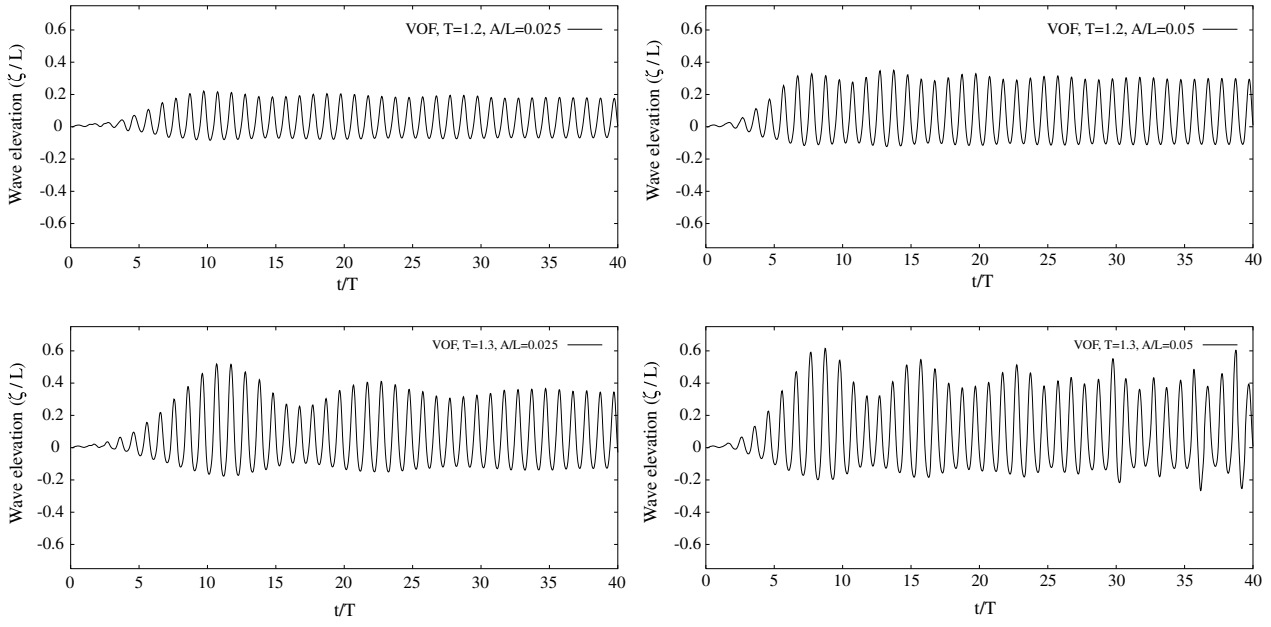


Fig. 8c. 2D tank: time history of wave elevation (Probe A1).

6.2. 3D dam-break wave interacting with a circular cylinder

The previous example validated the accuracy of the numerical model for studying dam breaking. This example considers a three-dimensional dam-break wave interacting with a circular cylinder. The tank is 20 m long, 5 m wide, and 10 m high. The volume of water initially contained behind a thin gate is $4\text{ m} \times 5\text{ m} \times 7\text{ m}$. The circular cylinder, which has a radius $r = 1$ m and height $h = 5$ m, is placed in the middle of the tank. The problem definition is shown in Fig. 7a. The entire tank is selected as the computational domain with `nelem = 1,315,224` elements. Fig. 7b shows a sequence of snapshots of the

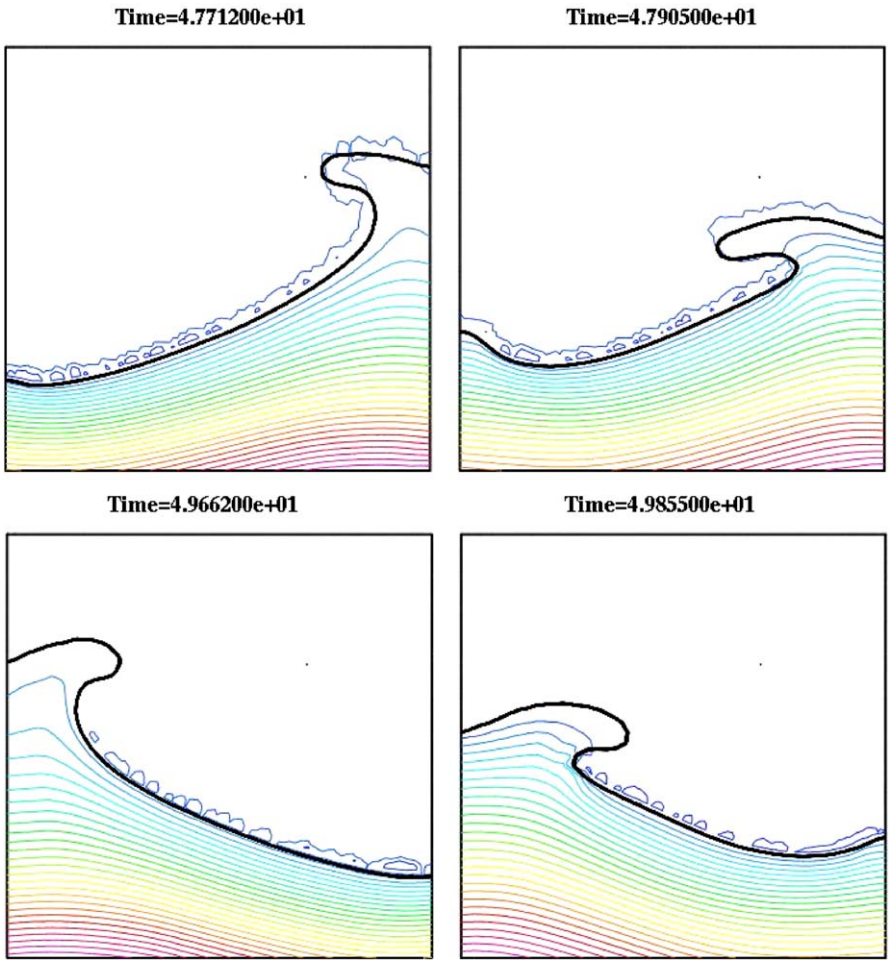


Fig. 8d. Snapshots of free surface wave elevation for $T = 1.3$ and $A/L = 0.05$.

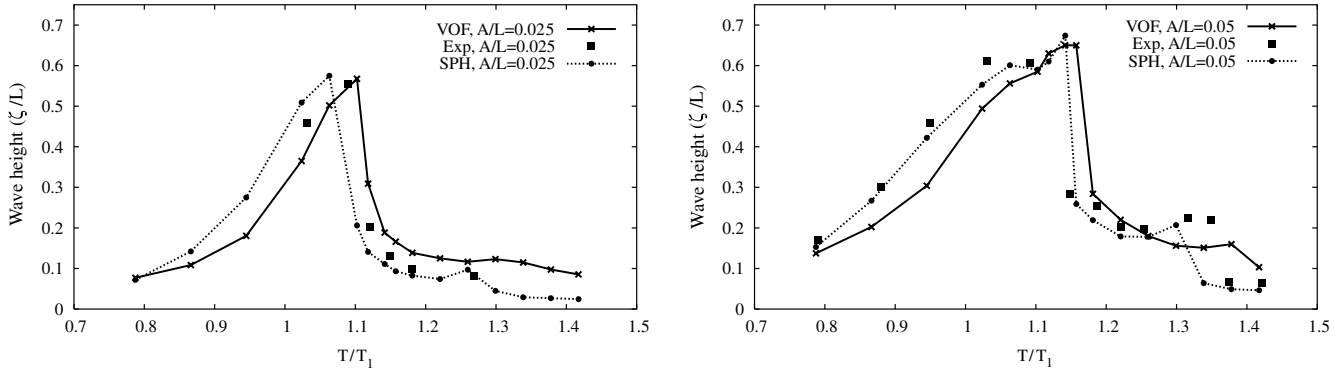


Fig. 8e. 2D tank: maximum wave height (Probe A1).

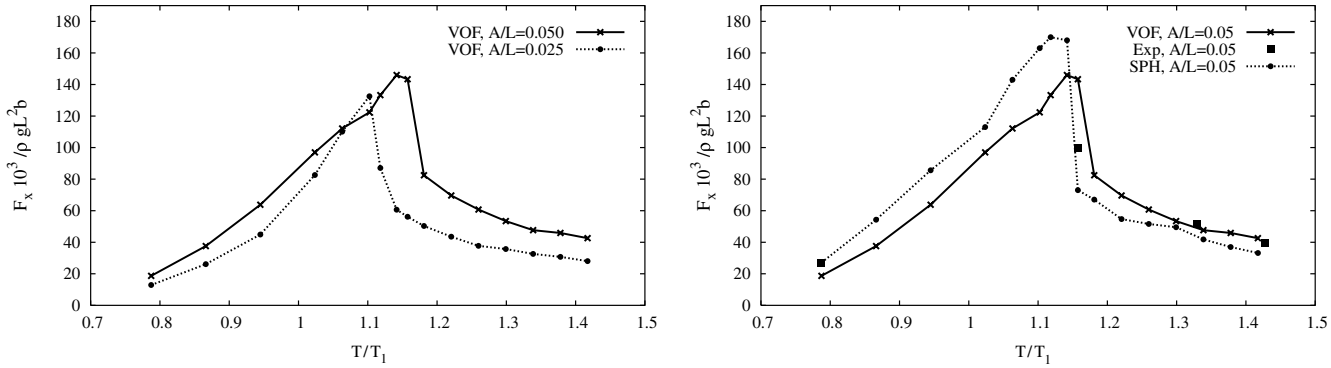


Fig. 8f. 2D tank: maximum absolute values of lateral force F_x for $A/L = 0.025, 0.05$.

free surface wave elevation, and Fig. 7c the time history of the horizontal force acting on the cylinder. The run took approximately 8 h on a Dell PC (3.2 GHz IP4, Linux OS, 2Gbyte RAM, Intel compiler).

6.3. Sloshing of a 2D tank due to sway excitation

This example considers the sloshing of a partially filled 2D tank. The main tank dimensions are $L = H = 1$ m, with tank width $B = 0.1$ m. The problem definition is shown in Fig. 8a. Experimental data for this tank with a filling level $h/L = 0.35$ have been provided by Olsen [46], and reported in Faltisen [14] and Olsen and Johnsen [47], where the tank was undergoing a sway motion, i.e., the tank oscillates horizontally with law $x = A\sin(2\pi t/T)$. A wave gauge was placed 0.05 m from the right wall and the maximum wave elevation relative to a tank-fixed coordinate system was recorded. In the numerical simulations reported by Landrini et al. [35] using the smoothed particle hydrodynamics (SPH) method, the forced oscillation amplitude increases smoothly in time and reaches its steady regime value in 10 T. The simulation continues for another 30 T and the maximum wave elevation is recorded in last 10 periods of oscillation.

We followed the same procedure as Landrini et al. [35] in our numerical simulation for 32 cases, which correspond to 2 amplitudes ($A = 0.025, 0.05$) and 16 periods, ranging from $T = 1.0$ to 1.8 s or $T/T_1 = 0.787$ to 1.42 s, where $T_1 = 1.27$ s. When $h/L = 0.35$ the primary resonances of the first and the third modes occur at $T/T_1 = 1.0$ and $T/T_1 = 0.55$, respectively. The secondary resonance of the second mode is at $T/T_1 = 1.28$ (see Landrini et al. [35]). The present VOF results for the time history of the lateral force F_x when $T = 1.2$, 1.3 and $A = 0.025$, 0.05 are shown in Fig. 8b. The corresponding time history of the wave elevation at the wave probe A1 (see Fig. 8a) is shown in Fig. 8c. Some free surface snapshots, together with pressure contours, are shown in Fig. 8d. Note the 'undershoot' of the pressure in the gas region due to

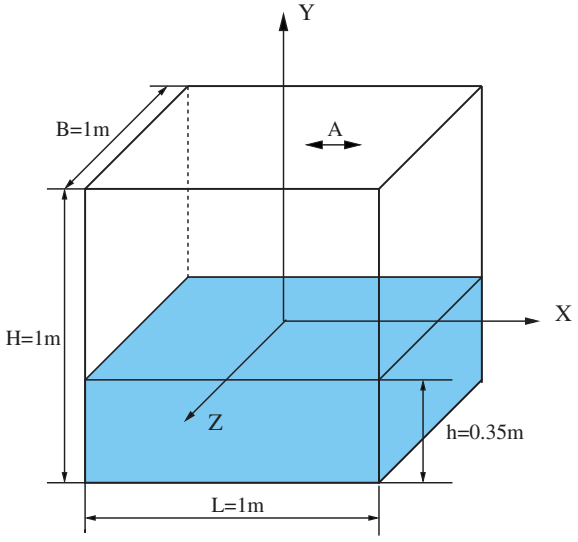


Fig. 9a. 3D tank: problem definition.

the extrapolation from the liquid region. The present VOF results for maximum wave elevation ζ at the wave probe A1 (see Fig. 8a) are compared with the experimental data and SPH results [35] in Fig. 8e for $A/L = 0.025, 0.05$.

The predicted lateral absolute values of maximum forces are compared with the experimental data and SPH results [35] in Fig. 8f for $A/L = 0.05$ (there is no force data available for $A/L = 0.025$). Fig. 8f shows the comparison of predicted lateral absolute values of maximum forces for $A/L = 0.025, 0.05$. It can be seen from Figs. 8e and 8f that both maximum wave height and lateral absolute values of maximum forces predicted by present VOF method agrees fairly well with the experimental data and SPH results, with a small phase shift among the three results. Figs. 8b and 8c are typical time history plots. It should be noted from these figures that even after a long simulation time (40 periods), steady-state results are not generally obtained. This is due to very small damping in the system. Landrini et al. [35] noted the same behavior in

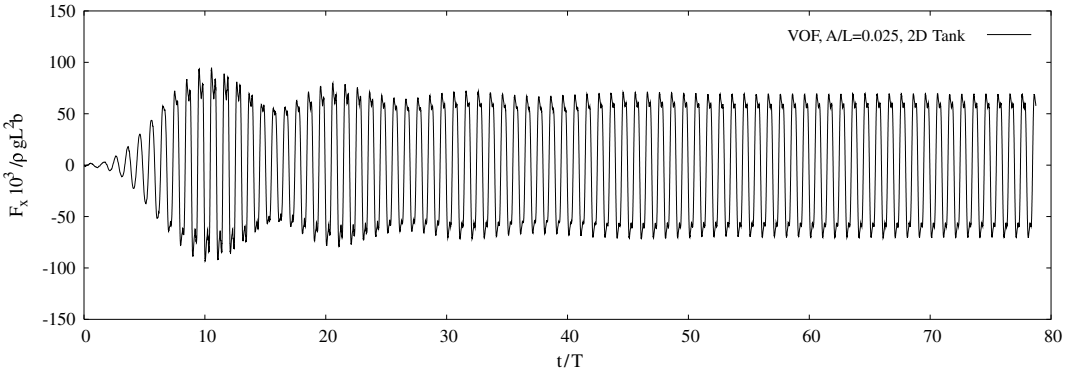


Fig. 9b. 3D tank: time history of force F_x for a 2D tank at $A/L = 0.025$, $T/T_1 = 1$.

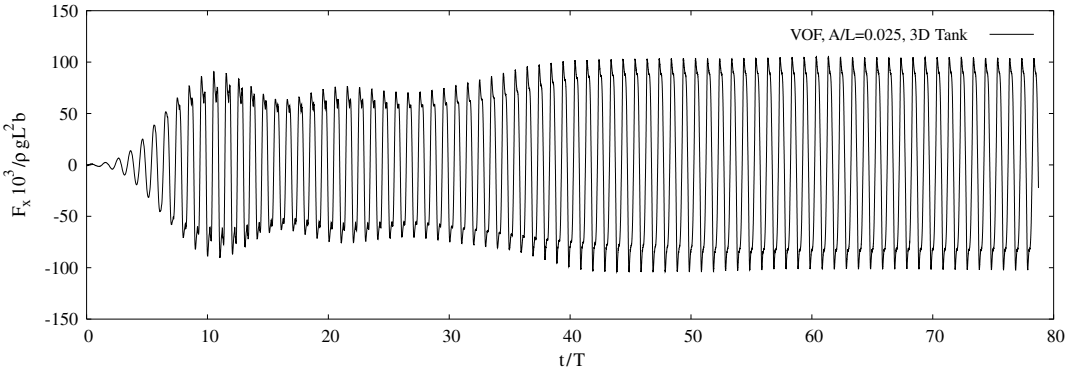


Fig. 9c. 3D tank: time history of force F_x for a 3D tank at $A/L = 0.025$, $T/T_1 = 1$.

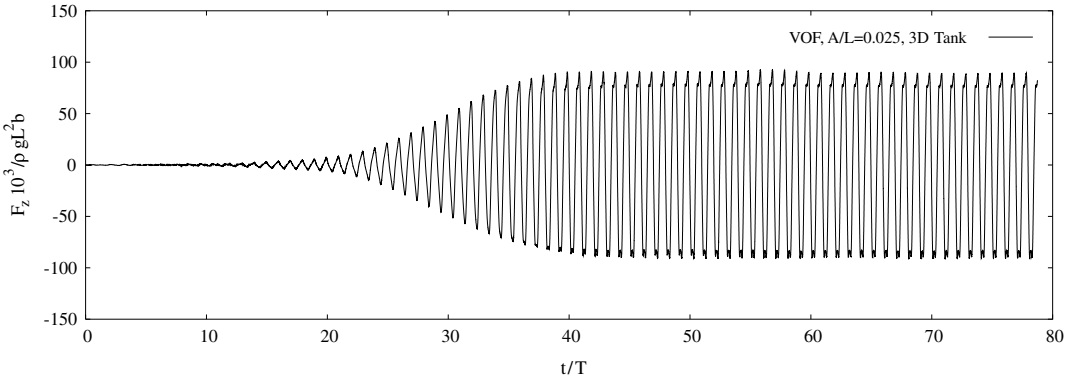


Fig. 9d. 3D tank: time history of force F_z for a 3D tank at $A/L = 0.025$, $T/T_1 = 1$.

their numerical simulations. As a result, the predicted maximum wave elevation and the lateral absolute values of maximum forces plotted in Fig. 8e are average maximum values for the last few periods for the cases when the steady-state is not reached.

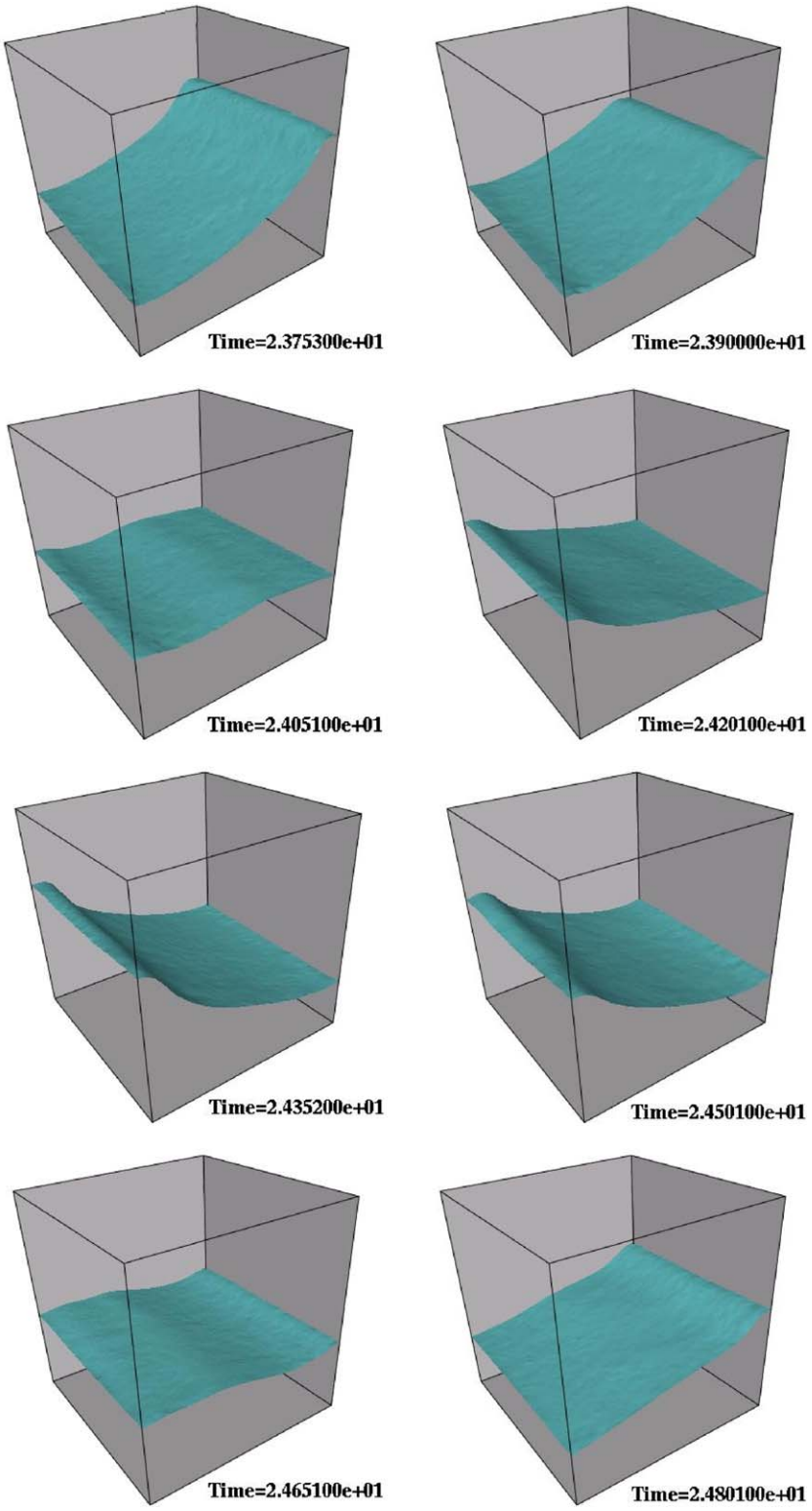


Fig. 9e. Snap shots of the free surface wave elevation for 3D tank.

6.4. Sloshing of a 3D tank due to sway excitation

In order to study the three-dimensional effects, the sloshing of a partially filled 3D tank is considered. The main tank dimensions are $L = H = 1$ m, with tank width $b = 1$ m. The problem definition is shown in Fig. 9a. The 3D tank has the

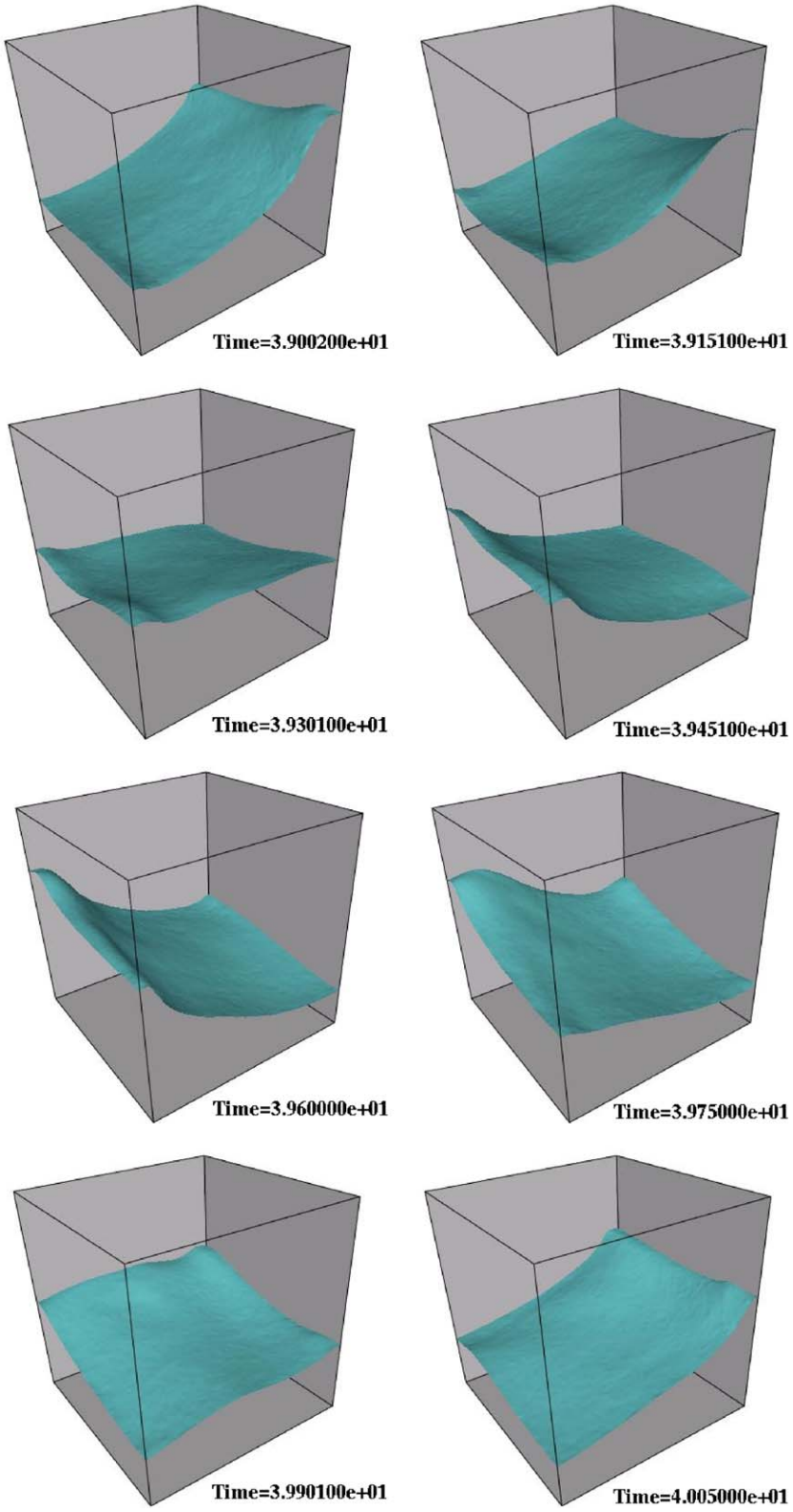


Fig. 9f. Snap shots of the free surface wave elevation for 3D tank.

same filling level $h/L = 0.35$ as the 2D tank. The 3D tank case is run on a mesh with `nelem = 561,808` elements, and the 2D tank is run on a mesh with `nelem = 54,124` elements. The numerical simulations are carried out for both 3D and 2D tanks, where both tanks are undergoing the same prescribed sway motion given by $x = A\sin(2\pi t/T)$. The simulations were

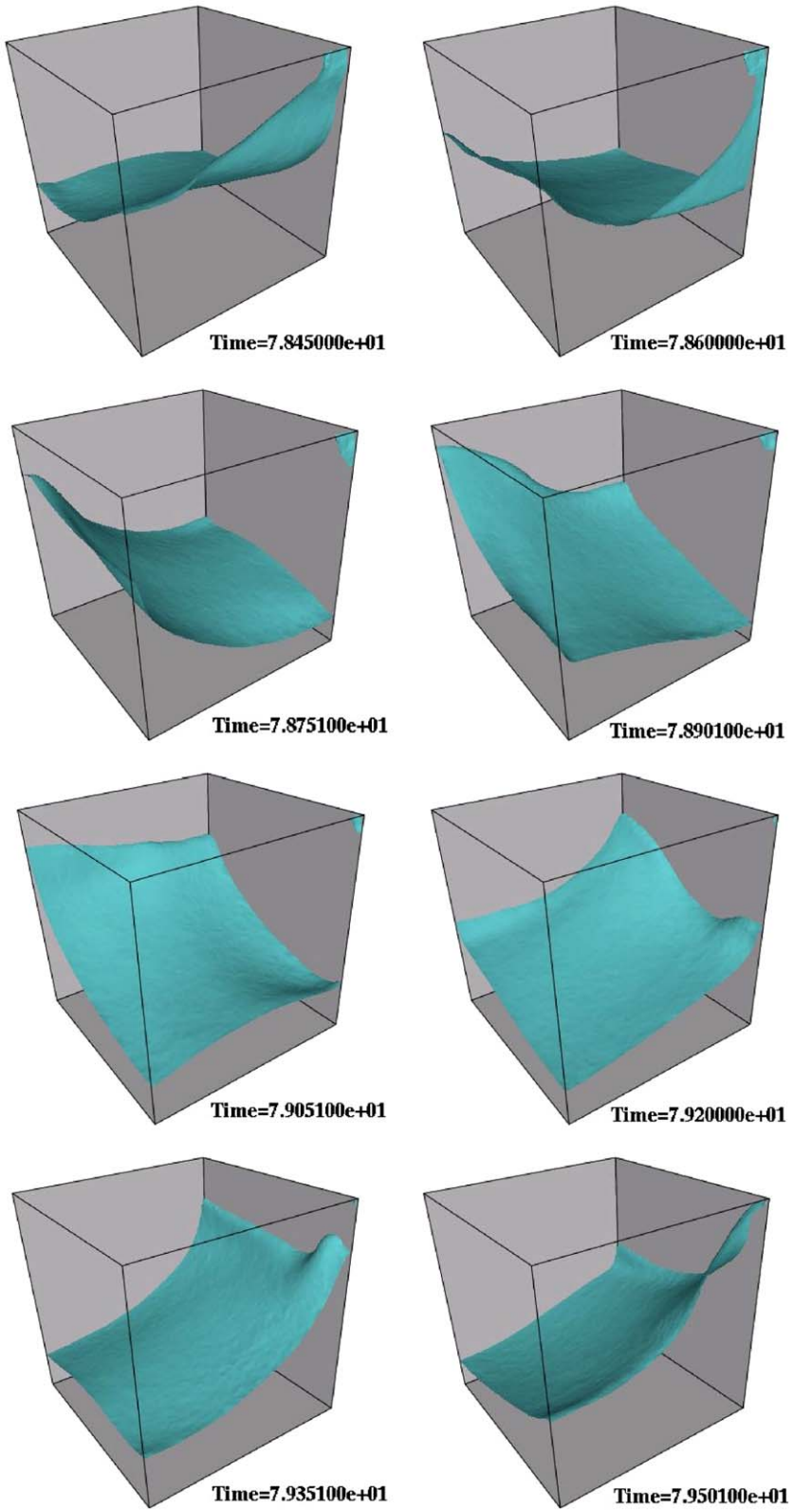


Fig. 9g. Snap shots of the free surface wave elevation for 3D tank.

carried out for $A = 0.025$ and $T = 1.27$ (i.e., $T/T_1 = 1$). The forced oscillation amplitude increases smoothly in time and reaches its steady regime value in 10 T. The simulation continues for another 70 T. In order to show the 3D effects, the forces are nondimensionalized with $\rho g L^2 b$ for both 2D and 3D tanks. Figs. 9b and 9c show the time history of the force F_x (horizontal force in the same direction as the tank moving direction) for both 2D and 3D tanks. Fig. 9d shows the time history of the force F_z (horizontal force perpendicular to the tank moving direction) for 3D tank. It is very interesting to

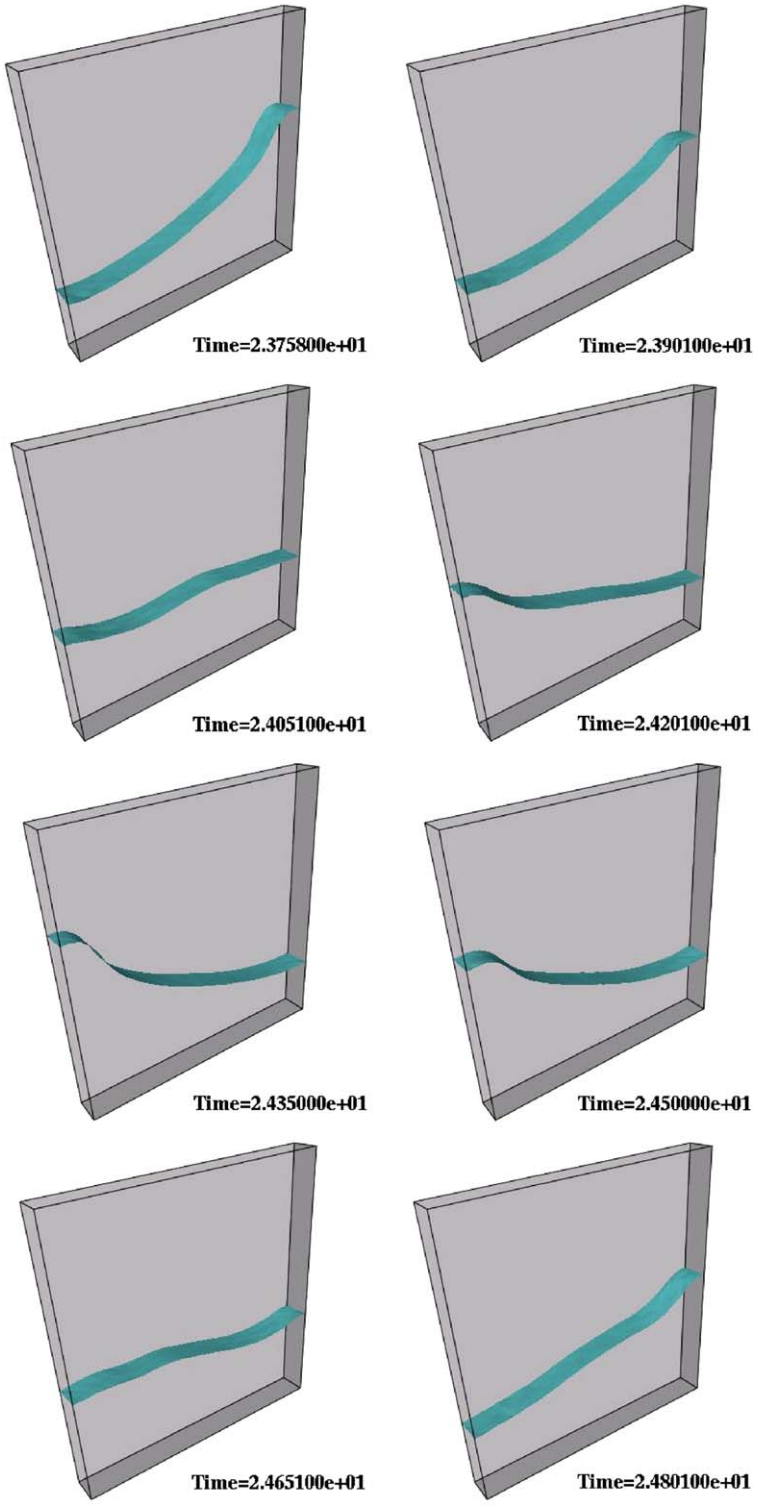


Fig. 9h. Snap shots of the free surface wave elevation for 2D tank.

observe from Figs. 9c and 9d that there are almost no 3D effects for the first 25 oscillating periods. The 3D modes start to appear after 25 T, and fully build up at about 40 T. The 3D flow pattern then remains steady and periodic for the rest of the simulation, which is about 40 more oscillating periods. CPU times were of the order of 8 h for the 2D cases and 36 h for the 3D cases on a Dell PC (3.2 GHz IP4, Linux OS, 2Gbyte RAM, Intel compiler).

Figs. 9e–9g show a sequence of snapshots of the free surface wave elevation for the 3D tank. For the first set of snapshots (see Fig. 9e), the flow is still two-dimensional. The 3D flow starts to build up in the second set of snapshots (see Fig. 9f). The flow remains periodic three-dimensional for the last 40 periods. Fig. 9g show the typical snapshots of the free surface for the last 40 periods. The 3D effects are clearly shown in these plots. Fig. 9h shows a sequence of snapshots of the free surface wave elevation for the 2D tank at the same time instance as those shown in Fig. 9e. The flow remains periodic and two-dimensional for the rest of the simulation.

6.5. Sinking tank

This test case is included to show the combination of the present techniques with mesh movement and remeshing. The problem definition is given in Fig. 10a. The tank is assumed filled with air, and a hole opens up at the bottom of one of the compartments. The mass of the tank was estimated by assuming that when floating empty, half of the tank is outside the water. This yielded approximately $m = 15{,}700$ kg. The moment of inertia was estimated at $\Theta_z = 80{,}000$ kg m^2. The filling of the tank, and the ensuing movement, can be seen from Figs. 10b–f, which show the free surface, velocity and mesh in the plane of symmetry as the calculation proceeds. Note the deformation of the mesh during the run, as well as the effect of the (only 4) automatic global remeshings required to guarantee a proper mesh. The mesh had approximately `nelem = 530,000` elements. The position and velocity of the center of mass as a function of time, as well as the trajectory are summarized in Figs. 10g–h. The run took approximately 3 h on a Dell PC (3.2 GHz IP4, Linux OS, 2Gbyte RAM, Intel compiler).

6.6. Bubble collapse beneath generic ship

This example shows the use of the present methodology to predict the effects of bubble collapse close to structures. The problem definition is given in Fig. 11a. The ship is a generic ferry. The reference values for the bubble, which was located at mid-ship and approximately 4 m from the hull, were set as follows: volume $V_0 = 128$ m^3, density $\rho_0 = 1.25$ kg/m^3, pressure $p_0 = 1.0\text{e}+8$ N/m^2, polytropic coefficient: $\gamma = 1.4$. The initial radius for the bubble was set to $r = 2$ m. The mesh had approximately `nelem = 1,530,000` elements. Figs. 11b–g show the evolution of the flowfield. Note the change of shape for the bubble, first into a torus and subsequently into a rather complex shape. The pressure recorded at midship on the hull is shown in Fig. 11h. CPU time was of the order of 9 h on a Dell PC (3.2 GHz IP4, Linux OS, 2 Gbyte RAM, Intel compiler).

7. Conclusions and outlook

A volume of fluid (VOF) technique has been developed and coupled with an incompressible Euler/Navier–Stokes solver operating on adaptive, unstructured grids to simulate the interactions of extreme waves and three-dimensional structures.

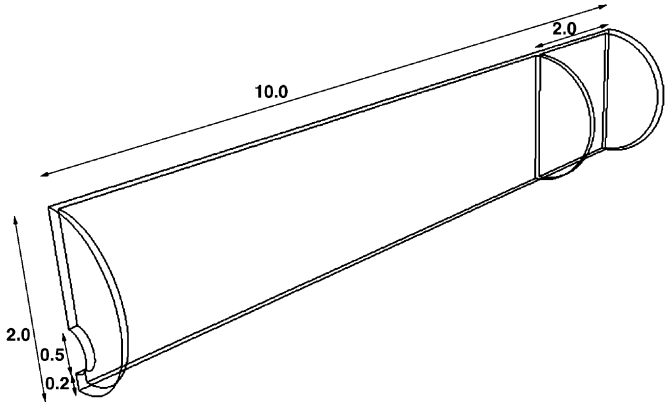
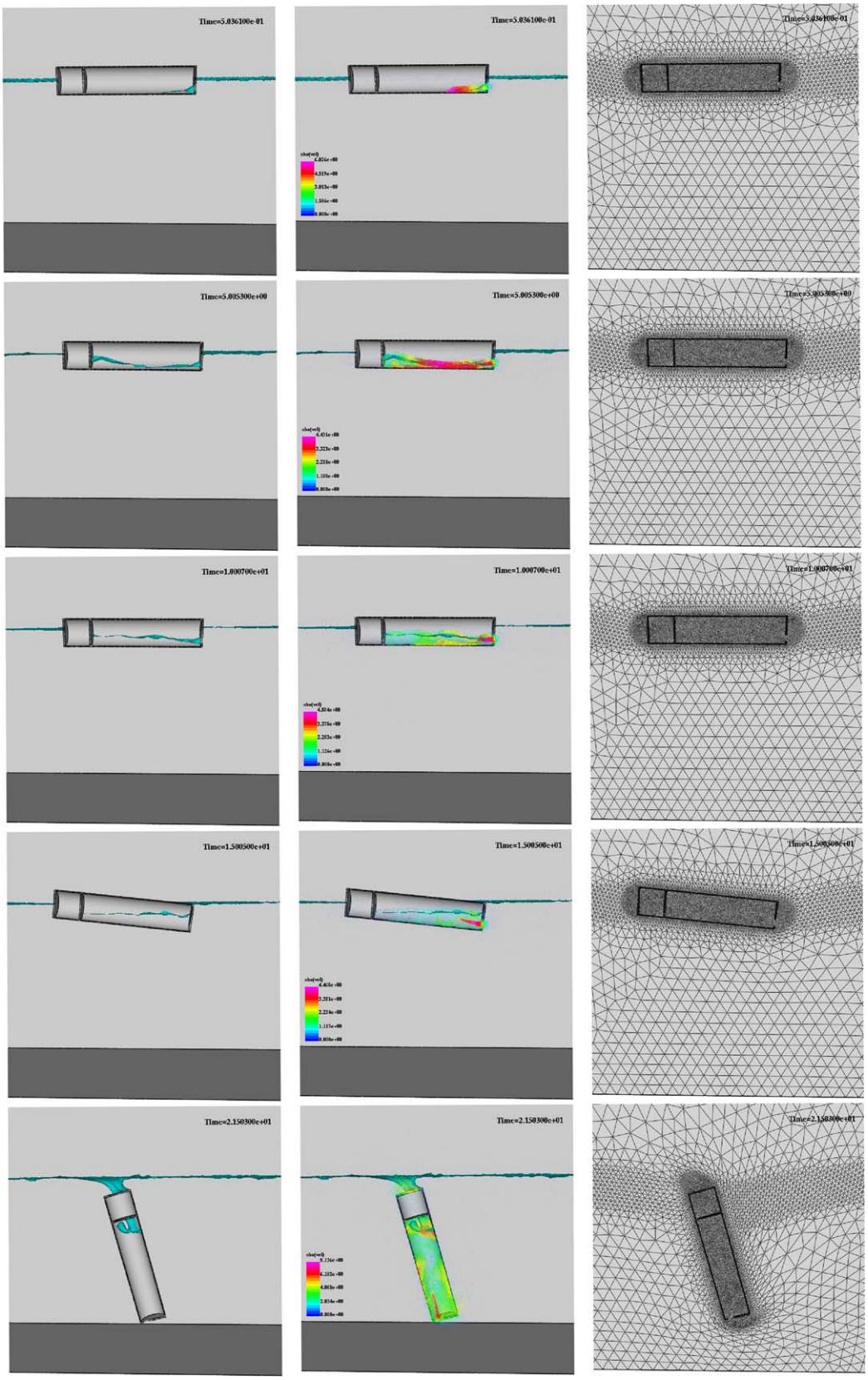
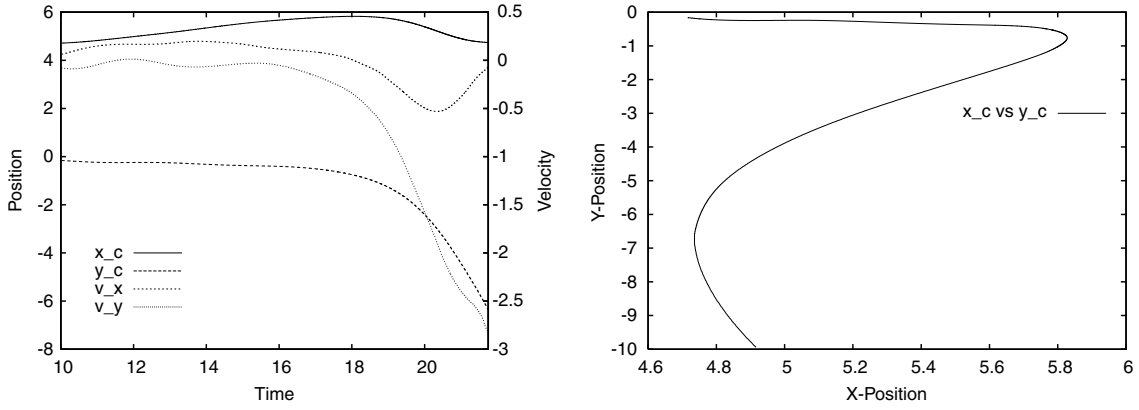


Fig. 10a. Sinking tank: problem definition.



Figs. 10b–f. Sinking tank: flowfield at different times.



Figs. 10g,h. Sinking tank: trajectory of center of mass.

The present implementation follows the classic VOF implementation for the liquid–gas system, considering only the liquid phase. Extrapolation algorithms to obtain velocities and pressure in the gas region near the free surface have been implemented. The VOF technique was validated against the classic dam-break problem, as well as series of 2D sloshing experiments and results from SPH calculations. Other examples presented include violent wave interaction with a column, a leaky tank filling with water and sinking, and a bubble collapsing under a generic ship.

When taken together, these recent advances, which include:

- accurate, fast incompressible Navier–Stokes solvers operating on adaptive, unstructured grids;
- robust volume of fluid (VOF) techniques for free surface flows;
- deactivation techniques to speed up calculations; and
- extensive parallelization of solvers

have made it possible to simulate flows with violent free surface motion with a high degree of accuracy, allowing decision-making based on them. Like every human endeavour, numerical algorithms are subject to continuous improvements. Present research is directed at the proper treatment of:

- surface tension;
- incoming and outgoing waves for 3D VOF-based free surface flows;
- free surface wall boundary conditions for RANS, NS cases (i.e., those cases where the velocity at the wall $\mathbf{v} = 0$); and
- multiple bubble interaction (splitting, merging, etc.).

Acknowledgement

A considerable part of this work was carried out at the International Center for Numerical Methods in Engineering (CIMNE) of the Universidad Politécnica de Catalunya, Barcelona, Spain. The support for this visit is gratefully acknowledged.

The authors also wish to thank Andrea Colagrossi of the INSEAN for providing both experimental data and numerical results for the sloshing of a 2D tank.

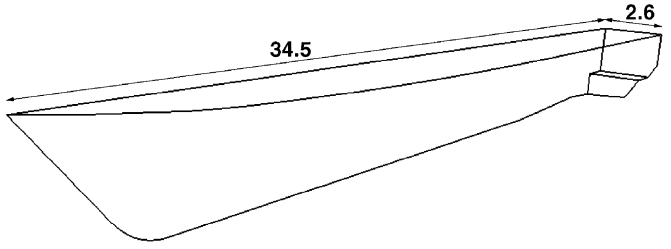
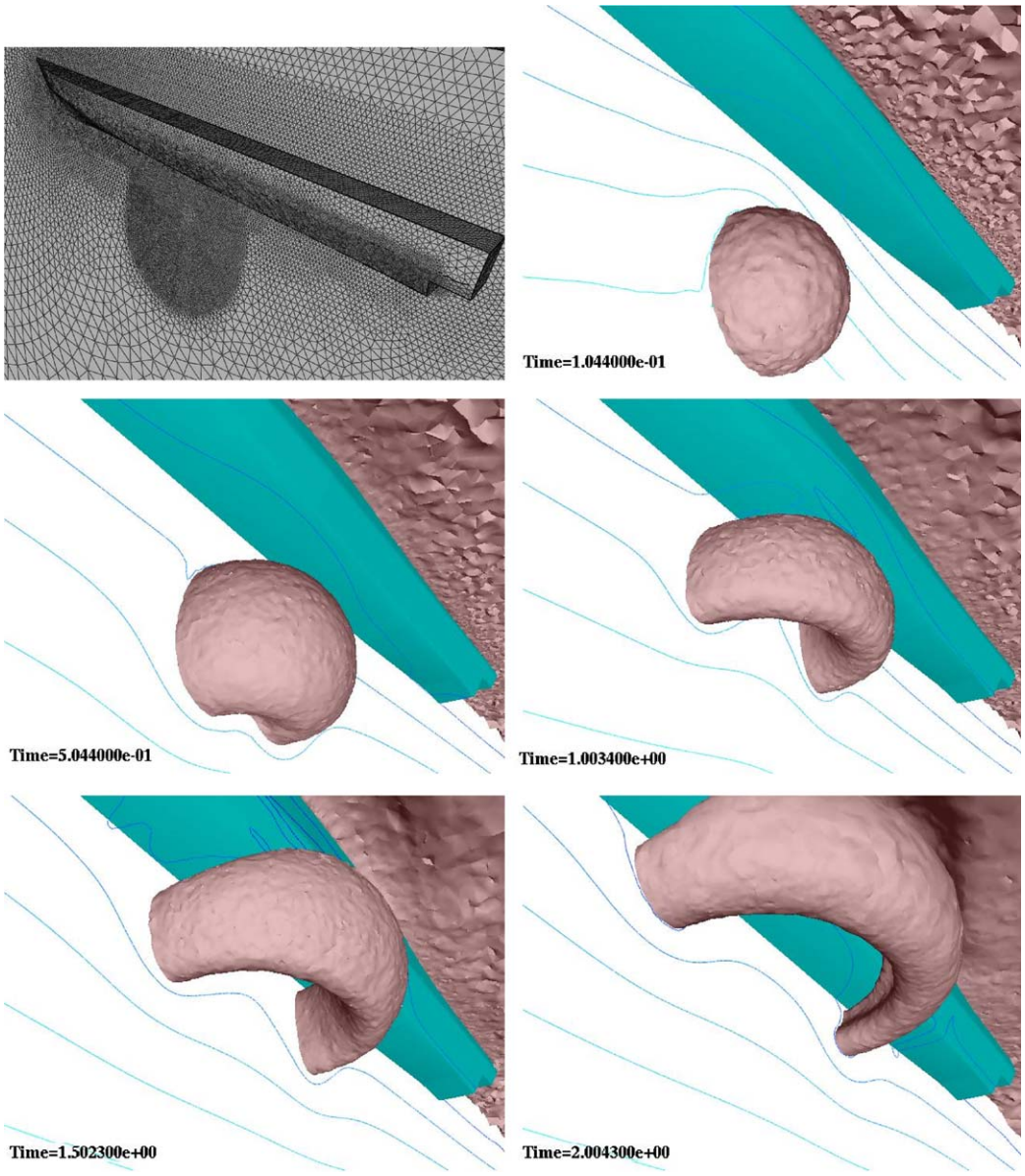


Fig. 11a. Bubble collapse: problem definition.



Figs. 11b–g. Surface mesh and evolution of bubble.

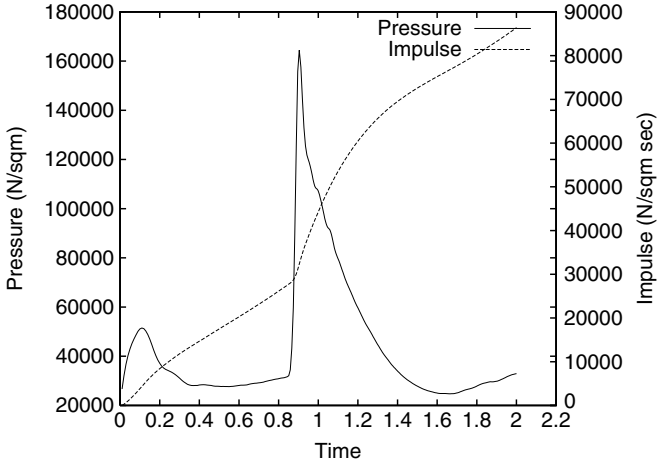


Fig. 11h. Pressure and impulse recorded at midship on Hull.

References

[1] B. Alessandrini, G. Delhommeau, A multigrid velocity–pressure – free surface elevation fully coupled solver for calculation of turbulent incompressible flow around a Hull, in: Proceedings of the 21st Symposium on Naval Hydrodynamics, Trondheim, Norway, June, 1996.
[2] J.B. Bell, P. Colella, H. Glaz, A second-order projection method for the Navier–Stokes equations, J. Comput. Phys. 85 (1989) 257–283.
[3] J.B. Bell, D.L. Marcus, A second-order projection method for variable-density flows, J. Comput. Phys. 101 (1992) 2.
[4] P. Bergan, J. Garcia, E. Oñate, T. Kvamsdal (Eds.), Computational Methods in Marine Engineering, CIMNE, Barcelona, 2005.
[5] B. Biausser, P. Fraunie, S. Grilli, R. Marcer, Numerical analysis of the internal kinematics and dynamics of three-dimensional breaking waves on slopes, Int. J. Offshore Polar Engrg. 14 (2004) 4.
[6] A.N. Brooks, T.J.R. Hughes, Streamline upwind/petrov Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.
[7] F. Camelli, R. Löhner, W.C. Sandberg, R. Ramamurti, VLES Study of Ship Stack Gas Dynamics, AIAA-04-0072, 2004.
[8] G. Chen, C. Kharif, Two-dimensional Navier–Stokes simulation of breaking waves, Phys. Fluids 11 (1) (1999) 121–133.
[9] R. Codina, Pressure stability in fractional step finite element methods for incompressible flows, J. Comput. Phys. 170 (2001) 112–140.
[10] R. Codina, O. Soto, A numerical model to track two-fluid interfaces based on a stabilized finite element method and the level set technique, Int. J. Numer. Methods Fluids 4 (2002) 293–301.
[11] A.H. Coppola-Owen, R. Codina, Improving Eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions, Int. J. Numer. Methods Fluids 49 (12) (2005) 1287–1304.
[12] J. Donea, S. Giuliani, H. Laval, L. Quartapelle, Solution of the unsteady Navier–Stokes equations by a fractional step method, Comput. Methods Appl. Mech. Engrg. 30 (1982) 53–73.
[13] E. Eaton, in: Aero-Acoustics in an Automotive HVAC Module American PAM User Conference, Birmingham, MI, October 24–25, 2001.
[14] O.M. Faltisen, A nonlinear theory of sloshing in rectangular tanks, J. Ship Res. 18/4 (1974) 224–241.
[15] J.R. Farmer, L. Martinelli, A. Jameson, A fast multigrid method for solving incompressible hydrodynamic problems with free surfaces, AIAA J. 32 (6) (1993) 1175–1182.
[16] D. Enright, D. Nguyen, F. Gibou, R. Fedkiw, Using the particle level set method and a second order accurate pressure boundary condition for free surface flows, in: M. Kawahashi, A. Ogut, Y. Tsuji (Eds.), Proceedings of the 4th ASME–JSME Joint Fluids Engineering Conference FEDSM2003-45144, Honolulu, HI, 2003, pp. 1–6.
[17] G. Fekken, A.E.P. Veldman, B. Buchner, Simulation of green water loading using the Navier–Stokes equations, in: Proceedings of the 7th International Conference on Numerical Ship Hydrodynamics, Nantes, France, 1999.
[18] M. Fortin, F. Thomasset, Mixed finite element methods for incompressible flow problems, J. Comput. Phys. 31 (1979) 113–145.
[19] L.P. Franca, T.J.R. Hughes, A.F.D. Loula, I. Miranda, A new family of stable elements for the stokes problem based on a mixed Galerkin/least-squares finite element formulation, in: T.J. Chung, G. Karr (Eds.), Proceedings of the 7th International Conference Finite Elements in Flow Problems, Huntsville, AL, 1989, pp. 1067–1074.
[20] L.P. Franca, S.L. Frey, Stabilized finite element methods: II. The incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 99 (1992) 209–233.
[21] P.M. Gresho, C.D. Upson, S.T. Chan, R.L. Lee, Recent progress in the solution of the time-dependent, three-dimensional, incompressible Navier–Stokes equations, in: T. Kawai (Ed.), Proceedings of the 4th International Symposium on Finite Element Methods in Flow Problems, University of Tokyo Press, 1982, pp. 153–162.
[22] J.P. Gregoire, J.P. Benque, P. Lasbleiz, J. Goussebaile, 3-D industrial flow calculations by finite element method, Springer Lect. Notes Phys. 218 (1985) 245–249.
[23] P.M. Gresho, S.T. Chan, On the theory of semi-implicit projection methods for viscous incompressible flows and its implementation via a finite element method that introduces a nearly-consistent mass matrix, Int. J. Numer. Methods Fluids 11 (1990) 621–659.
[24] M.D. Gunzburger, Mathematical aspects of finite element methods for incompressible viscous flows, in: D. Dwoyer, M. Hussaini, R. Voigt (Eds.), Finite Elements: Theory and Application, Springer-Verlag, Berlin, 1987, pp. 124–150.
[25] P. Hansbo, The characteristic streamline diffusion method for the time-dependent incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 99 (1992) 171–186.
[26] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201.
[27] J.D. Huffenus, D. Khaletzky, A finite element method to solve the Navier–Stokes equations using the method of characteristics, Int. J. Numer. Methods Fluid 4 (1984) 247–269.
[28] R.H.M. Huijsmans, E. van Grosen, Coupling freak wave effects with green water simulations, in: Proceedings of the 14th ISOPE, Toulon, France, May 23–28, 2004.
[29] Y. Kallinderis, A. Chen, An Incompressible 3-D Navier–Stokes Method with Adaptive Hybrid Grids, AIAA-96-0293, 1996.
[30] K.J. Karbon, S. Kumarasamy, Computational aeroacoustics in automotive design, computational fluid and solid mechanics, in: Proceedings of the 1st MIT Conference on Computational Fluid and Solid Mechanics, Boston, June, 2001, pp. 871–875.
[31] K.J. Karbon, R. Singh, Simulation and design of automobile sunroof buffeting noise control, in: 8th AIAA-CEAS Aero-Acoustics Conference, Brenckridge, June, 2002.
[32] D.W. Kelly, S. Nakazawa, O.C. Zienkiewicz, J.C. Heinrich, A note on anisotropic balancing dissipation in finite element approximation to convection diffusion problems, Int. J. Numer. Methods Engrg. 15 (1980) 1705–1711.
[33] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.
[34] A. Kölke, Modellierung und Diskretisierung bewegter Diskontinuitäten in Randgekoppelten Mehrfeldaufgaben, Ph.D. Thesis, TU Braunschweig, 2005.
[35] M. Landrini, A. Colagorossi, O.M. Faltisen, Sloshing in 2-D flows by the sph method, in: Proceedings of the 8th International Conference on Numerical Ship Hydrodynamics, Busan, Korea, 2003.
[36] Y. Li, T. Kamioka, T. Nouzawa, T. Nakamura, Y. Okada, N. Ichikawa, Verification of aerodynamic noise simulation by modifying automobile front-pillar shape, JSAE 20025351, JSAE Annual Conference, Tokyo, July, 2002.
[37] R. Löhner, A Fast Finite Element Solver for Incompressible Flows, AIAA-90-0398, 1990.
[38] R. Löhner, Some useful renumbering strategies for unstructured grids, Int. J. Numer. Methods Engrg. 36 (1993) 3259–3270.

[39] R. Löhner, Renumbering strategies for unstructured-grid solvers operating on shared-memory, cache-based parallel machines, Comput. Methods Appl. Mech. Engrg. 163 (1998) 95–109.
[40] R. Löhner, C. Yang, E. Oñate, S. Idelssohn, An unstructured grid-based, parallel free surface solver, Appl. Numer. Math. 31 (1999) 271–293.
[41] R. Löhner, Applied CFD Techniques, Wiley, New York, 2001.
[42] R. Löhner, Multistage explicit advective prediction for projection-type incompressible flow solvers, J. Comput. Phys. 195 (2004) 143–152.
[43] D. Martin, R. Löhner, An Implicit Linelet-Based Solver for Incompressible Flows, AIAA-92-0668, 1992.
[44] J.C. Martin, W.J. Moyce, An experimental study of the collapse of a liquid column on a rigid horizontal plane, Phil. Trans. Roy. Soc. Lond. A244 (1952) 312–324.
[45] B.D. Nichols, C.W. Hirt, Methods for calculating multi-dimensional, transient free surface flows past bodies, in: Proceedings of the 1st International Conference on Numerical Ship Hydrodynamics, Gaithersburg, ML, October 20–23, 1975.
[46] H. Olsen, Unpublished Sloshing Experiments at the Technical University of Delft, Delft, The Netherlands, 1970.
[47] H. Olsen, K.R. Johnsen, Nonlinear sloshing in rectangular tanks, A pilot study on the applicability of analytical models, Det Norske Veritas Rep. 74-72-S, vol. II, 1975.
[48] R. Ramamurti, R. Löhner, A parallel implicit incompressible flow solver using unstructured meshes, Comput. Fluids 5 (1996) 119–132.
[49] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[50] A. Soulaimani, M. Fortin, Y. Ouellet, G. Dhatt, F. Bertrand, Simple continuous pressure elements for two- and three-dimensional incompressible flows, Comput. Methods Appl. Mech. Engrg. 62 (1987) 47–69.
[51] M. Sussman, P. Smereka, S. Osher, A levelset approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[52] M. Sussmam, E. Puckett, A coupled level set and volume of fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301–337.
[53] A. Takamura, M. Zhu, D. Vinteler, Numerical simulation of pass-by maneuver using ALE technique, in: JSAE Annual Conference (Spring), Tokyo, May, 2001.
[54] C. Taylor, P. Hood, A numerical solution of the Navier–Stokes equations using the finite element method, Comp. Fluids 1 (1973) 73–100.
[55] T.E. Tezduyar, R. Shih, S. Mittal, S.E. Ray, Incompressible Flow Computations With Stabilized Bilinear and Linear Equal-Order Interpolation Velocity–Pressure Elements, UMSI Rep. 90, 1990.
[56] F. Thomasset, Implementation of Finite Element Methods for Navier–Stokes Equations, Springer-Verlag, Berlin, 1981.
[57] S.O. Unverdi, G. Tryggvason, A front tracking method for viscous incompressible flows, J. Comput. Phys. 100 (1992) 25–37.
[58] E. Walhorn, Ein Simultanes Berechnungsverfahren für Fluid- Struktur- Wechselwirkungen mit Finiten Raum- Zeit- Elementen, Ph.D. Thesis, TU Braunschweig, 2002.
[59] T. Yabe, T. Aoki, A universal solver for hyperbolic equations by cubic-polynomial interpolation, Comput. Phys. Commun. 66 (1991) 219–242.
[60] T. Yabe, Universal solver CIP for solid, liquid and gas, CFD Rev. 3 (1997).