

OMAE2005-67422

DEVELOPMENT OF A CFD SIMULATION METHOD FOR EXTREME WAVE AND STRUCTURE INTERACTIONS

Chi Yang¹

School of Computational Sciences
George Mason University
Fairfax, VA 22030
E-mail: cyang@gmu.edu

Rainald Löhner

School of Computational Sciences
George Mason University
Fairfax, VA 22030
E-mail: rlohner@gmu.edu

Solomon C. Yim

Dept. of Civil, Construction and
Environmental Engineering
Oregon State University
Corvallis, OR 97331
E-mail: solomon.yim@oregonstate.edu

ABSTRACT

A robust Volume of Fluid (VOF) technique is presented together with an incompressible Euler/Navier Stokes solver operating on adaptive, unstructured grids to simulate the interactions of extreme waves and three-dimensional structures. The incompressible Euler/Navier Stokes equations are solved using projection schemes and a finite element method. The classic breaking dam problem has been used to validate the computer code developed based on the method described above. The numerical simulations of a three dimensional dam-break wave interacting with a single cylinder and a cylinder array have been carried out. Computational results have demonstrated that the present CFD method is capable of simulating the interactions of extreme waves and three-dimensional structures, which are of great importance for the comprehension of many natural phenomena in marine, coastal and marine engineering.

INTRODUCTION

The knowledge of the mechanism responsible for the breaking of waves is of great importance for the comprehension of many natural phenomena and the development of several engineering processes. There are so many problems related with breaking waves. Breaking wave and structure interactions are especially important to the ocean and marine engineering. The entire interaction process includes the initiation of wave breaking, breaking, and wave interactions with surroundings (Rhee and Stern, 2002). With the rapid development of computer hardware and software, numerical studies of highly nonlinear free surface flows, including

breaking waves, have become increasingly popular.

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 process. There are basically two approaches to compute flows with free surface: interface-tracking and interface-capturing methods. The former compute 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 using the finite element method. 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 can not be used if the interface topology changes significantly (e.g. overturning or breaking waves).

On the other hand, interface-capturing methods consider both fluids as a single effective fluid with variable properties; the interface is captured as a region of sudden change in fluid properties. Either massless particles or an indicator function marks gas or fluid on either side of the interface. Among various interface-capturing methods, volume-of-fluid (VOF) methods and level-set (LS) methods are more economical than marker particles, as only one value (the volume fraction for VOF or the level-set function for LS) needs to be assigned to each mesh cell. Another ben-

¹Address all correspondence to this author.

enefit of using volume fractions or level-set functions is that only a scalar convection equation needs to be solved to propagate the volume fractions or level-set functions through the computational domain. The interface-capturing methods based on the Eulerian approach require no geometry manipulations after the mesh is generated and can be applied to interfaces of a complex topology such as overturning or breaking waves.

As the objective of this study is to model the highly nonlinear free surface flows, one of the most promising interface-capturing methods – VOF method is adopted and further developed. The VOF method was first reported in Nichols and Hirt (1975), and more completely in Hirt and Nichols (1981). This method has been improved in several aspects in the recent years (e.g. Scardovelli and Zaleski, 1999) and used to simulate breaking waves (e.g. Chen and Kharif, 1999; Biauasser et al., 2004), green water effects (e.g. Fekken et al., 1999; Huijsmans and van Groesen, 2004), and sloshing (e.g. Huijsmans et al., 2004). In the present study, an unstructured grid based solver for the incompressible Navier-Stokes equations has been extended to handle the highly nonlinear free surface flows via the VOF techniques. A fixed grid is used which covers the space occupied by both the liquid and the gas phase. Since the grid does not follow the deformation of the free surface, the grid movement is only necessary if the shape or location of the solid boundary changes (e.g. in the case of sloshing, slamming, floating bodies etc.). Only the liquid phase is simulated. An extrapolation algorithm is developed for obtaining the pressure and velocity in the gas region.

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

BASIC ELEMENTS OF THE SOLVER

The equations describing incompressible, Newtonian flows may be written as

$$\rho \mathbf{v}_{,t} + \rho \mathbf{v} \nabla \mathbf{v} + \nabla p = \nabla \mu \nabla \mathbf{v} + \rho \mathbf{g} \quad , \quad (1)$$

$$\nabla \cdot \mathbf{v} = 0 \quad . \quad (2)$$

Here ρ denotes the density, \mathbf{v} the velocity vector, p the pressure, μ the viscosity and \mathbf{g} the gravity vector. We remark

that both the gaseous and liquid phases are considered incompressible, thus Eqn.(2). The liquid-gas interface is described by a scalar equation of the form:

$$\Phi_{,t} + \mathbf{v} \cdot \nabla \Phi = 0 \quad . \quad (3)$$

For the classic VOF technique, Φ represents the total density of the material in a cell/element or control volume. For pseudo-concentration techniques, Φ represents the percentage of liquid in a cell/element or control volume. For the level set approach Φ represents the signed distance to the interface.

Since over a decade the numerical schemes chosen to solve the incompressible Navier-Stokes equations given by Eqns.(1,2) 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 3-D problems); and
- Steady results that are independent from the timestep chosen (in order to have confidence in convergence studies).

A detailed description of the numerical solution procedure can be found in authors' previous work (Löhner, 1990; Löhner et al., 1998; Yang and Löhner, 1998; Löhner et al., 1999).

TEMPORAL AND SPATIAL DISCRETIZATIONS

In this section, we will discuss the temporal and spatial discretizations used in the present unstructured grid based solver for the incompressible Navier-Stokes equations.

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:

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

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: $\mathbf{v}^n \rightarrow \mathbf{v}^*$

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

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

$$\nabla \cdot \mathbf{v}^{n+1} = 0 ; \quad (6)$$

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

which results in

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

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

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

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 ($\theta = 1$: 1st order, fully implicit, $\theta = 0.5$: 2nd order, Crank-Nicholson). One can replace the one-step explicit advective-diffusive predictor by a multistage Runge-Kutta scheme (Löhner, 2004), 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 of order k for the advective parts may be written as:

$$\begin{aligned} \rho \mathbf{v}^i &= \rho \mathbf{v}^n + \\ \alpha^i \gamma \Delta t & \left(-\rho \mathbf{v}^{i-1} \cdot \nabla \mathbf{v}^{i-1} - \nabla p^n + \nabla \mu \nabla \mathbf{v}^{i-1} + \rho \mathbf{g} \right) ; \\ & i = 1, k-1 ; \end{aligned} \quad (10)$$

$$\begin{aligned} \left[\frac{\rho}{\Delta t} - \theta \nabla \mu \nabla \right] (\mathbf{v}^k - \mathbf{v}^n) + \rho \mathbf{v}^{k-1} \cdot \nabla \mathbf{v}^{k-1} + \nabla p^n \\ = \nabla \mu \nabla \mathbf{v}^{k-1} + \rho \mathbf{g} . \end{aligned} \quad (11)$$

Here, the α^i are the standard Runge-Kutta coefficients $\alpha^i = 1/(k+1-i)$. As compared to the original scheme given by Eqn.(5), the $k-1$ stages of Eqn.(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:

$$\Delta t_a \approx \frac{h}{|\mathbf{v}|} ; \quad \Delta t_v \approx \frac{\rho h^2}{\mu} , \quad (12)$$

we immediately obtain

$$\gamma = \frac{\Delta t_v}{\Delta t_a} \approx \frac{\rho |\mathbf{v}| h}{\mu} \approx Re_h , \quad (13)$$

or, in its final form:

$$\gamma = \min(1, Re_h) . \quad (14)$$

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 (Eqn.(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 (Kim and Moin, 1985; Bell et al. 1989; Bell and Marcus, 1992; Alessandrini and Delhommeau, 1996), finite volumes (Kallinderis and Chen, 1996), and finite elements (Löhner, 1990; Martin and Löhner, 1992; Ramamurti and Löhner, 1996; Löhner, 1999, Takamura et al., 2001; Eaton, 2001; Karbon and Kumarasamy, 2001; Codina, 2001; Li, et al. 2002; Karbon and Singh, 2002).

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

- Predict velocity (advective-diffusive predictor, Eqns.(5,10));
- Extrapolate the pressure (imposition of boundary conditions);
- Update the pressure (Eqn.(8));
- Correct the velocity field (Eqn.(9));
- Extrapolate the Velocity Field; and
- Update the scalar interface indicator.

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 Eqns.(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. The edge-based upwinding scheme (Löhner et al., 1999) is adopted to modify (or stabilize) the Galerkin discretization of the advection terms. 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 Eqn.(3). For the divergence constraint given by Eqn.(2), the approach taken here is based on consistent numerical fluxes, as it fits naturally into the edge-based framework (Löhner et al., 1999).

The detailed discussion about the spatial discretization can be found from the authors' previous work (Löhner et al., 1999).

VOLUME OF FLUID EXTENSIONS

The extension of a solver for the incompressible Navier-Stokes equations to handle free surface flows via the VOF or level set techniques requires a series of extensions which are the subject of the present section.

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 points 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 the 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 Figure 1). This step is required as the pressure gradient for these points can not 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.

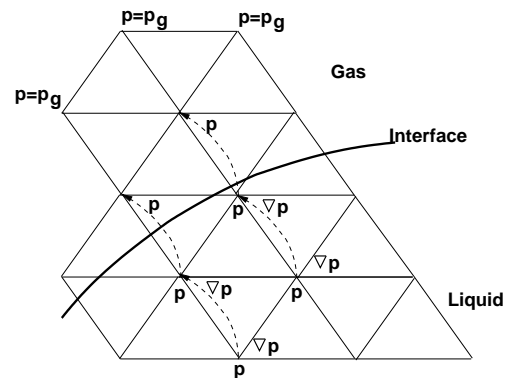


Figure 1: Extrapolation of the Pressure

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 Figure 2).

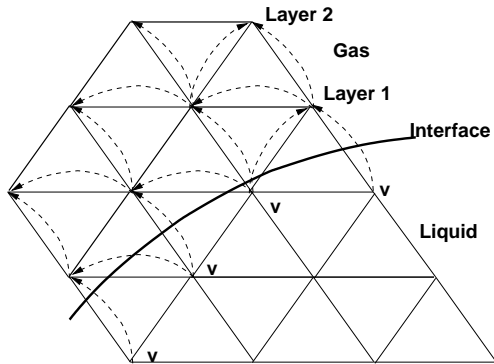


Figure 2: Extrapolation of the Velocity

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 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. 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:

$$v_n = \left| \mathbf{v} \cdot \frac{\nabla \Phi}{|\nabla \Phi|} \right|. \quad (15)$$

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.

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 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 (Löhner, 1993, Löhner, 1998), 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 (Löhner, 2001). In this way, the basic loops over edges do not require any changes. The *if*-test whether an edge group is active or inactive occurs outside the inner loops over edges, leaving them unaffected. On scalar processors, edge-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.

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 (spatially) constant. As long as the bubble is not in contact with the atmospheric air (see Figure 3), the pressure can be obtained from the isentropic relation:

$$\frac{p_b}{p_{b0}} = \left(\frac{\rho_b}{\rho_{b0}} \right)^\gamma, \quad (16)$$

where p_b, ρ_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 Eqn.(3):

$$b_t + \mathbf{v} \cdot \nabla b = 0. \quad (17)$$

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 Eqn.(16).

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.

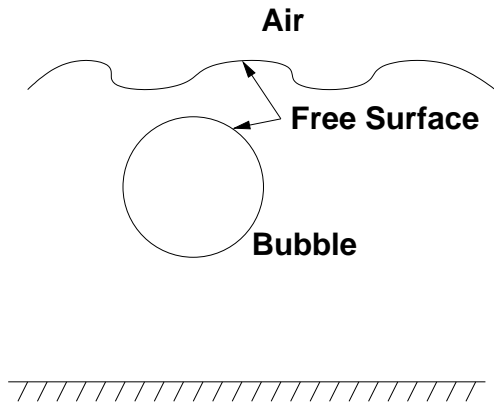


Figure 3: Bubble in Water

NUMERICAL RESULTS

In the following examples, the fluid is assumed to be a laminar Newtonian fluid with reference viscosity $\mu = 0.01$. The free slip condition is imposed on the solid boundaries. The numerical force is calculated by integrating the pressure force.

Breaking Dam Problem

This is a classic test case for free surface flows. The problem definition is shown in Figure 4.

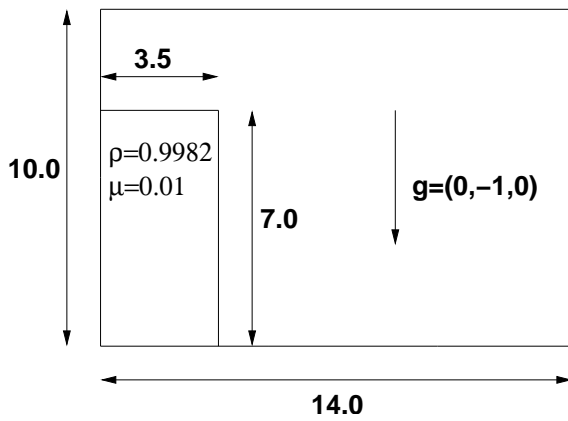


Figure 4: Breaking Dam: Problem Definition

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. Figure 5 shows the discretization for the coarse mesh. 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 (1952), as well as the numerical results obtained by Hansbo (1992), Kölke (2005), Walhorn (2002) in Figure 6. The dimensionless time and displacement are given by $\tau = t\sqrt{2g/a}$ and $\delta = x/a$. 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. Figure 7 shows the development of the flowfield and the free surface until the column of water hits the right wall. Note the mesh adaptation in Time.

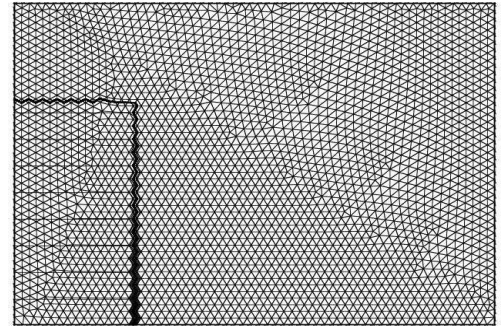


Figure 5: Breaking Dam: Discretization for the Coarse Mesh

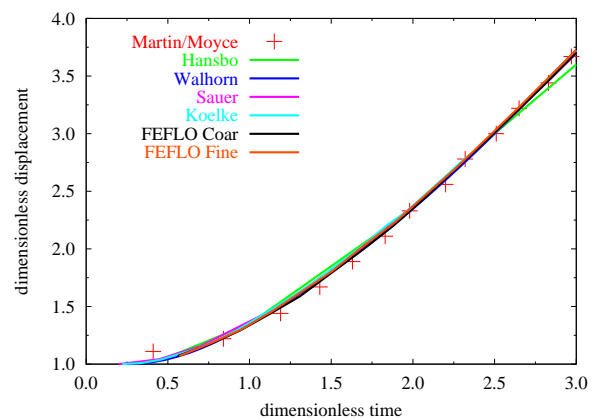


Figure 6: Breaking Dam: Horizontal Displacement

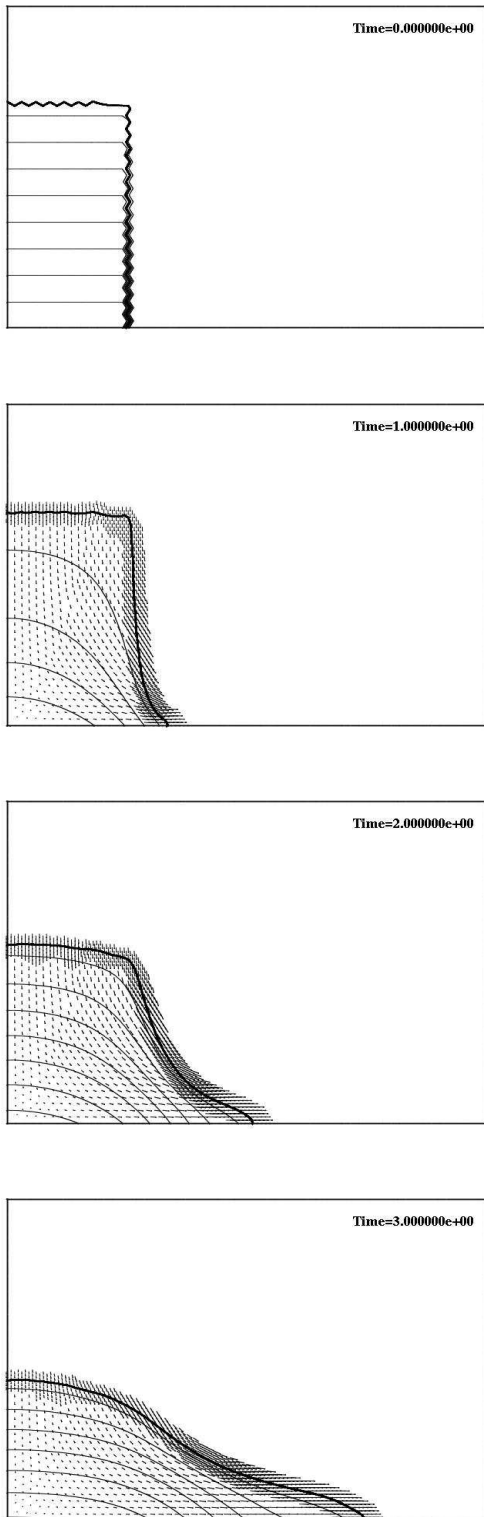


Figure 7: Breaking Dam: Flowfield at Different Times

3D Dam-Break Waves Interacting with a Circular Cylinder

In the previous section, we have already validated the accuracy of our numerical model for studying wave breaking. In this section, we shall apply our numerical model to study a three-dimensional dam-break wave interacting with a circular cylinder. The tank is 20m long, 5m wide, and 10m high. The volume of water initially contained behind a thin gate is 4m x 5m x 7m. The circular cylinder, which has a radius 1m and height 5m, is placed in the middle of the tank. The problem definition is shown in Figure 8a. The entire tank is selected as the computational domain with $nelem=1,315,224$ elements. Figure 8b shows the time history of the horizontal force acting on the cylinder. Figure 8c shows a sequence of snapshots of the free surface wave elevation.

3D Dam-Break Waves Interacting with Two Circular Cylinders

The three-dimensional dam-break wave interacting with two circular cylinders is simulated. The tank has the same dimension and water volume as in the one cylinder case. The problem definition is shown in Figure 9a. The entire tank is selected as the computational domain with $nelem=1,510,935$ elements. Figure 9b shows the time history of the horizontal force acting on the cylinders. Figure 9c shows a sequence of snapshots of the free surface elevation taken at the same time as one cylinder case. The interaction between two cylinders can be observed from Figures 9b and 9c.

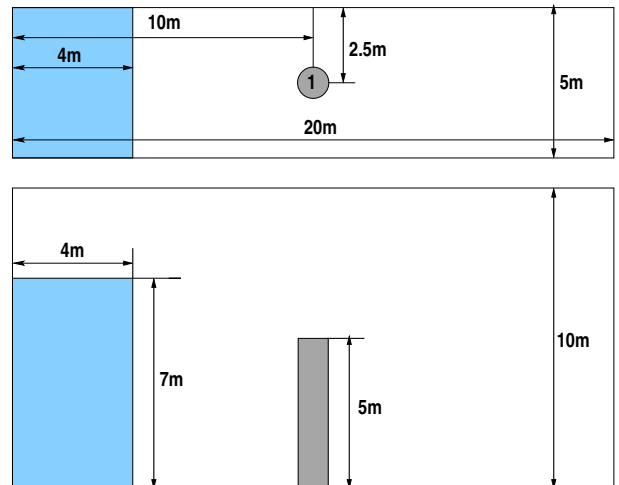


Figure 8a: 3D Dam-Break Waves Interacting with a Circular Cylinder: Problem Definition

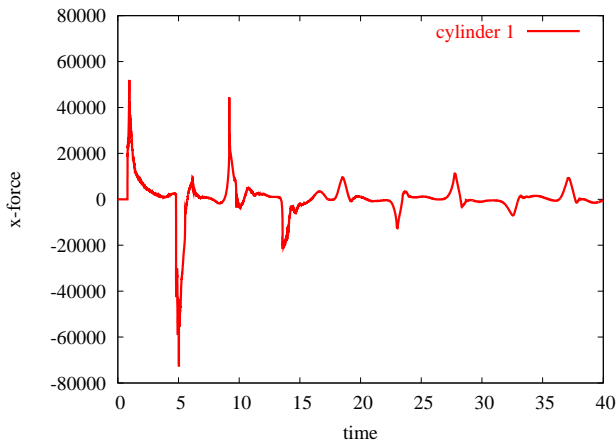


Figure 8b: Time History of Wave Impact Force on a Circular Cylinder

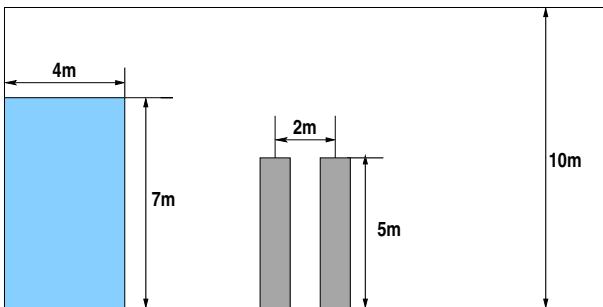
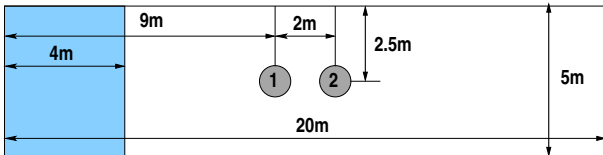


Figure 9a: 3D Dam-Break Waves Interacting with Two Circular Cylinder: Problem Definition

3D Dam-Break Waves Interacting with Four Circular Cylinders

The three-dimensional dam-break wave interacting with four circular cylinders is simulated. The tank has the same dimension and water volume as in the one cylinder case. The problem definition is shown in Figure 10a. The entire tank is selected as the computational domain with $nelem=1,900,223$ elements. Figure 10b shows the time history of the horizontal force acting on the two front cylinders, cylinder 1 and cylinder 3, defined in Figure 10a. Figure 10c shows the time history of the horizontal force acting on the two back cylinders, cylinder 2 and cylinder 4, defined in Fig-

ure 10c. Figure 10d shows a sequence of snapshots of the free surface elevation taken at the same time as one cylinder case. It can be seen from Figure 10b that the time history of the force acting on cylinder 1 and cylinder 3 is identical when the first wave hits the cylinders. However, the force differs when the wave bounces back from the wall of the tank. The same unsymmetric behavior can be observed from Figure 10c as well. The unsymmetric behavior may result from the unsymmetric unstructured mesh, which will be further studied in the future.

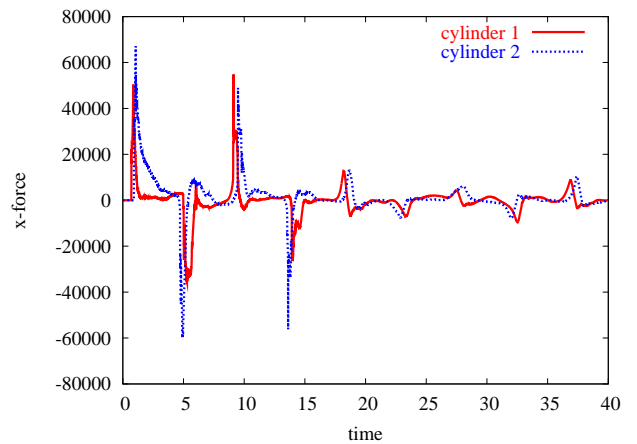


Figure 9b: Time History of Wave Impact Force on Two Circular Cylinders

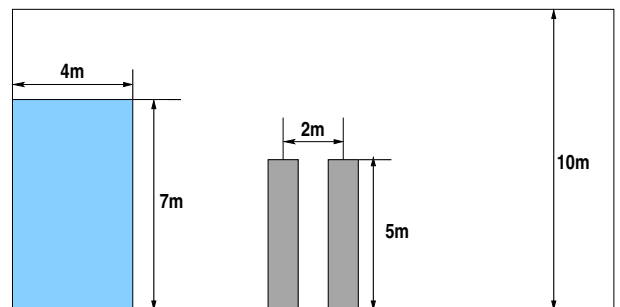
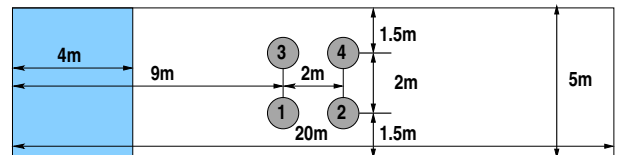


Figure 10a: 3D Dam-Break Waves Interacting with Four Circular Cylinder: Problem Definition

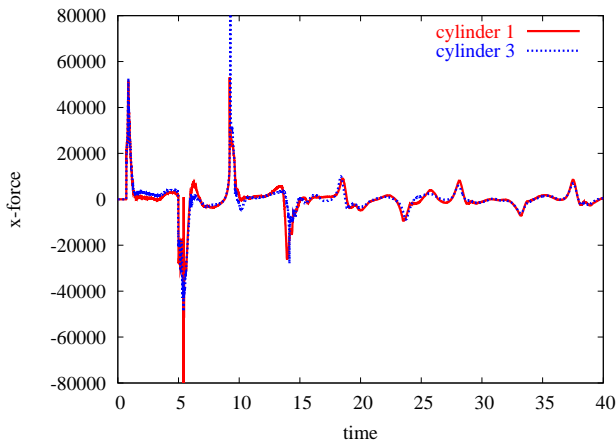


Figure 10b: Time History of Wave Impact Force on Two Front Cylinders for the Four Circular-Cylinder Case

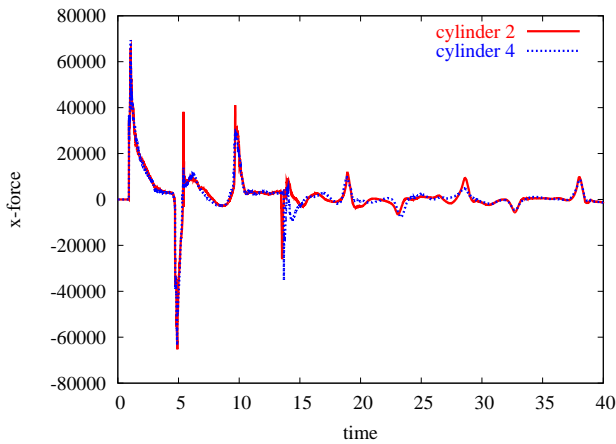


Figure 10c: Time History of Wave Impact Force on Two Back Cylinders for the Four Circular-Cylinder Case

CONCLUSIONS

A robust 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. In the present approach, only the liquid phase needs to be simulated. An effective extrapolation technique has been developed to obtain the pressure and velocity in the gas region.

In this study, the classic breaking dam problem has

been used to validate the computer code. The results obtained for the horizontal location of the free surface along the bottom wall are compared to the experimental values and other numerical predictions. The agreement is very good, even for the coarse mesh. Numerical simulations of a three dimensional dam-break wave interacting with a single cylinder, two cylinders and four cylinders are performed. Computational results have demonstrated that the present CFD method is capable of simulating the interactions of extreme waves and three-dimensional structures, which are of great importance for the comprehension of many natural phenomena in marine, coastal and offshore engineering.

REFERENCES

- Alessandrini, B. and Delhommeau, G., 1996, "A Multi-grid Velocity- Pressure-Free Surface Elevation Fully Coupled Solver for Calculation of Turbulent Incompressible Flow Around a Hull," *Proc. 21st Symp. on Naval Hydrodynamics*, Trondheim, Norway, June.
- Bell, J.B., Colella, P. and Glaz, H., 1989, "A Second-Order Projection Method for the Navier-Stokes Equations," *J. Comp. Phys.* 85, 257-283.
- Bell, J.B. and Marcus, D.L., 1992, "A Second-Order Projection Method for Variable-Density Flows," *J. Comp. Phys.* 101, 2.
- Biausser, B., Fraunie, P., Grilli, S. and Marcer R., 2004, "Numerical analysis of the internal kinematics and dynamics of three-dimensional breaking waves on slopes," *International Journal of Offshore and Polar Engineering*, Vol. 14, No. 4.
- Chen, G., Kharif, C., 1999, "Two-Dimensional Navier-Stokes Simulation of Breaking Waves," *Physics of Fluids*, 11(1), 121-133.
- Codina, R., 2001, "Pressure Stability in Fractional Step Finite Element Methods for Incompressible Flows," *J. Comp. Phys.* 170, 112-140.
- Fekken, G., Veldman, A.E.P. and Buchner, B., 1999, "Simulation of Green Water Loading Using the Navier-Stokes Equations," *Proceedings of the 7th International Conference on Numerical Ship Hydrodynamics*, Nantes, France.
- Hansbo, P., 1992, "The Characteristic Streamline Diffusion Method for the Time-Dependent Incompressible Navier-Stokes Equations," *Comp. Meth. Appl. Mech. Eng.* 99, 171-186.
- Hirt, C.W. and Nichols, B.D., 1981, "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries," *Journal of Computational Physics* 39, 201.
- Huijsmans, R.H.M. and van Groenou, E., 2004, "Coupling Freak Wave Effects with Green Water Simulations," *Proceeding of the 14th ISOPE*, Toulon, France, May 23-28.

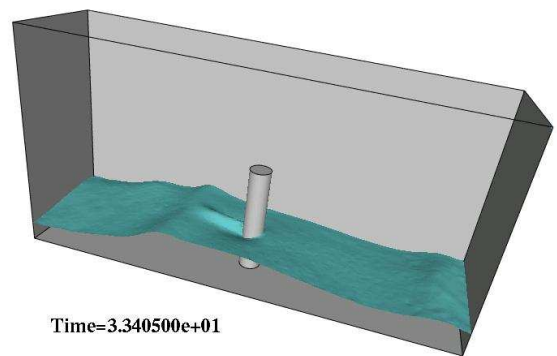
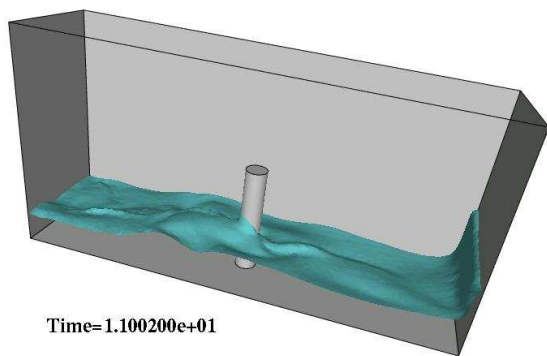
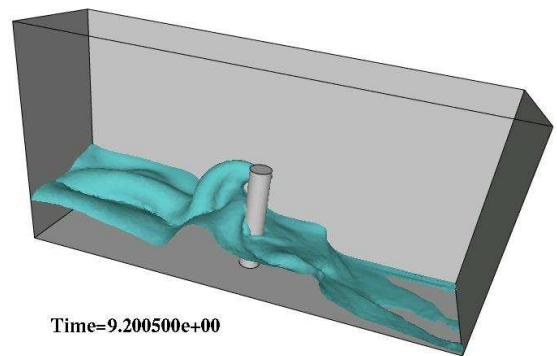
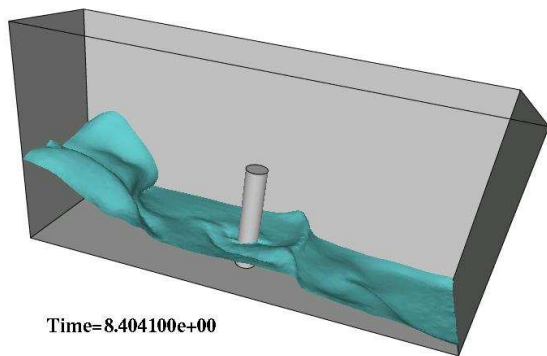
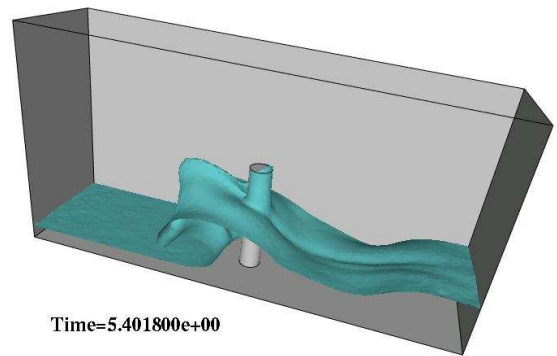
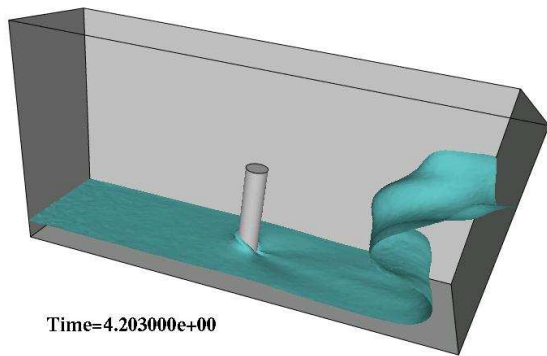
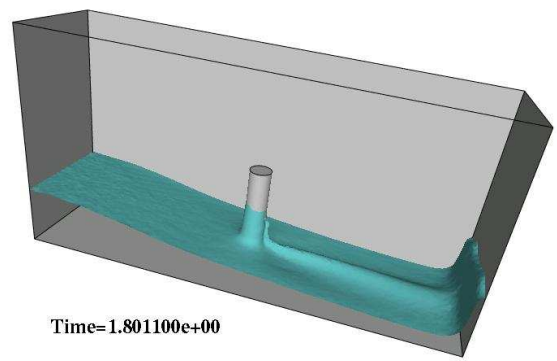
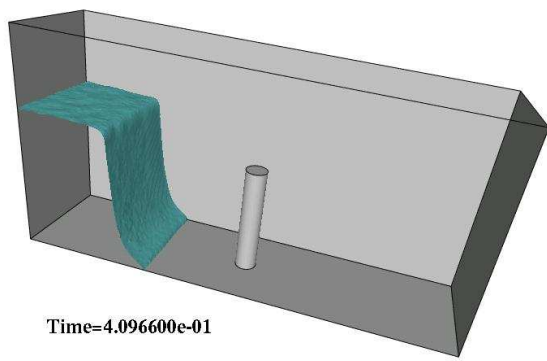


Figure 8c: Snap Shots of the Free Surface Wave Elevation for One-Cylinder Case

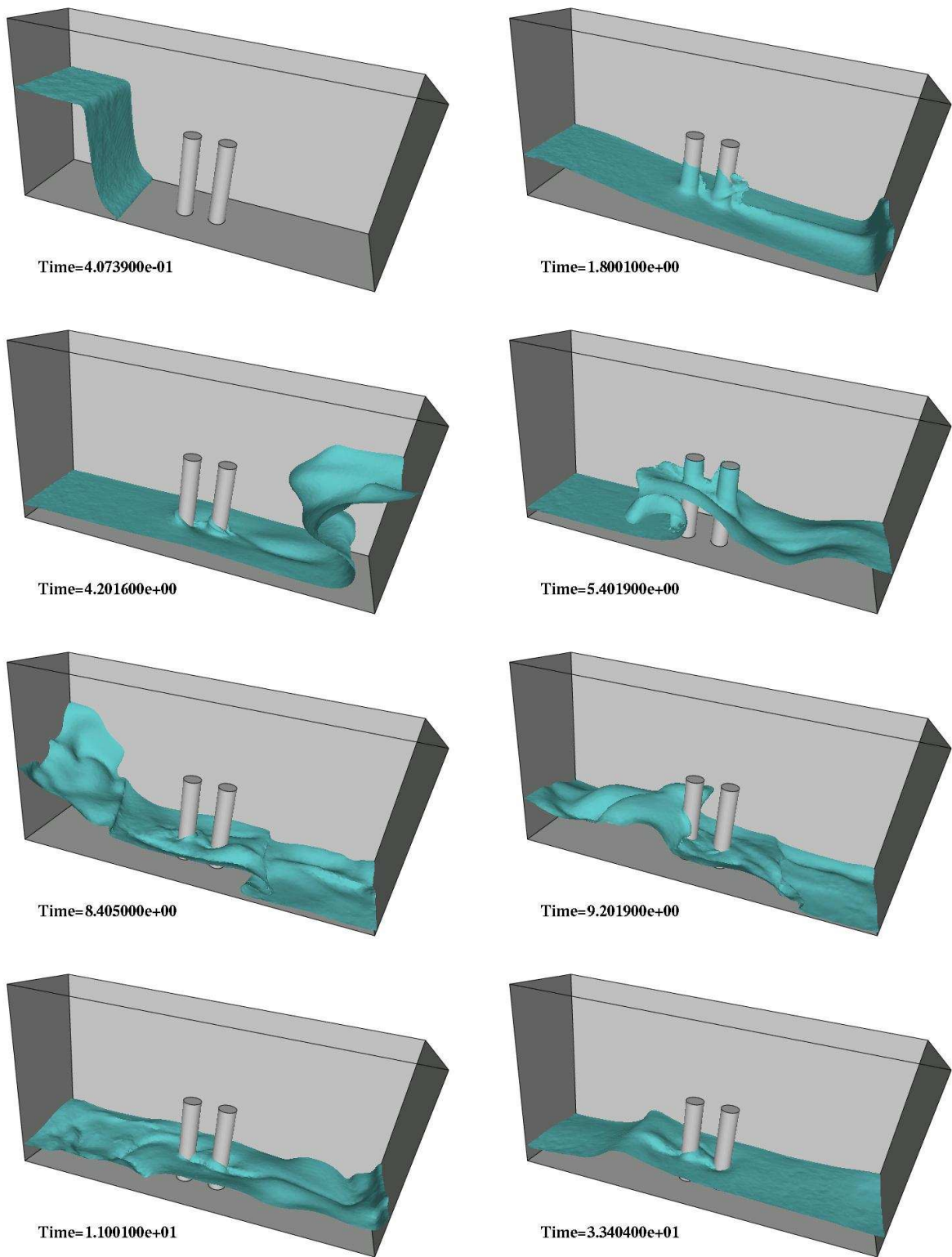


Figure 9c: Snap Shots of the Free Surface Wave Elevation for Two-Cylinder Case

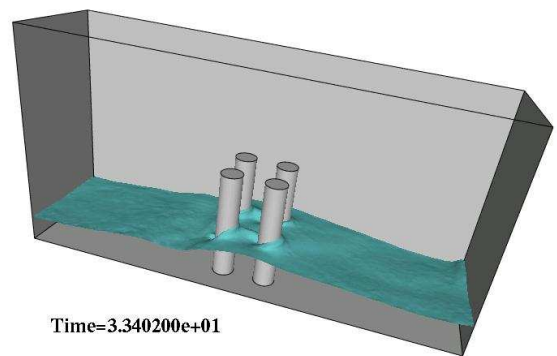
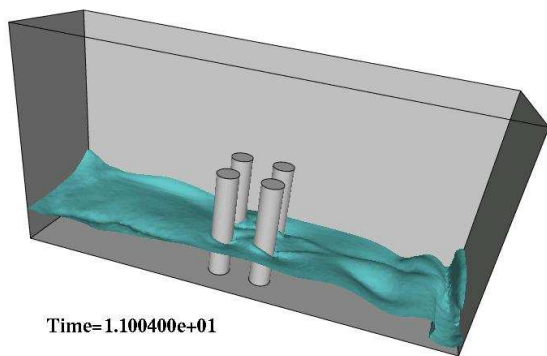
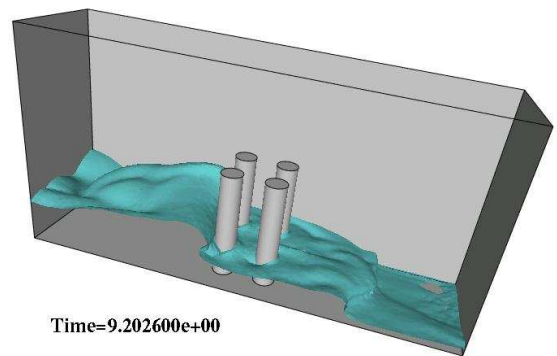
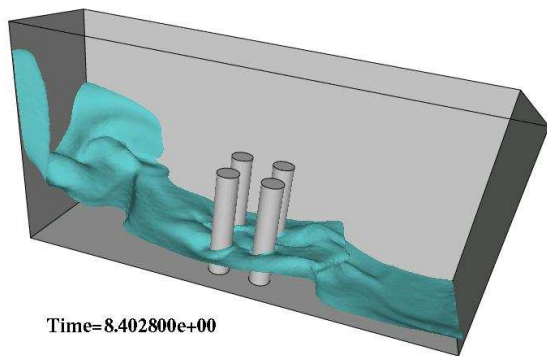
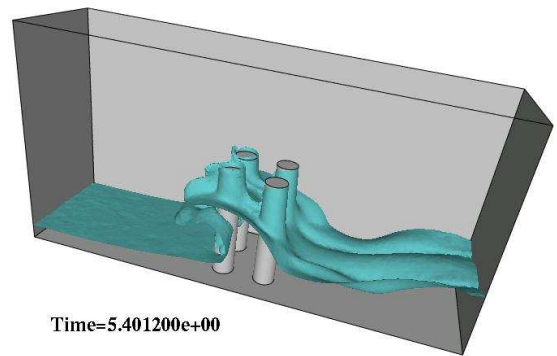
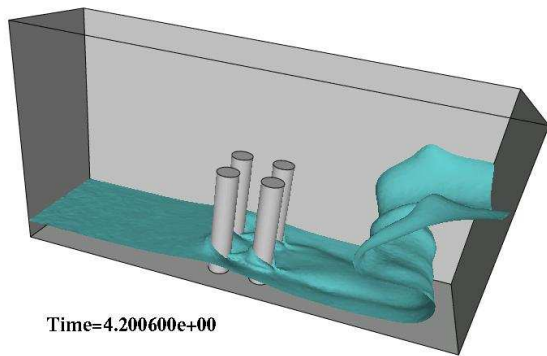
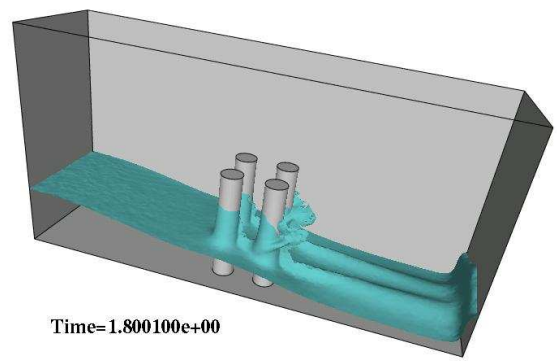
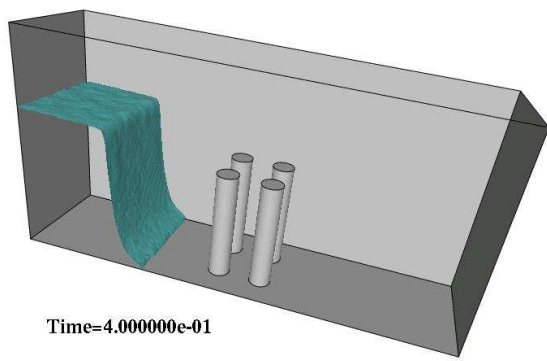


Figure 10d: Snap Shots of the Free Surface Wave Elevation for Four-Cylinder Case

Kallinderis, Y. and Chen, A., 1996, "An Incompressible 3-D Navier-Stokes Method with Adaptive Hybrid Grids," *AIAA-96-0293*.

Karbon, K.J. and Kumarasamy, S., 2001, "Computational Aeroacoustics in Automotive Design, Computational Fluid and Solid Mechanics," *Proc. First MIT Conference on Computational Fluid and Solid Mechanics*, 871-875, Boston.

Karbon, K.J. and Singh, R., 2002, "Simulation and Design of Automobile Sunroof Buffeting Noise Control," *8th AIAA-CEAS Aero-Acoustics Conf.*, Brenckridge.

Kim, J. and Moin, P., 1985, "Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations," *J. Comp. Phys.* 59, 308-323.

Kölke, A., 2005, "Modellierung und Diskretisierung bewegter Diskontinuitäten in Randgekoppelten Mehrfeldaufgaben," *Ph.D. Thesis*, TU Braunschweig.

Li, Y., Kamioka, T., Nouzawa, T., Nakamura, T., Okada, Y. and Ichikawa, N., 2002, "Verification of Aerodynamic Noise Simulation by Modifying Automobile Front-Pillar Shape," *JSAE 20025351, JSAE Annual Conf.*, Tokyo, Japan.

Löhner, R., 1990, "A Fast Finite Element Solver for Incompressible Flows," *AIAA-90-0398*.

Löhner, R. 1993, "A Fast Finite Element Solver for Incompressible Flows," *AIAA-90-0398*.

Löhner, R., 1998, "Renumbering Strategies for Unstructured-Grid Solvers Operating on Shared-Memory, Cache-Based Parallel Machines," *Comp. Meth. Appl. Mech. Eng.* 163, 95-109.

Löhner, R., Yang, C., Oñate, E. and Idelsohn, S., 1999, "An Unstructured Grid-Based, Parallel Free Surface Solver," *Appl. Num. Math.* 31, 271-293.

Löhner, R. 2001, *Applied CFD Techniques*; J. Wiley & Sons.

Löhner, R., 2004, "Multistage Explicit Advective Prediction for Projection-Type Incompressible Flow Solvers," *J. Comp. Phys.* 195, 143-152.

Martin, D. and Löhner, R., 1992, "An Implicit Linelet-Based Solver for Incompressible Flows," *AIAA-92-0668*.

Martin, J.C. and Moyce, W.J., 1952, "An Experimental Study of the Collapse of a Liquid Column on a Rigid Horizontal Plane," *Phil. Trans. Royal Soc. London A244*, 312-324.

Nichols, B.D. and Hirt, C.W., 1975, "Methods for Calculating Multi-Dimensional, Transient Free Surface Flows Past Bodies," *Proc. First Intern. Conf. Num. Ship Hydrodynamics*, Gaithersburg, MD, Oct. 20-23.

Ramamurti, R. and Löhner, R., 1996, "A Parallel Implicit Incompressible Flow Solver Using Unstructured Meshes," *Computers and Fluids* 5, 119-132.

Rhee, S.H., and Stern, F., 2002, "RANS Modeling of Spilling Breaking Waves," *ASME J. Fluids Eng.*, Vol. 124, No. 2, pp. 424-432.

Scardovelli, R.; Zaleski, S., 1999, "Direct numerical simulation of free-surface and interfacial flow," *Annual Review of Fluid Mechanics* 31: 567-603.

Takamura, A., Zhu, M. and Vinteler, D., 2001, "Numerical Simulation of Pass-by Maneuver Using ALE Technique," *JSAE Annual Conf.*, Tokyo, Japan.

Walhorn, E., 2002, "Ein Simultanes Berechnungsverfahren für Fluid-Struktur-Wechselwirkungen mit Finiten Raum-Zeit-Elementen," *Ph.D. Thesis*, TU Braunschweig.

Yang, C. and Löhner, R., 1998 "Fully Nonlinear Ship Wave Calculation Using Unstructured Grids and Parallel Computing," *Proc. 3rd Osaka Colloquium on Advanced CFD Applications to Ship Flow and Hull Form Design*, Osaka, Japan.