Elsevier Editorial System(tm) for Graphical Models

Manuscript Draft

Manuscript Number: GMOD-07-3

Title: Interaction of Multiphase Flow with Animated Models

Article Type: Regular Article

Section/Category:

Keywords: animation; level sets; volume of fluid; multiphase flow

Corresponding Author: Dr. Viorel Mihalef,

Corresponding Author's Institution: Rutgers University

First Author: Viorel Mihalef

Order of Authors: Viorel Mihalef; Samet Kadioglu, PhD; Mark Sussman, PhD; Dimitris Metaxas, PhD; Vassilios Hurmusiadis, PhD

Manuscript Region of Origin:

Abstract: In the present work we propose a novel fluid simulation method that enables us to perform interactions of animated
bodies with a coupled gas-liquid flow. This way we can animate flows in which not only the liquid but also the air
is a performer. Our method allows the use of rigidly moving, articulated or deforming meshes. The paper shows
how to do this practically, using a coupled level set and volume of fluid method. Animations of a swimmer in a
pool, a hand scooping out water and a heart beating and spurting out either liquid or gas, showcase the strengths

of our method.

Cover Letter

Paper Title: Interaction of multiphase flow with animated models

Corresponding Author: Viorel Mihalef

Work Address
Department of Computer and Information Science
Rutgers, the State University of New Jersey
110 Frelinghuysen Rd.
Piscataway, NJ 08854
USA
Tel: (732) 445-2795 (Office)
Fax: (732) 445-2914
Email: mihalef@cs.rutgers.edu

Home Address
6 Civic Center Dr Apt 18
East Brunswick, NJ 08816

Interaction of Multiphase Flow with Animated Models

Viorel Mihalef [a,*], Samet Kadioglu [b], Mark Sussman [c], Dimitris Metaxas [a], Vassilios Hurmusiadis [d]

[a] *Department of Computer and Information Science, Rutgers University, 110 Frelinghuysen Rd., Piscataway, NJ 08854, USA*

[b] *Department of Mathematics, University of North Carolina, CB 3250, Phillips Hall, Chapel Hill, NC 27599, USA*

[c] *Department of Mathematics, Florida State University, Tallahassee, FL 32306, USA*

[d] *Primal Pictures, Tennyson House, 159-163 Great Portland Street London, W1W 5PA, UK*

Abstract

In the present work we propose a robust fluid simulation method that enables us to perform interactions of animated bodies with a coupled gas-liquid flow. This way we can animate flows in which not only the liquid but also the air is a performer. Our method allows the use of rigidly moving, articulated or deforming meshes. The paper shows how to do this practically, using a coupled level set and volume of fluid method. Animations of a swimmer in a pool, a hand scooping out water and a heart beating and spurting out either liquid or gas, showcase the strengths of our method.

Key words:

* corresponding author

Email addresses: mihalef@cs.rutgers.edu (Viorel Mihalef), kadioglu@email.unc.edu (Samet Kadioglu), sussman@math.fsu.edu (Mark Sussman), dnm@cs.rutgers.edu (Dimitris Metaxas), vassilios@primalpictures.com (Vassilios Hurmusiadis).

URL: http://paul.rutgers.edu/~mihalef (Viorel Mihalef).

1 Introduction

As the animation and special effects industries grow in sophistication (as a consequence of the constant need for more realism or versatility), the subfield of fluid animation experiences itself an expansion of its boundaries and capabilities. The present paper addresses a specific problem that derives from the aforementioned expansion of capabilities. Namely, it is common now in fluid animation to have an animated character interacting with a body of *liquid* (see for example Foster & Fedkiw (7)), but more research is needed in order to obtain good interaction with a coupled *gas-liquid* flow. In the present work we propose a novel physics-based method that enables us to provide an automatic solution for such interactions, in which *both* the liquid and the air act as performers. Our paper outlines the practical aspects of this approach, in a coupled level set and volume of fluid setting. The contribution of our work consists in offering a robust and unified automatic solution to the problem in question, thus avoiding non-physical approximations in which air-bubbles are modeled separately from the liquid flow.

Our approach is geared towards the needs of animators working in the movie industry. A common scenario in a production setting is for the animator to receive a sequence of moving meshes, and be asked to have them interact with a body of water or other liquid, or with a multiphase flow (e.g. water+air). The methods of choice in the last decade for simulating fluids have been Eulerian, and we follow the trend, with a two-phase flow simulator that uses a coupled level set and volume of fluid method for surface advection. Our system consists thus of two components, the aforementioned fluid simulator and a component that handles the interaction with the moving meshes, interpolating (at each time step) the animated mesh surface and velocity from the Lagrangian grid to the Eulerian grid.

Regarding the fluid simulator, we use the two-phase flow version of the Coupled Level Set and Volume Of Fluid (CLSVOF) method of Sussman et al. (35). This method has been used in a simpler setting (only one phase/fluid) by Mihalef et al. (24) to generate and control breaking ocean waves, and in its present form by Mihalef et al. (25) to generate boiling flows. Our method combines a level set ingredient for generating smooth surfaces and a volume of fluid ingredient that improves the mass conservation properties of the level set solver. While other methods like Hong & Kim (16) or Mihalef et al. (25) generated bubbly water flow interacting with fixed objects, we go one step further in functionality, showing interactions with prescribed complex object motion. We also point out that our method is physics-based, whereas the methodology used by Hong & Kim (16) features a non-physical approximation of the variable-density projection equation for pressure (equation 5 in their paper). In Jimenez et al. (21) our numerical model is compared favorably with exper-

imental photographs of *real* bubbles. There has been work on interaction with moving meshes, such as Carlson et al. (3), Guendelman et al. (14), Chentanez et al. (2), where only one-phase water flow is used, but two-way interactions are shown as well. We note also that Kim et al. (19) and Carlson et al. (3) report fluid interaction with solids that undergo only rigid body motion. In fact, Kim et al. (19) present a two-phase flow system in the same vein with ours, but developed in a different direction. Their system handles two-way interactions, but they do not focus, like us, on simulations in which the solid performs very complex prescribed motion, as is usually the case in animation. Such motion can easily produce artifacts for non-robust systems, and proposing a robust system is one of our contributions.

Another feature of our paper is the introduction to graphics of a boundary condition model suitable for open-field simulations, like the ones encountered for example by a character swimming in the ocean. A common practice for such a case is to choose the domain large enough so that the walls (enforcing Dirichlet boundary conditions) do not make their influence sensed, but such an approach is obviously inefficient. A choice of Neumann boundary conditions would inevitably drain the domain. We counter that by enforcing special hydrostatic pressure conditions along the vertical boundaries. This approach has been used in the computational fluid dynamics community before, see for example Dommermuth et al. (4). More details regarding this approach can be found in section 3.2 and the appendix.

In summary, our paper proposes a physics-based system that handles robustly interactions of two-phase flow with complex animated meshes, and also introduces to graphics a useful method for modeling open-field boundary conditions. As we show in our examples, this type of one-way interaction between animated meshes and coupled air-water flow that we tackle in our paper leads to very complex, novel and useful animations.

The following section of the paper reviews previous work related to ours, concerning Navier-Stokes solvers and interactions between solid objects and liquids. Our simulation method is presented in section 3, followed by several illustrative animations in section 4. The final part contains concluding remarks, as well as an overview of the directions that our future research will follow.

2 Previous work

In what follows we present a quick overview of methods for solving the three dimensional Navier-Stokes equations used within the graphics community, with emphasis on those works featuring interactions between fluids and solid objects.

The advantage of using the full 3D Navier-Stokes in animation was first demonstrated by Foster & Metaxas (8). Further improvements were presented by Stam (36), who introduced the stable semi-Lagrangian techniques for computing the advection part of the Navier-Stokes equations, and Foster & Fedkiw (7) respectively Enright et al. (5), who improved the representation of the liquid surface with hybrid models, combining implicit surfaces and particles placed inside the liquid, respectively level sets and particles placed in a narrow band about the interface. Level set based solvers are widely used lately due to their capabilities for capturing and rendering the interface as a smooth implicit surface, and also for the theoretical and implementation ease with which they can deal with topology changes. Hong & Kim (16) is such a work in which the authors use the particle level set of Enright et al. (5) (PLS) as a surface capturing device in a two-phase flow setting. One of the main strengths of the PLS method is that it improves drastically the mass conservation properties of the level set method. Other methods that achieve the same goal are the CLSVOF method used by Mihalef et al. (24) for single-phase flow respectively by Mihalef et al. (25) for two-phase flow, also Yang et al. (39) with another CLSVOF implementation, and the CIP method used by Takahashi et al. (38) for one phase flow and Song et al. (32) for two-phase flow. The CLSVOF method was devised in Sussman & Puckett (33) and it addressed formally the loss of mass of the level set method by coupling it with a volume of fluid method. Volume of fluid methods are probably the most widely used Eulerian methods in industrial computational fluid dynamics, due to their unconditional mass preservation properties. In graphics Hong & Kim (15) used such a method for generating bubbles in liquid. Their subsequent switch to PLS in Hong & Kim (16) reflects also a recent trend in academic research, of embracing more and more easy-to-implement level sets methods for capturing interfaces. Two of the problems encountered by the volume of fluid methods, namely the generation of spurious volume fractions ("flotsam and jetsam") and the recovery of a smooth surface from the volume fraction field are addressed and solved by the CLSVOF method.

Another class of Navier-Stokes solvers for free surfaces consists of the Lagrangian methods, which are mainly implementations and extensions of the smooth particle hydrodynamics method (SPH) proposed by Monaghan (26). Several papers implemented this recently with good results, for example Muller et al. (27) and (28), or Keiser et al. (18). Kondoh et al. (20) coupled an SPH model for water to thin deformable cloth pointing out that particle based fluid methods can be coupled without leaking using robust point face collisions. As underlined by Guendelman et al. (14) though, their method will leak if the time step is not chosen sufficiently (sometimes severely) small. Nevertheless this could be alleviated with a more robust point face collision method as in Bridson et al. (1).

A newer type of Navier-Stokes fluid solvers used in graphics consists of the hy-

brid methods implementing the arbitrary Lagrangian-Eulerian methods (ALE) from the CFD literature. Feldman et al. (9) and Feldman et al. (10) show nice results of one-phase flow (smoke) interaction with static respectively moving meshes. Their system is updated in Chentanez et al. (2) to include also interaction of gas or liquid with deformable bodies.

Some of the works already cited include interactions of fluid flow and solid (rigid or deformable) structures. For completeness we need to add Peskin (30) and Griffith et al. (13), which are good sources for the immersed boundary method, also Genevaux et al. (11), Takahashi et al. (37), Glowinski et al. (12), Carlson et al. (3), and Guendelman et al. (14). While Carlson et al. (3) do use a multi-phase flow setup in their paper, it is used for modeling the interface between water and rigid bodies, rather than between water and air. Guendelman et al. (14) provide an algorithm that works well for one phase fluids such as smoke and for fluids with interfaces such as water, but they do not use a coupled multiphase flow that combines air and water. More recently, Losasso et al. (22) simulate multiple interacting fluids and report adding a plethora of extra capabilities to their system, like air-coupling, temperature fields and surface reactions, but do not focus on interaction with complex animated meshes, like we do here.

In the following we present our multi-phase flow framework which accommodates simultaneous interactions of water and air with animated meshes.

3 Simulation Method

Within our Eulerian fluids code, all the elements of the scene (liquid or solid) are described with the use of two level sets (fig.1), one for the air/liquid phase, another for the fluid/solid phase (the latter level set has a similar functionality with the one introduced by Carlson et al. (3)). The liquid is simulated using a two-phase flow formulation as in Sussman et al. (35) (see also Sussman et al. (34) for the original framework of a Navier-Stokes multiphase solver using level sets). This means air is also modeled, and the zero level set coincides with the air/water interface. This interface may suffer from excessive smoothing in the classical level set multi-phase flow method, but we maintain it sharp in this work. Thus, the standard effects appearing at the liquid/gas interface (droplet and bubble formation) are captured quite well (this claim is supported by evidence presented in the animations section and in the accompanying videos). The non-rigid bodies in the scene are pre-animated, and given in a Lagrangian format as nodes and planar faces. One component of our code performs the conversion of the Lagrangian information into Eulerian.

At this point it is probably clear that the high-level description of our al-

gorithm is that each computational step has two parts: we first perform the Lagrangian-to-Eulerian mesh and velocity conversion, then we use the solid level set to help the fluid solver advance the flow calculation. In the remainder of this section we describe the first part (section 3.1), followed by an interlude about the CLSVOF method and our Navier-Stokes formulation (3.2), and by a detailed description of the second part (3.3).

3.1 Converting Lagrangian node and face data to level set data

We perform this conversion in two ways, in order to accommodate both one-sided volumes and double sided ones (thin shells). In this respect we resemble in spirit the thickened triangle volume method proposed by Kondoh et al. (20) for treating collisions between fluids and thick or thin solids (cloth). In our algorithm we call the one sided volumes singly-wetted" and the double sided ones doubly-wetted".

Doubly-Wetted Interface We traverse through all Lagrangian faces and update Eulerian cells in a small narrow band in the neighborhood of each face with the positive minimum distance to the face. Uninitialized cells are given the value $+\alpha\Delta x$, where α is the width of a small narrow band about the zero level set (for us alpha is 3). After all the faces have been traversed, we let $\phi = \phi - \Delta x$.

Singly-Wetted Interface a. For converting Lagrangian solid data to the Eulerian grid, our algorithm relies on the ability of the Eulerian grid to be coarsened. In our case, we use an adaptive hierarchy of rectangular grids which enables one to naturally coarsen to the coarsest adaptive level. From the coarsest level, one can further coarsen until the grid dimensions on the coarsest grid are no longer divisible by 2. On the coarsest level, we traverse all grid cells, and for each grid cell, traverse all Lagrangian faces.

b. On the next finer level, we interpolate (piecewise constant interpolation) the level set function from the previous level. We traverse through all Lagrangian faces, and update Eulerian cells in a small narrow band about each face with the signed minimum distance to the face. Uninitialized cells are given the value $\pm\alpha\Delta x$ where the sign of the level set function is determined by the previously interpolated value.

c. repeat step (b) for each of the finer levels.

The velocity is converted from node/face data to Eulerian data as follows: for a given computational cell, the closest lagrangian face is found during the calculation of the level set (distance) function. The velocity then becomes the weighted average of the nodal velocities from the face. The weights are determined by the distance from the cell center to the node by using a finite-support kernel function.

3.2 Fluid Solver

Our fluid solver consists of two components: (1) interface capturing and (2) Navier-Stokes solver. The air/water surface is "captured" using the Coupled Level Set and Volume-Of-Fluid method (CLSVOF, (33)). The level set function is used to follow the interface as it is coupled with the Navier-Stokes solver. In other words, interface cells in which the level set function changes sign are used to create a signed distance function and also denote where the jump in density and viscosity occurs. Mass conservation is explicitly built with the help of the volume-of-fluid method. The coupling between the level set function and the volume-of-fluid function occurs primarily during the level set reinitialization step, in which the level set function is set to be the exact distance to the volume-of-fluid reconstructed interface (in cells in which the level set function changes sign), and the volume-of-fluid reconstructed interface gets the normal from the level set function (fig. 2). The resulting "coupled" level set function preserves mass to a fraction of a percent in all the standard tests. We use the level set to reconstruct the surface, hence we do not have to worry about the problems associated with volume-of-fluid surface reconstruction.

The two-phase Navier-Stokes solver that we implement in this work is a sharp interface method for two-phase flow, which we report in Sussman et al. (35). We have used a similar solver with nice results in our paper on boiling simulation (25). This solver, as used in this paper, features a rewriting of the Navier-Stokes equation such that it accommodates open-boundary conditions, and we elaborate on it in the Appendix. This new formulation, although not a must for any of our animations (or the ones in the boiling simulation paper), brings extra flexibility to the standard liquid-in-a-box simulation setting. It is worthwhile noting how CLSVOF compares to the usual surface tracker used by the graphics community, the particle level set. While PLS favors the formation of thin sheets of liquid, CLSVOF displays more numerical surface tension, in that, if a sheet of liquid gets too thin and is close to disappearing, the method favors the formation of droplets of liquid (or bubbles of air), in order to preserve mass. In the limit, as the domain is refined, both methods converge to the same solution. Thus, it appears that in simulations in which one prefers the formation of many bubbles or droplets, CLSVOF is a strong alternative to PLS.

All the simulations we present in this work assume outflow conditions on the vertical boundaries. Also, in order to ensure that the liquid does not drain from the domain we had to reformulate the Navier-Stokes equation in the form given below. In this way we practically enforce that the pressure at the vertical domain boundaries equals the initial hydrostatic pressure. The appendix contains details on how one reaches this alternative formulation.

The governing equations consist of the level set equation, volume-of-fluid equation, inviscid Navier-Stokes equation, and continuity condition:

$$\frac{D\phi}{Dt} = 0 \quad \frac{DF}{Dt} = 0$$

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p^{live} - (\rho^L - \rho^G) g (z - z_{surf}) \nabla H$$

$$\nabla \cdot \mathbf{u} = 0$$

ϕ is the level set function which is positive in the liquid and negative in the gas. F is the volume-of-fluid function which satisfies $F = 0$ in computational cells containing only gas, $F = 1$ in computational cells containing only liquid and $0 < F < 1$ otherwise. The density ρ and Heaviside function H satisfy

$$\rho = \begin{cases} \rho^L & \phi \geq 0 \\ \rho^G & \phi < 0 \end{cases} \quad H = \begin{cases} 1 & \phi \geq 0 \\ 0 & \phi < 0 \end{cases}$$

p^{live} is the "live" pressure, which is the total pressure offset by the hydrostatic pressure (z_{surf} is the initial surface water level),

$$p^{live} = p - \rho g (z - z_{surf})$$

From now on, when we write or talk about the pressure p we understand p^{live}. $\frac{D}{Dt}$ is the material (advective) derivative,

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$$

Please note that, by integrating the volume-of-fluid equation over the simulation domain and using the divergence theorem, one obtains an easy mathematical proof that the volume of liquid is preserved in time. This gives a quick mathematical intuition of how the CLSVOF method (or at least its volume-of-fluid portion) works.

3.3 Outline of numerical method for gas-liquid-solid coupling

At the beginning of each time step, we are given the liquid (gas, solid) velocity $\mathbf{u}^L(\mathbf{u}^G,\mathbf{u}^S)$, the gas/liquid level set function ϕ, the solid/fluid level set function ψ obtained from the solid mesh, and the gas/liquid volume-of-fluid function F. The face centered density is defined in terms of the "height fraction" θ,

$$\rho = \rho^S\theta_{solid} + (1-\theta_{solid})(\rho^L\theta + \rho^G(1-\theta))$$

The height fraction θ for the liquid level set gives the one dimensional fraction of water between adjacent cells (the solid height fraction is analogously defined). For example, in 2D the height fraction would be

$$\theta_{i+1/2,j} = \frac{\phi^+_{i+1,j} + \phi^+_{i,j}}{|\phi_{i+1,j}| + |\phi_{i,j}|}$$

where $\phi^+ = max(\phi, 0)$. For each time step, one does the following:

(1) calculate provisional advective states,

$$\frac{D\mathbf{u}^L}{Dt} = 0 \quad \frac{D\mathbf{u}^G}{Dt} = 0 \quad \frac{D\phi}{Dt} = 0 \quad \frac{DF}{Dt} = 0$$

(2) update of gravity terms,

$$\mathbf{v^P} = \mathbf{u^{advect,P}} - \frac{\Delta t(\rho^{\mathbf{L}} - \rho^{\mathbf{G}})g(z - z_{surf})\nabla H}{\rho}$$

where $\mathbf{P}$ is taken first to be $\mathbf{L}$, then $\mathbf{G}$.

(3) pressure correction step, in which we also take in account the solid velocity.

$$\mathbf{v}_{i+1/2,j} = \begin{cases} \mathbf{v^{solid}}_{i+1/2,j} & \psi_{i+1,j} \le 0 \text{ or } \psi_{i,j} \le 0 \\ \mathbf{v^L}_{i+1/2,j} & \phi_{i+1,j} \ge 0 \text{ or } \phi_{i,j} \ge 0 \\ \mathbf{v^G}_{i+1/2,j} & \text{otherwise} \end{cases}$$

$$\nabla \cdot \frac{\nabla p}{\rho} = \nabla \cdot \mathbf{v}$$

(4) update the density and height fractions using the current values of the level sets, then update the global velocity

$$\mathbf{u^{n+1}}_{i+1/2,j} = \begin{cases} \mathbf{v^{solid}}_{i+1/2,j} & \psi_{i+1,j} \leq 0 \text{ or } \psi_{i,j} \leq 0 \\ \mathbf{v^{L}}_{i+1/2,j} - \frac{\nabla p}{\rho} & \phi_{i+1,j} \geq 0 \text{ or } \phi_{i,j} \geq 0 \\ \mathbf{v^{G}}_{i+1/2,j} - \frac{\nabla p}{\rho} & \text{otherwise} \end{cases}$$

(5) set the liquid and gas velocity to be the global velocity (they will pick up their correct component by doing this) and extrapolate the liquid velocity as in Sussman et al. (35). This is done in a mass-conservative manner, with results comparable with the approach used by Rasmussen et al. (31), thus obtaining realistic dynamics upon liquid surface merger. In the bubble formation section of Sussman et al. (35) we show that we can also calculate accurately bubble pinch-off.

4 Animations

In order to demonstrate the power and wide applicability of our framework, we generated several animations in which we have interactions of animated models with a coupled body of air/water. The presented animations are of a hand scooping water, a heart pump spurting liquid and blowing air bubbles, and a swimmer interacting with water. All the simulations were run on an Athlon 64 workstation with 2.4GHz and 3GB RAM and were rendered in Vue d'Esprit (Vue).

The hand-water interaction animation uses a predefined sequence of a moving hand (obtained with an articulated hand model). The experiment was run on a $192 \times 128 \times 128$ grid, in about 15 minutes per frame of simulation. The multi-phase flow formulation of the algorithm provides correct treatment of the air-water interface, resulting in multiple droplet generation. In this experiment and in the next one we used the singly-wetted variant of our setup (3.1).

The swimmer animation shows a predefined sequence of a swimming character interacting with the water in a pool. Occasional air-entrainment and small splashing are visible in this animation. It is important to note that, in a single phase flow (most common in liquid animations nowadays), the air bubbles visible in fig.4 would collapse very fast. Our multi-phase flow implementation handles the situation correctly and makes the animation more realistic. The grid resolution is $256 \times 128 \times 128$, which is important for capturing small scale droplets and bubbles. We noticed that in smaller resolution runs the bubble clusters that are visible here would simply coalesce into larger bubbles due to the volume-of-fluid component of CLSVOF which preserves the relative volumes of liquid and air.

The heart-pump animations feature a functional model of the left side of

the heart (Hurmusiadis et al. (17)), including atrium, the left ventricle, inner valves and vessel stubs (the most prominent being the aorta). The movement of the heart wall produces pressure differentials that lead to a series of spectacular effects, including pulling more water inside the heart or pushing it out forcefully through the aorta. The thin heart walls are treated as doubly wetted surfaces. We present here two simulations. In the first one, the heart is completely submerged and is holding a certain volume of air. The valves open in sequence and the air is blown out through the vessel stubs when the heart pumps (in the first half of the simulation). In the second half of the simulation the hearts movement is stopped, and the air is allowed to leak through the open valves of the pulmonary artery. In the second simulation, the heart is partially submerged in liquid, and the liquid level is the same inside and outside the heart. The pumping motion produces two effects: first, the aortic valve is closed, and a mixture of air and water is blown out through the pulmonary artery. Please note that the turbulent mixture of air and water that is pumped out exits the domain unrestricted (its quite visible in the video) due to our choice of open boundary (outflow) conditions. The second effect is produced when the aortic valve opens and water is spurted out through the aorta, then lands in the water due to gravity. All in all, these are visually compelling animations that showcase well our system. Each simulation ran in about 4 minutes per frame in a $96 \times 96 \times 96$ discrete setup.

5 Conclusions and Future Work

We presented a robust method of interaction between two-phase flow and pre-animated meshes. A second-order multi-phase flow formulation ensures non-diffusive treatment of the air-liquid interface. An outflow boundary condition was introduced and implemented, enabling open-boundary type of simulations. Using this animation tool, one can model for example, interaction of marine life with its surroundings, low Reynolds number swimming (e.g. sperm, micro-organisms), physiological fluids mechanics (blood flow in heart, arteries, blood vessels, veins). As future additions to our system we plan to extend our method to also perform two-way interactions with deformable bodies and identify efficient ways to increase the resolution in our simulations.

6 Acknowledgements

We would like to take the opportunity to extend thanks to Karan Singh, for his help with the animated swimmer model, and to Iuliana Radu who helped with the overall presentation of the paper. We also thank Autodesk for providing

the Maya software.

A Appendix

In order to enforce outflow boundary conditions on all the vertical sides of the domain one defines the live pressure as mentioned in 3.2. This makes the initial live pressure zero on the side boundaries. The resulting Navier-Stokes equations in terms of the live pressure become:

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p^{live} - g(z - z_{surf})\nabla\rho$$

where we simplified the resulting gravity terms. Recall that the density can be expressed in terms of the Heaviside function, namely $\rho = \rho^G + (\rho^L - \rho^G)H(\phi)$. Substitution of this equation for the density into the Navier-Stokes equation above gives the final formulation we gave in 3.2. When solving the Poisson equation for the live pressure we enforce zero boundary conditions. This is equivalent to enforcing vertical boundary conditions corresponding to the hydrostatic pressure for the total pressure, and this enforces the water surface to relax over time to the initial water level. This mechanism ensures open-field boundary conditions and, as a consequence, the water doesn't drain from the domain.

References

[1] Bridson R., Fedkiw R., Anderson J.: Robust treatment of collisions, contact and friction for cloth animation. In *ACM Trans. Graph. (Proceedings of SIGGRAPH)* (2002), vol. 21, pp. 594–603.

[2] Chentanez, N., Goktekin, T.G., Feldman, B.E., O'Brien, J.F. Simultaneous Coupling of Fluids and Deformable Bodies In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2006*

[3] Carlson M., Mucha P., Turk G.: Rigid fluid: Animating the interplay between rigid bodies and fluid. In *Proceedings of SIGGRAPH 2004* (2004), Hart J., (Ed.), vol. 23, ACM Press / ACM SIGGRAPH, pp. 377–384.

[4] Dommermuth D., Sussman M., Beck R., O'Shea T., Wyatt D.: The numerical simulation of ship waves using cartesian grid methods with adaptive mesh refinement. In *Proceedings of the 25th Symposium on Naval Hydrodynamics* (St. John's, New Foundland and Labrador, Canada, 2004).

[5] Enright D., Fedkiw R., Ferziger J., Mitchell I.: A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics 183* (2002), 83–116.

[6] Enright D., Marschner S., Fedkiw R.: Animation and rendering of complex water surfaces. *ACM TOG 21*, 3 (2002), 736–744.

[7] Foster N., Fedkiw R.: Practical animation of liquids. *Proceedings of SIGGRAPH 2001*, 23–30 (2001).

[8] Foster N., Metaxas D.: Realistic animation of liquids. *Graphical Models and Image Processing 58* (1996), 471–483.

[9] Feldman B., O'Brien J. F., Klingner B.: Animating gases with hybrid meshes. In *Siggraph 2005* (2005), ACM Proceeedings, ACM, ACM Press / ACM SIGGRAPH.

[10] Feldman B. E., O'Brien J. F., Klingner B. M., Goktekin T. G.: Fluids in deforming meshes. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), ACM Press / ACM SIGGRAPH.

[11] Genevaux O., Habibi A., Dischler J.-M.: Simulating fluid-solid interaction. In *Graphics Interface* (2003), CIPS, Canadian Human-Computer Communication Society, pp. 31–38.

[12] Glowinski R., Pan T.-W., Tesla T., Joseph D.: A distributed Lagrange multiplier/fictious domain method for particulate flows. *International Journal of Multiphase Flow 25*, 5 (1999), 755–794.

[13] Griffith B. E., Hornung R. D., McQueen D. M., Peskin C. S.: An adaptive, formally second order accurate version of the immersed boundary method. *Journal of Computational Physics*, in press (2006).

[14] Guendelman E., Selle A., Losasso F., Fedkiw R.: Coupling water and smoke to thin deformable and rigid shells. *ACM Transactions on Graphics 24*, 3 (2005), 973–981.

[15] Hong J.-M., Kim C.-H.: Animation of bubbles in liquid. *Computer Graphics Forum 22*, 3 (2003), 253–262.

[16] Hong J.-M., Kim C.-H.: Discontinuous fluids. *ACM Transactions on Graphics 24*, 3 (2005), 915–920.

[17] Hurmusiadis V., Briscoe C.: 2005. A Functional Heart Model for Medical Education. In *Proceedings of Functional Imaging and Modelling of the Heart: Third International Workshop, FIMH 2005*.

[18] Keiser R., Adams B., Gasser D., Bazzi P., Dutre P., Gross M.: A unified Lagrangian approach to solid-fluid animation. *Proceedings of Eurographics Symposium on Point-Based Graphics* (2005).

[19] Byungmoon Kim, Yingjie Liu, Ignacio Llamas, Jarek Rossignac: FlowFixer: Using BFECC for Fluid Simulation In *Proceedings of Eurographics Workshop on Natural Phenomena* (2005)

[20] Kondoh N., Sasagawa S., Kunimatsu A.: Creating animations of fluids and cloth with moving characters. In *Proceedings of ACM SIGGRAPH 2004* (2004), Sketches and Applications, ACM SIGGRAPH, ACM Press.

[21] Jimenez E., Ohta M., Sussman M. A computational study of bubble motion in newtonian and viscoelastic fluids Fluid Dynamics and Materials Processing 1(2) (2005), 97107.

[22] Losasso, F., Shinar, T. Selle, A. and Fedkiw, R.: Multiple Interacting Liquids. In *Proceedings of SIGGRAPH* (2006), ACM TOG vol. 25, 721-728

(2006).

[23] Mauch S.: A fast algorithm for computing the closest point and distance transform. *Tech. rept. CalTech.* (2000).

[24] Mihalef V., Metaxas D., Sussman M.: Animation and control of breaking waves. In *Proceedings of the 2004 Symposium of Computer Animation* (2004), ACM SIGGRAPH/Eurographics, ACM Press, pp. 315–324.

[25] Mihalef V., Unlusu B., Metaxas D., Sussman M., Hussaini Y.: Physics-based boiling simulation In *Proceedings of the 2006 Symposium of Computer Animation*

[26] Monaghan J. J.: Smoothed particle hydrodynamics. *Annu. Rev. Astron. Physics 30* (1992), 543.

[27] Muller M., Schirm S., Teschner M.: Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics. *Technology and Health Care 12*, 1 (2004), 25–31.

[28] Muller M., Schirm S., Teschner M., Heidelberger B., Gross M.: Interaction of fluids with deformable solids. *Computer Animation and Virtual Worlds 15*, 3 (2004).

[29] Nguyen, D., Fedkiw, R. and Jensen, H.: Physically Based Modeling and Animation of Fire. In *Proceedings of SIGGRAPH* (2002), ACM TOG vol. 21, 721-728 (2002).

[30] Peskin C. S.: The immersed boundary method. *Acta Numerica 11* (2002), 479–517.

[31] Rasmussen N., Enright D., Nguyen D., Marino S., Sumner N., Geiger W., Hoon S., Fedkiw R.: Directable Photorealistic Liquids. In *Proceedings of the 2004 Symposium of Computer Animation* (2004), ACM SIGGRAPH/Eurographics, ACM Press, pp. 193–202.

[32] Song O.Y., Shin H., Ko H.S.: Stable but non-dissipative water. *ACM Transactions on Graphics 24*, 1 (2005), 81–97.

[33] Sussman M., Puckett E. G.: A coupled level set and volume of fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics 162* (2000), 301–337.

[34] Sussman M., Smereka P., Osher S.: A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics 114* (1994), 146–159.

[35] Sussman M., Smith K.M., Hussaini M.Y., Ohta M., Zhi-Wei R.: A sharp interface method for incompressible two-phase flows. *Journal of Computational Physics*, 2006, to appear.

[36] Stam J.: Stable fluids. *ACM SIGGRAPH 1999* (1999), 121–128.

[37] Takahashi T., Fujii H., Kunimatsu A., Hiwada K., Saito T., Tanaka K., Ueki H.: The simulation of fluid-rigid body interaction. In *Siggraph 2002*, Sketches and Applications, 266, ACM, ACM Press / ACM SIGGRAPH.

[38] Takahashi T., Fujii H., Kunimatsu A., Hiwada K., Saito T., Tanaka K., Ueki H.: Realistic animation of fluid with splash and foam. In *Proceedings of Eurographics* (2003), pp. 391–399.

[Vue] Vue-d'Esprit: http://www.e-onsoftware.com. ewblock *Journal of Com-*

putational Physics 169 (2001), 556–593.

[39] Yang X., James A. J., Lowengrub J., Zheng X., Cristini V.: An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics 217* (2006), pp. 364–394.

Figure 1

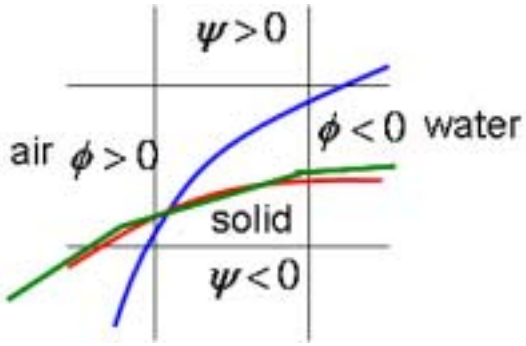


Figure 2

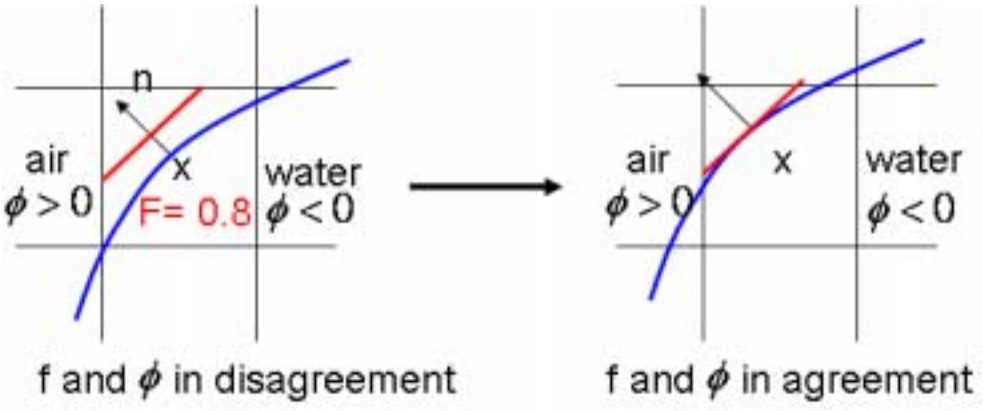

n
air
$\phi > 0$
x
F= 0.8
water
$\phi < 0$
f and ϕ in disagreement
air
$\phi > 0$
x
water
$\phi < 0$
f and ϕ in agreement

Figure 3

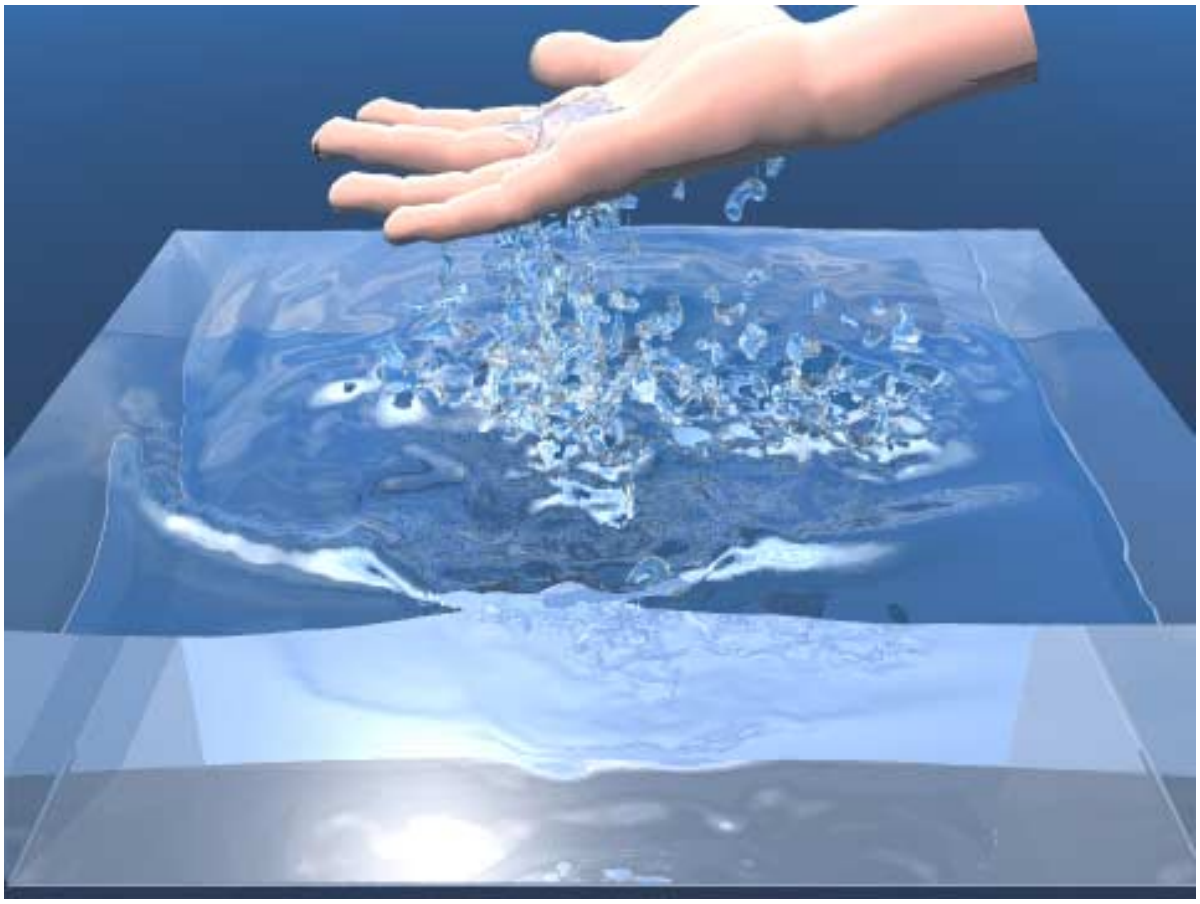


Figure 4

Figure 5

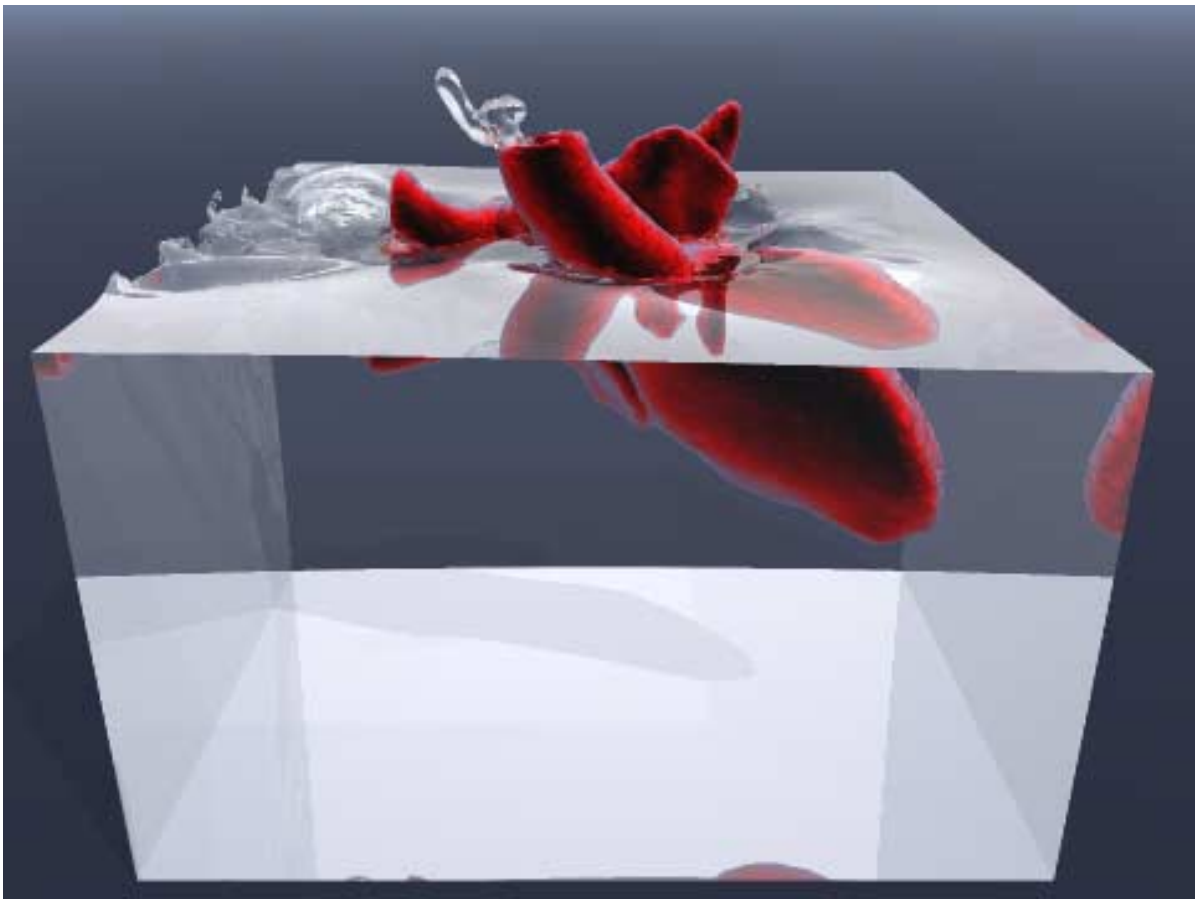
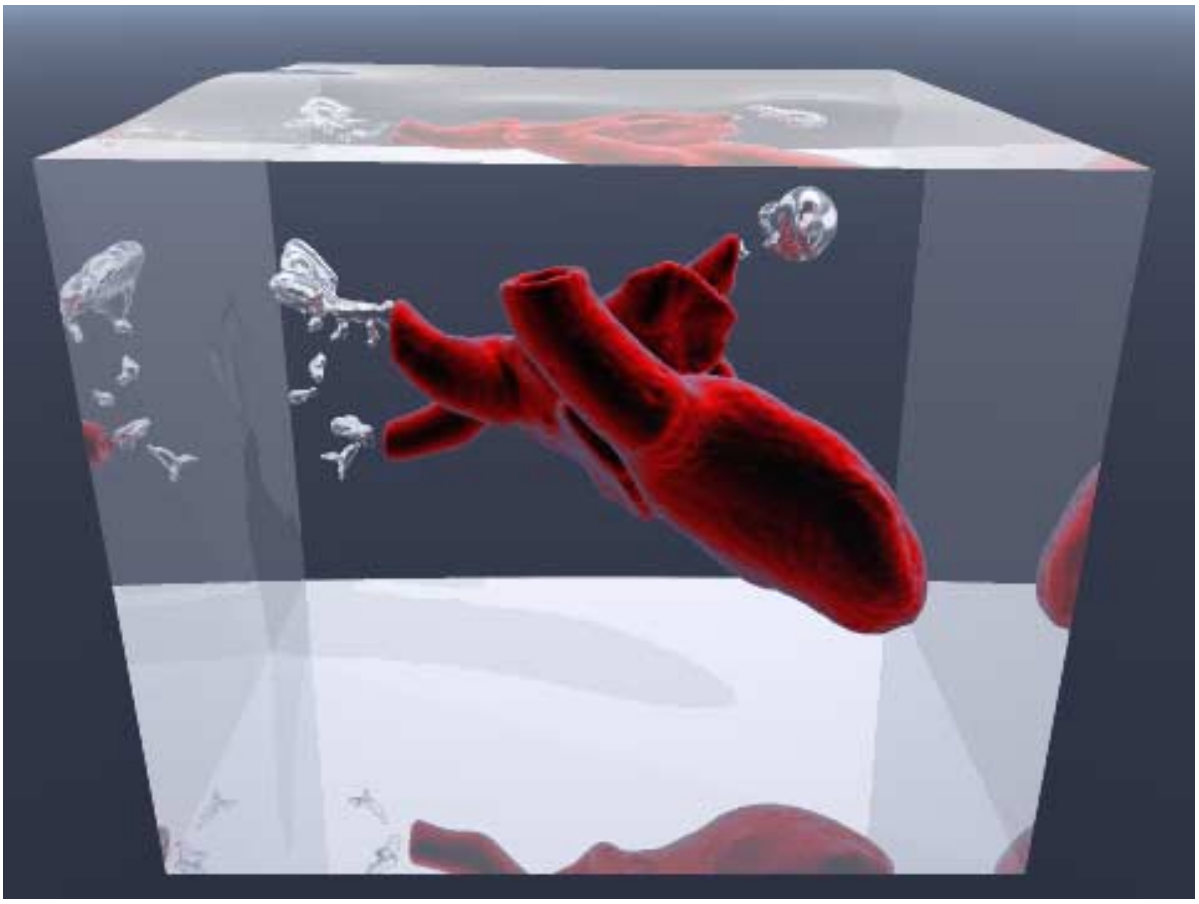


Figure 6



Hand Simulation

Click here to download Supplementary Material: hand.mov

Swimming Simulation

Click here to download Supplementary Material: swimmerhres.mov

Heart Simulation Bubbles

Click here to download Supplementary Material: heart_bubble.mov

Heart Simulation Spurting

Click here to download Supplementary Material: heart_spurt.mov