A Practical Simulation of Dispersed Bubble Flow

Doyub Kim
Seoul National University

Oh-young Song
Sejong University

Hyeong-Seok Ko
Seoul National University

Abstract

In this paper, we propose a simple and efficient framework for simulating dispersed bubble flow. Instead of modeling the complex hydrodynamics of numerous small bubbles explicitly, our method approximates the average motion of these bubbles using a continuum multiphase solver. Then, the subgrid interactions among bubbles are computed using our new stochastic solver. Using the proposed scheme, we can efficiently simulate complex scenes with millions of bubbles.

CR Categories: I.3.7 [Computing Methodologies]: Computer Graphics—Three-Dimensional Graphics and Realism;

Keywords: bubble dynamics, two-phase flow, dispersed bubble flow, level set method, fluid animation

1 Introduction

Attractive underwater scenes always contain bubbles. Especially, viewers are fascinated by the interesting movements of multitudes of small bubbles rising through the water. At the macroscopic level, the global tendency of bubble movements roughly corresponds to our intuitions about fluid dynamics, such as buoyancy, swirl, and diffusion. At the microscopic level, however, each bubble is dispersed in a chaotic and unpredictable manner. Such microscopic flow patterns, in which numerous small bubbles are distributed in the two-phase flow regime, is called *dispersed bubble flow*. The chaotic motion and complexity of this bubble distribution makes numerical simulation of dispersed bubble flow very challenging.

Small numbers of bubbles can be simulated by multiphase fluid solvers combined with a Eulerian surface tracking framework, and various such methods have been introduced to the computer graphics community [Hong and Kim 2003; Hong and Kim 2005; Song et al. 2005; Zheng et al. 2006; Kim et al. 2007]. However, the methods developed to date have focused on handling relatively large bubbles due to the numerical limitations of grid-based solvers. Alternatively, particle–particle interaction models for simulating thousands of small bubbles have also been presented [Müller et al. 2005; Cleary et al. 2007; Hong et al. 2008; Mihalef et al. 2009]. However, simulation of millions of dispersed bubbles using these models is impractical, as it requires the direct computation of the interactions among vast collections of particles. The impracticality of these models is not just a matter of computation time, but also of system stability.

As a means of reducing the computational burden, a multi-layering framework has become prevalent in recent movies with special effects [Geiger et al. 2006]. In this framework, multiple particle layers are added for dispersed droplet effects such as spray, mist, and white-foam. The movements of these droplets are so simple that they can be controlled by artists in a manual (or pseudo-physical) way. However, the movements of the dispersed bubbles are affected by bubble–bubble and bubble–liquid interactions, making them too complicated to be mimicked by artists.

Figure 1: *Simulation of rising bubbles generated around hot objects. The simulation was carried out using a 192×256×128 regular grid with a maximum of about 1,160,000 bubbles. The simulation took about 15 seconds per animation frame.*

In this paper, we present a new framework for realistically and efficiently simulating dispersed bubble flow. As mentioned above, the most challenging aspect of simulating such flows is the computation of the individual interactions among large numbers of particles and the integration of physical contributions from subgrid bubble structures into the fluid solver. Given that, at present, reproducing such a situation using direct simulation of governing equations is not practically feasible, in this work we take a spatial averaging and stochastic approach. Instead of computing the discrete boundary condition of two different media explicitly, we model the average motion of the dispersed region of bubble flow as a continuum. Based on a grid-based solver, the overall bubble–bubble and bubble–liquid interactions can be solved using this continuum model. Also, to simulate the complex interactions among bubbles, we introduce a new stochastic approach for simulating the subgrid dynamics of bubbles. By separating the computation into two stages, the global motion of bubbles can be effectively solved using the continuum solver, while still maintaining the visual complexity of dispersed bubble flow by means of the stochastic solver.

ACM Reference Format
Kim, D., Song, O., Ko, H. 2010. A Practical Simulation of Dispersed Bubble Flow.
ACM Trans. Graph. 29, 4, Article 70 (July 2010), 5 pages. DOI = 10.1145/1778765.1778807
http://doi.acm.org/10.1145/1778765.1778807.

2 Previous Work

Hong and Kim [2003] introduced the volume-of-fluid method (VOF) [Hirt and Nichols 1981] to simulate bubble dynamics. Subsequently, Hong and Kim [2005] adopted the ghost fluid method [Kang et al. 2000] to capture the discontinuity of physical properties across the bubble–liquid interface. Song et al. [2005] presented a two-phase fluid solver for simulating systems containing both gas and liquid phases . Mihalef et al. [2006] simulated boiling water by applying a two-phase version of the coupled level set and volume-of-fluid (CLSVOF) method of [Sussman 2003], and Kim and Carlson also proposed simple boiling framework [2007]. Zheng et al. [2006] developed a regional level set method to capture the very thin interfaces of bubbles, and Kim et al. [2007] extended this work by introducing a volume control method for bubbles and foams. Alternatively, based on smoothed particle hydrodynamics (SPH) [Monaghan 1992], Muller et al. [2005] developed a two-phase fluid solver and Cleary et al. [2007] devised a simulation method for bubbling and frothing liquids. These methods mainly focus on the dynamics of large bubbles or gas–liquid interfaces.

To capture motion of bubbles whose diameters are much smaller than the grid resolution, particle models have been applied to the grid solver. Based on the particle level set method [Enright et al. 2002], Greenwood and House [2004] changed escaped particles into small bubbles. Hong et al. [2008] extended this hybrid framework by designing SPH-based dynamics for escaped bubble particles. Similarly, Thürey et al. [2007] coupled SPH bubbles to a shallow water framework, and Mihalef et al. [2009] incorporated a particle model into a grid-based simulator using the marker level set. However, none of these methods was applied to dispersed bubble flows including millions of small bubbles, such as that shown in Figure 1.

3 Our Method

To couple dispersed bubbles with a conventional gas–liquid solver, the gas and liquid volumes are represented by a level set surface, and the dispersed bubbles are represented by spherical particles. In this section, we first review the basic fluid solver, then describe how to compute the average motion of a flow, and finally present the subgrid bubble dynamics solver.

3.1 Basic Fluid Solver

In this paper, we assume viscous incompressible two-phase flow when modeling the gas-liquid dynamics. The incompressible Navier–Stokes equations are given by the momentum conservation equation

$$\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p/\rho = \nabla\cdot(\mu\nabla\mathbf{u})/\rho + \mathbf{f}/\rho \tag{1}$$

and the mass conservation equation

$$\nabla\cdot\mathbf{u} = 0, \tag{2}$$

where $\mathbf{u}$, p, ρ, μ, and $\mathbf{f}$ represent the velocity, pressure, density, viscosity, and external force, respectively. We solve the above equations using the stable fluids framework of Stam [1999]. In addition, we used the level set method [Osher and Fedkiw 2002] to capture and track the gas-liquid interface, which defines signed-distance function ϕ such that $|\nabla\phi| = 1$ for all domains and the interface is defined at $\phi = 0$. Also, we define the region of $\phi < 0$ as a liquid, and $\phi > 0$ as a gas.

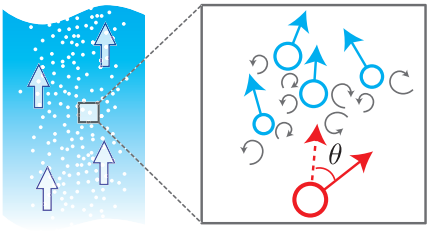

Figure 2: *Overview of the dynamics of dispersed bubble flow. θ indicates the altered direction (see equation 9).*

3.2 Simulating Dispersed Flow as a Continuum

When a liquid and gas are in contact, they are separated by a clear interface. This discontinuity generates complex hydrodynamics [Hong and Kim 2005]. For a dispersed flow, however, the length scale of the interface shape is much smaller than the grid size. Therefore, instead of computing the discrete boundary condition of two different media explicitly, we encode volume information of the media into a fraction field, which is a spatial average of each medium as a continuum. Here, information from both level set and bubble particles is transferred to the fraction field, so that the global dynamics of the liquid, large level set bubbles, and microscale bubbles can be computed in a single framework.

First, we show how to compute liquid fraction field f, and utilize this information to couple small bubble particles with liquid or large bubbles. To convert the level set field to a fraction field, we use $f^{\phi} = H(\phi)$, where H is a Heaviside (step) function. Here, we use a smoothed Heaviside function, which can be written as,

$$H(\phi) = \begin{cases} 0 & \phi < -\epsilon, \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi}\sin(\frac{\pi\phi}{\epsilon}) & -\epsilon \le \phi \le \epsilon, \\ 1 & \epsilon < \phi. \end{cases} \tag{3}$$

We employed $\epsilon = 1.5h$, where h is size of the grid cell. Using this intermediate fraction field f^{ϕ}, the contribution of bubble particles is added. We can write this in the following simple equation:

$$f_{cell} = f^{\phi}_{cell} - \sum_{i\in cell}\frac{4\pi}{3}(r_i/h)^3, \tag{4}$$

where r_i is the radius of particle i in a grid cell.

After computation of fraction field f, a pressure Poisson equation

$$\nabla\cdot(\nabla p^*/\rho) = \nabla\cdot\mathbf{u}, \tag{5}$$

which results from equations 1 and 2, can be discretized as follows

$$\sum_{faces}(p^*_a - p^*_b)/\rho_{face} = \sum_{faces}\mathbf{u}_{face}\cdot\mathbf{n}_{face} \tag{6}$$

where $p^* = \Delta t\, p$, $\sum_{face}$ is the summation for the surrounding faces between cell a and its neighboring cells b, and $\mathbf{n}_{face}$ is the cell-face normal. Here, the density at the face, ρ_{face}, can be computed as follows using the fraction field,

$$\rho_{face} = \rho_{gas} + (\rho_{liquid} - \rho_{gas})f_{face}. \tag{7}$$

The same approach can be easily applied to compute μ_{face} in a Poisson equation from implicit integration of the diffusion equation. After computing the pressure and viscosity, the bubble particles are advected according to the velocity field.

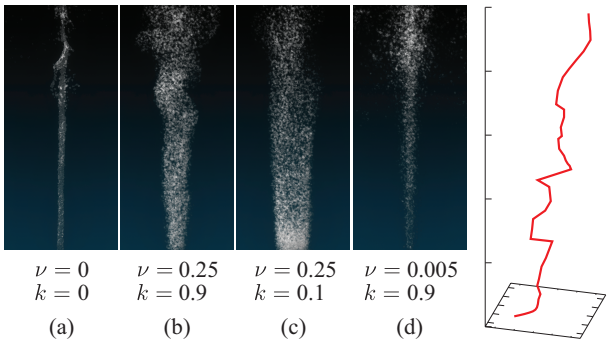

Figure 3: *Simulation of rising bubbles. (a), (b), (c), and (d) show the results obtained using different ν and k values. The right-most plot shows the unstable path of the 13,000th bubble particle of (b).*

This fractional density/viscosity formulation naturally generates buoyancy and swirl fields in the cells of dispersed bubbles. Instead of our formulation, a sophisticated approach such as the two-fluid model [Ishii 1975], which analyzes two-phase flow as two separated fluid layers, may be applied. However, such models incur a greater computational cost because most of the governing equations must be computed twice, and require twice as much memory to store two fluid layers. Also, the relative motion between bubbles and liquid in a dispersed flow is much smaller than in bubbly flows with fewer and larger bubbles [Trafalis et al. 2005]. Thus, it is more efficient to model dispersed flows as a single averaged velocity field.

3.3 Stochastic Solver for Subgrid Bubble Dynamics

Each rising bubble creates a wake that affects other bubbles (including itself), giving rise to complicated hydrodynamics. Although it is theoretically possible to accurately capture the vortices generated by each bubble, this approach is impractical when attempting to simulate huge numbers of microscale bubbles.

Instead, we model this phenomenon as a discrete random walk. As shown in Figure 2, a bubble particle is scattered due to the turbulent flow generated by other bubbles. Thus, when a bubble particle approaches other, sparsely distributed bubbles, it may be scattered by those bubbles, or it may be unaffected and move freely to another position. To model this process, let $s(\mathbf{x})$ be the probability function over space where the scattering of a bubble might occur. Here, we define this function as

$$s(\mathbf{x}) = \nu\rho_{gas}[1 - f(\mathbf{x})]|\mathbf{u}(\mathbf{x})|^2, \tag{8}$$

where ν is the user-specified scattering frequency and $1 - f$ is the fraction of gas. Similarly to Brownian motion, which results from the combined effects of intra- and intermolecular interactions, the velocity $\mathbf{u}$ is squared to reflect the kinetic energy density of bubbles ($\rho_{gas}[1 - f(\mathbf{x})]|\mathbf{u}(\mathbf{x})|^2$). By this definition, more scattering events will occur when the concentration of bubbles and the magnitudes of their velocities are high. For each particle, we measure $s(\mathbf{x})$ after updating the average particle velocity (described in the previous section). If a certain particle is determined to be scattered, the direction of that particle's velocity is altered. Since the volume of the individual particle is relatively smaller than the volume of a grid cell, the particle's new direction can be calculated using Schlick's phase function [Blasi et al. 1993] for tracing massless particles in participating media, which can be expressed as,

$$\cos\theta = (2\xi + k - 1)/(2k\xi - k + 1) \tag{9}$$

where θ, $\xi \in [0, 1]$, and $k \in [-1, 1]$ represent the altered direction (in radian), a uniform random number, and the scattering coefficient, respectively. Here, $k = 0$ gives isotropic scattering, $k > 0$ is forward scattering, and $k < 0$ is backward scattering. The Schlick phase function, which is an approximation of the empirical Henyey-Greenstein phase function, determines the direction of particle after particle-dust collision event based on the distribution of dust. We adopted this function and reinterpreted it to determine the direction of bubble after bubble-turbulent vorticities collision event based on the distribution of bubbles (sources of the turbulent vorticities).

Of course, the Schlick function (equation 9) was not originally invented for computing diffusion process of suspended particles under turbulence flow. In the aspect of computational fluid dynamics (CFD), it could be right to introduce stochastic differential equations (SDE) [Kloeden and Platen 1992; Yuu et al. 1978], such as Langevin equation or Fokker-Plank equation to solve accurate distribution of particles. In our simulation, however, each bubble is very light compared to the surrounding liquid, and is reasonable to compute its path with mass-less particle tracer - the Schlick function. Most of all, the Schlick phase function can be easily controlled by artists with intuitive parameters, and it is also very simple to be implemented. Thus, as shown in Figure 3, it can produce various kinds of visual effects without struggling with complex physical parameters.

After solving equation 9, momentum of each bubble should be transferred to grids. This process is computed be defining scattering force $m_i(\mathbf{u}_i^{altered} - \mathbf{u}_i)/\Delta t$, which can be applied to the cell containing particle i, as an external force $\mathbf{f}$ of equation 1. Here, $m_i = \rho_{gas}(4\pi/3)r_i^3$ is the mass of particle i. Figure 3 shows the results obtained using various values of ν and k. Note that simulations with large ν or small k exhibit diffusive characteristics. Solving equation 8 and 9 is similar to simulating Brownian motion, which describes the molecular motion of diffusion processes. Thus, it is natural that ν and k control the diffusiveness of the particle distribution. In addition, examination of the trajectory of a single bubble (Figure 3) reveals zig-zag and spiral oscillatory motions, thus demonstrating that our method can reproduce a bubble path instability [Shew and Pinton 2006; Hassan et al. 2008].

3.4 Break-up of Bubbles

To simulate break-up events of microscale bubbles, we adopt Kolmogorov's concept of viewing solid particle breakup as a discrete random process [Kolmogorov 1949]. For each particle, a random number $\xi \in [0, 1]$ is compared to a user-specified break-up frequency γ. If ξ is smaller than γ, the bubble particle is broken into two particles, each half the volume of the initial particle (radius of $1/\sqrt[3]{2}$), and the resulting particles are scattered using equation 9.

4 Results

All experiments reported here were performed on an Intel Core i7 920 2.66 GHz processor with 6 GB of memory. We used a uniform regular staggered grid for discretization. For computing the advection term in equation 1, we used the unsplit semi-Lagrangian CIP method [Kim et al. 2008]. Also, our solver is parallelized using OpenMP library. We used 8 threads with full 4 cores with hyper-threading enabled ($\times 2$ threads) for the computation, which is the default setting using OpenMP in Intel Core i7 920. Each part of the simulator shows different scale-up factor: our multigrid solver did not show good scalability performance, while the advection and level set solver have shown relatively good performance. It depends on the scene complexity, but multigrid solver took more than 60% of the total computation time for most of the experiments. Updating bubbles step was easily parallelized, except for the accumulation step, which converts bubbles to the fraction field (since it requires

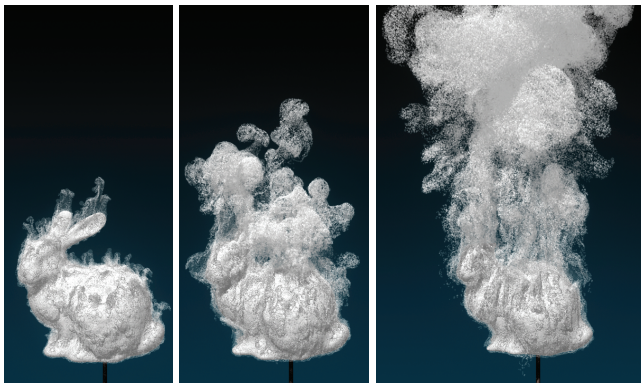

Figure 4: *Simulation of bubbles generated around a melting bunny object.*

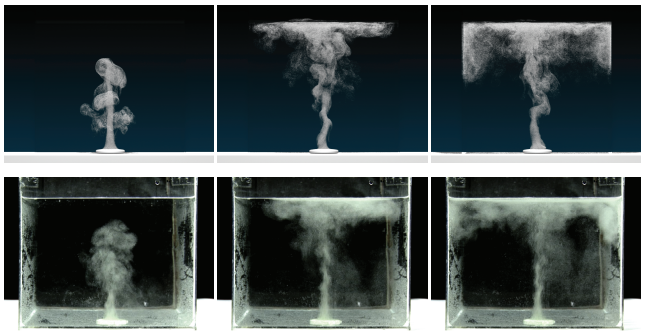

Figure 5: *Comparison of simulated and real systems in which a bubble agent is placed at the bottom of a water tank. The top image row shows the simulator output, and the bottom image row shows the video images.*

critical section to protect from race conditions). For all of our experiments, we used $1.25kg/m^3$ for ρ_{gas}, $1000kg/m^3$ for ρ_{liquid}, $2.0 \times 10^{-5}kg/ms$ for μ_{gas}, $1.0 \times 10^{-3}kg/ms$ for μ_{liquid}, 0.25 for ν, 0.9 for k, and 10^{-4} for γ.

Figure 1 shows the result of a simulation of bubbles generated around hot objects. In this experiment, we did not simulated boiling as shown in [Mihalef et al. 2006], but just simply emitted bubbles from randomly sampled points on the object's surface. The simulation was carried out using a $192 \times 256 \times 128$ grid with a maximum of approximately 1,160,00 particles. Approximately 15 seconds were required to advance a single animation frame (1/30 seconds).

Figure 4 shows the result of a simulation of bubbles generated around a melting bunny-shaped object. This animation of melting was created heuristically by subtracting a constant value from the level set field of the bunny model. The simulation was carried out using a $160 \times 192 \times 96$ grid with a maximum of approximately 1,570,000 particles. The simulation took only about 7 seconds to advance a single animation frame.

In Figure 5, we assess the quality of our simulation (Figure 5 (a)) by comparing the results with video taken from a real-world experiment (Figure 5 (b)). When a bubble agent is placed at the bottom of a water tank, a massive number of microscale bubbles are generated from the agent's surface. Although the individual path of each bubble is clearly different, the overall motion of the simulated bubbles qualitatively resembles the real bubble motions. Both the simulated and real systems show an unstable rising pattern of bubbles and smoke-like diffusion after reaching the water surface. The simulation was carried out using a 80^3 grid with a maximum of approximately 2,600,000 particles. The simulation took about 16 seconds per single animation frame.

5 Conclusion

In this paper, we have presented a simple and fast method for simulating dispersed bubble flow. To simulate a large set of bubbles, we separated the direct numerical solver into two stages. First, instead of computing the discrete boundary condition between gas and liquid, we took a spatial averaging approach to interpret dispersed bubble flow as a continuum. Then, the subgrid interactions among bubbles were calculated using our new stochastic solver. As a result, we could simulate complex scenes with millions of microscale bubbles in less than 16 seconds per frame on a single PC.

Our limitation mainly comes from the approximation of bubble dynamics, which does not correctly handle per-bubble collision, merging, shape variation, and overlapping problem. In order to avoid this kind of problems, one should perform proximity test or collision detection between large set of bubbles. Without the consideration of direct interaction between bubbles, sometimes they will get overlapped, or introduce such kind of artifacts. However, performing per-bubble interaction will eventually reduce the performance of the solver. Thus, we approximated bubble interaction by accumulating them into the fraction field.

Future extensions of our method include simulation of other kinds of particulate flow, which involves dust or sand, and simulation of mixed water sprays and bubbles, which can be observed from waterfalls or breaking waves.

Acknowledgements

First of all, we would like to thanks to Eunchan Jo, Ji-Young Park, and Oh-Hwan Kwon at Sejong University for helping us doing the experiment of Figure 5. This work was supported by the Brain Korea 21 Project in 2010, Seoul Research and Business Development Program (10557), Seoul Metropolitan Government R&BD Program (10581), the IT R&D program of MKE/MCST/IITA (ITAA1100090201260001000100100, Development of Digital Clothing Software Technology), ASRI (Automation and Systems Research Institute at Seoul National University), Ministry of Science and Technology under National Research Laboratory (NRL) grant 20100008010, and Ministry of Culture, Sports and Tourism (MCST) and Korea Culture Content Agency (KOCCA) in the Culture Technology(CT) Research & Developement Program 2010.

References

BLASI, P., SAEC, B. L., AND SCHLICK, C. 1993. A rendering algorithm for discrete volume density objects. *Comput. Graph. Forum 12*, 3, 201–210.

CLEARY, P. W., PYO, S. H., PRAKASH, M., AND KOO, B. K. 2007. Bubbling and frothing liquids. *ACM Trans. Graph. 26*, 3, 97.

ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and rendering of complex water surfaces. *ACM Trans. Graph. 21*, 3, 736–744.

GEIGER, W., LEO, M., RASMUSSEN, N., LOSASSO, F., AND FEDKIW, R. 2006. So real it'll make you wet. In *ACM SIGGRAPH 2006 Sketches*, 20.

GREENWOOD, S. T., AND HOUSE, D. H. 2004. Better with bubbles: enhancing the visual realism of simulated fluid. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 287–296.

HASSAN, N. M. S., KHAN, M. M. K., AND RASUL, M. G. 2008. a study of bubble trajectory and drag co-efficient in water and non-newtonian fluids. *WSEAS Transactions on Fluid Mechanics 3*, 3, 261–270.

HIRT, C. W., AND NICHOLS, B. D. 1981. Volume of fluid (vof) method for the dynamics of free boundaries. *J. Comp. Phys. 39*, 1, 201–255.

HONG, J.-M., AND KIM, C.-H. 2003. Animation of bubbles in liquid. *Comput. Graph. Forum 22*, 3, 253–262.

HONG, J.-M., AND KIM, C.-H. 2005. Discontinuous fluids. *ACM Trans. Graph. 24*, 3, 915–920.

HONG, J.-M., LEE, H.-Y., YOON, J.-C., AND KIM, C.-H. 2008. Bubbles alive. *ACM Trans. Graph. 27*, 3, 48.

ISHII, M. 1975. *Thermo-Fluid Dynamic Theory of Two-Phase Flow*. Eyrolles, Paris.

KANG, M., FEDKIW, R. P., AND LIU, X.-D. 2000. A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput. 15*, 3, 323–360.

KIM, T., AND CARLSON, M. 2007. A simple boiling module. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 27–34.

KIM, B., LIU, Y., LLAMAS, I., JIAO, X., AND ROSSIGNAC, J. 2007. Simulation of bubbles in foam with the volume control method. *ACM Trans. Graph. 26*, 3, 98.

KIM, D., SONG, O.-Y., AND KO, H.-S. 2008. A semi-lagrangian cip fluid solver without dimensional splitting. *Comput. Graph. Forum 27*, 2, 467–475.

KLOEDEN, P. E., AND PLATEN, E. 1992. *Numerical solution of stochastic differential equations*. Springer-Verlag, Berlin Heidelberg New York.

KOLMOGOROV, A. N. 1949. On the fragmentation of drops in turbulent streams. *Dokl. Akad. Nauk SSSR 66*, 825–828.

MIHALEF, V., UNLUSU, B., METAXAS, D., SUSSMAN, M., AND HUSSAINI, M. Y. 2006. Physics based boiling simulation. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 317–324.

MIHALEF, V., METAXAS, D., AND SUSSMAN, M. 2009. Simulation of two-phase flow with sub-scale droplet and bubble effects. *Comput. Graph. Forum 28*, 2, 229–238.

MONAGHAN, J. 1992. Smoothed particle hydrodynamics. *Ann. Rev. Astron. Astrophys. 30*, 543–74.

MÜLLER, M., SOLENTHALER, B., KEISER, R., AND GROSS, M. 2005. Particle-based fluid-fluid interaction. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 237–244.

OSHER, S., AND FEDKIW, R. 2002. *The Level Set Method and Dynamic Implicit Surfaces*. Springer-Verlag, New York.

SHEW, W. L., AND PINTON, J.-F. 2006. Dynamical model of bubble path instability. *Phys. Rev. Lett. 97*, 144508.

SONG, O.-Y., SHIN, H., AND KO, H.-S. 2005. Stable but nondissipative water. *ACM Trans. Graph. 24*, 1, 81–97.

STAM, J. 1999. Stable fluids. *Computer Graphics (Proc. ACM SIGGRAPH '99) 33*, Annual Conference Series, 121–128.

SUSSMAN, M. 2003. A second order coupled levelset and volume of fluid method for computing growth and collapse of vapor bubbles. *J. Comp. Phys. 187*, 1, 110–136.

THÜREY, N., SADLO, F., SCHIRM, S., MÜLLER-FISCHER, M., AND GROSS, M. 2007. Real-time simulations of bubbles and foam within a shallow water framework. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 191–198.

TRAFALIS, T. B., OLADUNNI, O., AND PAPAVASSILIOU, D. V. 2005. Two-phase flow regime identification with a multiclassification support vector machine (svm) model. *Ind. Eng. Chem. Res. 44*, 12, 4414–4426.

YUU, S., YASUKOUCHI, N., HIROSAWA, Y., AND JOTAKI, T. 1978. Particle turbulent diffusion in a dust laden round jet. *AIChE Journal 24*, 509–519.

ZHENG, W., YONG, J.-H., AND PAUL, J.-C. 2006. Simulation of bubbles. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 325–333.