Modeling Two Phase Flow Dynamics for Deformable Interfaces (Extended Abstract)

Chandrajit Bajaj, Albert Chen, Richard Hankins, Bong-Soo Sohn
Department of Computer Science,
Center for Computational Visualization,
Institute for Computational Engineering and Sciences
University of Texas, Austin, TX 78712
Email: bajaj@cs.utexas.edu

Abstract

We describe a Stokesian (slow viscous) flow model for producing interacting deformable surfaces. This captures several phenomena in their nativity, such as cell membranes interacting, or soft docking of flexible molecules, etc. Starting from any initial configuration of closed, compact "interfaces", the interfaces are continually evolved, as well as deformed based on the relative viscosities and the interfacial tension. The velocity computation and the interfacial dynamics are achieved via a Boundary Element formulation of the governing Stokesian flow equation, while the interface evolution and topology maintenance utilizes level set representation and underlying function updates. Effects such as coalescence, break-up, and additional near interface interactions, can also be accurately captured. These last three effects in particular require adaptive refinement of meshed geometry and controlled coupling of the numerical errors in computation to yield topologically realistic looking phenomological modeling.

1 Introduction

We present a technique to model the interaction of deformable surfaces using two-phase Stokesian (slow viscous) flows. These interfaces can represent air bubbles in a viscous liquid, oil droplets in a suspension, or cellular membranes subjected to hydrodynamics forces.

Two-phase fluid simulations have been a topic of interest in computer graphics. In [5] Foster and Metaxas describe a technique for simulating free-surface water flows by solving the full Navier-Stokes equations in three dimensions via a finite difference scheme and representing the free surface with massless marker particles. In [4] Foster and Fedkiw implement a solution to the full Navier-Stokes equations using a modified version of the semi-Lagrangian technique introduced by Stam in [10] to capture the complex behavior of free water surfaces. To represent the air-water interface, they used a level set method as introduced in [9]. In [3] a particle level set method was introduced for improved interface capturing. Massless marker particles are advected with the level set data and used to repair the level set in regions of degradation due to the use of a coarse grid for animation. The combination of a semi-Langrangian

Navier-Stokes solver coupled to a particle based level set method allows for complex modeling of the dynamics at the air-water interface. In all of the above treatments, the density of the air is assumed to be zero. This is a reasonable approximation as the density of air is 1000 times less than that of water. A constant pressure boundary condition is applied everywhere at the interface and the Navier-Stokes equations are solved only within the liquid region. Using these techniques, while visually realistic simulations can be achieved which capture the convective (and turbulent) interaction of gases and liquids, many interesting features of air-water interfacial dynamics arising from non-convective, non-turbulent, and slow viscous flows cannot be observed. These interfacial deformations and dynamics arise from interfacial tension and the curvature of the interface. In order to capture such interfacial deformations in a simulation, for even visual realism, the Navier-Stokes equations must be accurately solved in both fluid regions, and pressure boundary conditions must be applied at the two phase fluid interface, while carefully orchestrating the accuracy of interfacial geometry and the interfacial velocities. Such an accurate two phase fluid simulation to produce realistic visual animations of deforming interfaces, is the main contribution of this paper.

In [7], Hong and Kim with perhaps similar goals to ours, modified the semi-Lagrangian scheme [10] coupled with the volume-of-fluid method (VOF) introduced in [6] to simulate bubbles in liquids. The authors calculate the interfacial tension, and thereby are able to capture certain interfacial deformations, the VOF method has accuracy limitations for effects such as bubble flattening, coalescence, bubble necking, and break-up, and additional subtle near bubble interaction. We achieve this accuracy through our precise representation of the interface geometry, coupled to a topology tracking technique and a stable and accurate boundary element fluid solver, allowing us to observe these phenomena at significantly higher resolution. Our solution can be broken down into three main sub-areas: accurate boundary element mesh representation of the interface, careful error bounded calculation of physical quantities (velocities, surface tension) on the interface by regularization and adaptive mesh refinement, and topological tracking for precise near-bubble interactions.

For the calculation of physical quantities we use the boundary element method (BEM) on adaptive geometries. Given an initial configuration of bubbles, our adaptive BEM solver, estimates the velocities on the interfacial boundary with greater accuracy, due to several advantages it has over competing methods. It reduces the dimensionality of the problem by one, and focuses computational effort on the boundary which, for two phase simulations, is the region of interest thereby yielding superior accuracy of the BEM over both finite element methods (FEM) and finite difference methods (FDM), where the interface is represented indirectly using a discretized volume domain.

We also present a dynamic remeshing algorithm for smooth evolving interfaces of an objects such as bubbles. The interface is discretized to triangular or quadrilateral mesh where the velocity of each vertex is computed by the BEM. Since an interface changes its geometric shape including surface area and curvature distribution, meshes with fixed vertex count and connectivity cannot represent the evolving interfaces accurately. Therefore, the number of vertices and vertex connectivity need to be adjusted according to given geometric properties of an interface for each timestep. The quality of triangular or quadrilateral elements, often measured with element shape, also needs to be good enough for accurate BEM calculation. The main difficulty occurs when the topology of interfaces change. For instance, two bubbles may coalesce or a single bubble may break up into two bubbles. We utilize up-sampling / down-sampling methods and maintain a dynamic octree for tracking and controlling the mesh topology.

2 Deformable Interfaces

A sketch of our method of accurate two-phase simulations is below. Details of each individual computational step is given in [1].

Algorithm Sketch

1. **Initial interface** : specify an initial configuration and level set representation of bubbles, as well as initialize the fluid physical parameters
2. **Interface parametrization** : quality mesh extraction for the level set
3. **Interfacial velocity computation** : use an accurate BEM to calculate interfacial velocities and analyze its associated errors
4. **Oracle** : check topology change condition through the Oracle and modify underlying functions that define the bubble interface. This process involves channel identification and modification of interfaces.
5. **Interface evolution** : evolve the interface using a level set method
6. **Interference check** : check geometric interference and adjust timestepping, go back to mesh extraction and BEM calculation for continuing the simulation.

2.1 Initialization of bubbles

Initial geometry data is specified and given as inputs to the boundary element solver. The sign distance field to the given initial geometry is computed using an implicit C^2 cubic B-Spline level set function [2]. Further normals and curvatures of the interfaces are then estimated from this piecewise analytic representation.

2.2 Stokes flow and boundary integral equations

Let us denote ρ as the fluid density, μ as the fluid viscosity, u as the fluid velocity field, p as the fluid pressure. In addition, let g represent some external field, such as gravity, that acts on the fluid. The full Navier-Stokes equations for incompressible flow are given by, $\rho(\frac{\partial u}{\partial t} + u \cdot \nabla u) = -\nabla p + \mu \nabla^2 u + \rho g$, and $\nabla \cdot u = 0$. By specializing to flows with low Reynolds number and low acceleration parameter we ignore the inertial and convective terms in the momentum equation. In this limit, viscous forces dominate and we have a linearized version of the Navier-Stokes equations called Stokes equation, $-\nabla p + \mu \nabla^2 u + \rho g = 0$

Our choice of numerical solution technique requires that we reformulate the governing equations as integral equations over the boundary interfaces. Since we are interested in updating the interface between the two fluids we only need an integral relation for the interfacial velocity. The most general expression for the velocity at a point x_0 on a bubble interface Γ is given by [14],

$$u_k(x_0) = \frac{1}{4\pi}\frac{1-\lambda}{1+\lambda}\int_\Gamma u_i(x) T_{ijk}(x, x_0)\hat{n}_k(x) d\Gamma(x) + F_j(x_0) \tag{2.1}$$

$$F_j(x_0) = \frac{2}{1+\lambda}(u_j^{\infty}(x_0) - \frac{1}{8\pi\mu_s}\int_{\Gamma} f(x)\hat{n}_i(x)G_{ij}(x,x_0)d\Gamma(x)) \tag{2.2}$$

2.3 Numerical Technique

In the above relations we assume N bubbles with varying viscosities immersed in a Stokes fluid. The viscosity ratio of the bubble to the fluid it is embedded in is, $\lambda = \frac{\mu}{\mu_s}$, and u_j^{∞} is the asymptotic Stokes velocity that the bubbles are immersed in. The function f is the boundary condition specifying the pressure difference at the interface and is given by, $f = 2\gamma\kappa + (\rho^{fluid} - \rho^{bubble})z$. In the above, γ is the surface tension of the drop, $\hat{n}$ is the normal pointing into the suspending fluid and $\kappa = \frac{1}{2}\nabla \cdot n$ is the extrinsic mean curvature of the boundary. For our calculations we use the free space Green's functions for Stokes flow given by,

$$G_{ij}(x-x_0) = \frac{\delta_{ij}}{r} + \frac{\hat{x}_i\hat{x}_j}{r^3}, \hat{x} = x - x_0$$

and

$$T_{ijk}(x-x_0) = -6\frac{\hat{x}_i\hat{x}_j\hat{x}_k}{r^5}$$

We generate a quadrilateral mesh from our B-spline level set approximation[11], to represent the interface separating the two fluids and discretize the integral equation. We break up the integrals over each surface into a sum of integrals over each quadrilateral face of the mesh. The integration over most faces may be done using standard quadrature techniques. However, the Green's functions appearing in these expressions show divergent behavior as the evaluation point approaches the surface over which we are integrating. These singular and near-singular integrals must be handled carefully in order to obtain accurate results from a BEM when two droplets are close. Details are given in [1].

2.4 Error Analysis

Proper error analysis is important for our algorithm. We use this information to determine the validity of the boundary element calculation and how to improve it by surface mesh refinement. We describe two different error tools that we use for feedback during the simulation. The first one is a global error measure and derives from the incompressibility condition of the governing equations we are using to model the fluids. This condition in integral form becomes,

$$\int_{\Gamma} u \cdot \hat{n} d\Gamma = 0 \tag{2.3}$$

We calculate this value and use it to determine which octree level to mesh the geometry for each timestep. The other method we implement is a local measure of the error. We use bilinear interpolation when calculating values on the surface from the values at the vertices which were obtained through solving the boundary element system. We implement a technique to estimate the error at each face of the mesh and add more geometry if the error is outside acceptable tolerance. The error analysis implemented is that

of [8]. The technique is chosen based on computational speed and ease of implementation. The error on the m-th face is estimated to be

$$E_u(m)^2 = \int_{\Gamma_m} \mid u^0 - \hat{u} \mid^2 d\Gamma_m \tag{2.4}$$

where u^0 is the predicted exact solution which we approximate by higher order interpolation and $\hat{u}$ is the numerical solution for the velocity. Finally we measure the relative error by calculating,

$$E_{rel,u}(m)^2 = \frac{E_u(m)^2}{\int_\Gamma \mid u^0 \mid^2 d\Gamma} \tag{2.5}$$

While rigorous mathematical bounds do not exist for local errors in collocation methods this technique serves its purpose in quantifying the error regions of sparse geometry where the calculation could be improved and have been implemented by authors in boundary element methods as well as finite element methods [12] [13]. This error analysis is done at each time step following the boundary element calculation of the interfacial velocities. Given a user defined tolerance $\varepsilon_1 > 0$ we check that the velocity error computed over each face of the mesh satisfy $E_{rel,u} < \varepsilon_1$. If this condition is satisfied then we proceed with the interface evolution. If it is not satisfied then we refine the faces of the mesh where the tolerance is exceeded and recalculate the interfacial velocities using this refined mesh.

2.5 Topology Control

Assume f^t is a function at time t where its level set represents deformable interfaces. $\widehat{f^t}$ is a piecewise trilinear function that approximates f^t. M^t is a mesh that approximates the level set. The level set topology defined in $\widehat{f^t}$ is preserved during mesh extraction process.

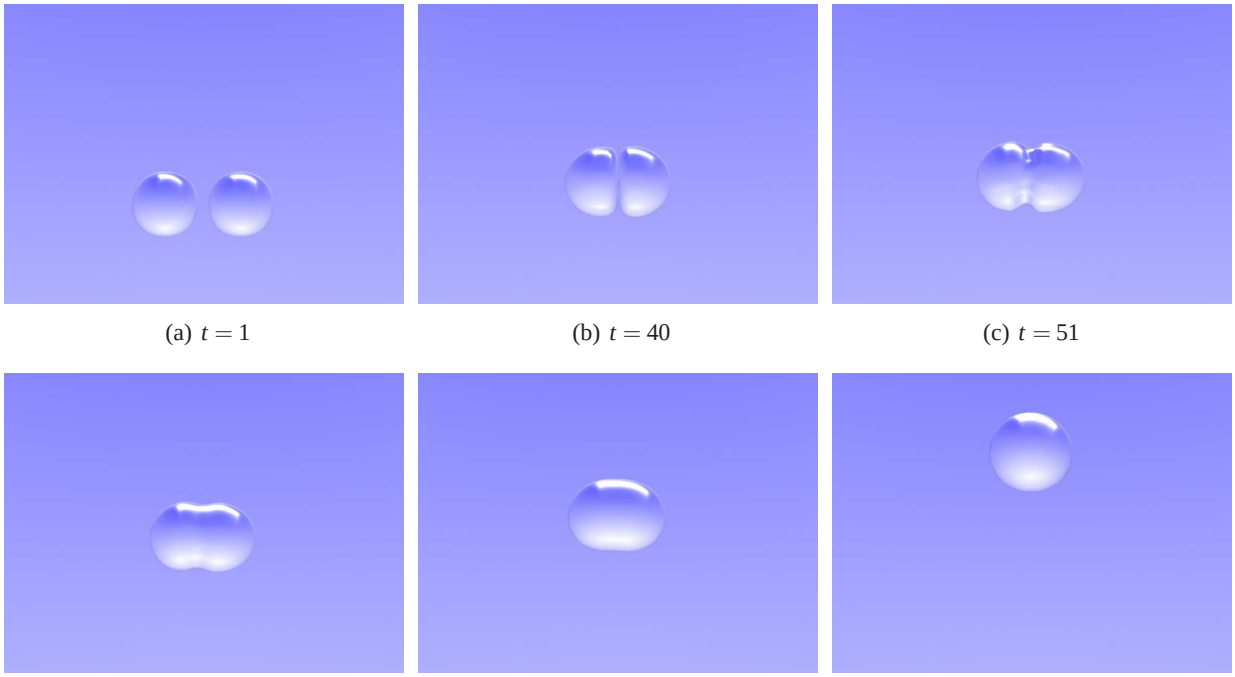
Boundary Element Method (BEM) is applied to computing velocities which can be used for updating the function $\widehat{f^t}$ at time t to evolve the interface in viscous flows. This generates the function $\widehat{f^{t+1}}$ at time t, where level set mesh M^{t+1} can be extracted.

Topology changes of bubbles may occur under various conditions such as minimum distance or contact surface area between two bubbles. The moment of topology changes can be also chosen manually. We introduce an *oracle* that is an independent procedure to decide whether the topology change occurs or not based on the user-specified conditions. We also need a remeshing procedure to actually change the topology of meshes for the bubbles if the oracle decides that. Our algorithms for the oracle and remeshing support coalescence and breakup of bubble interfaces. Details are given in [1].

2.6 Interface Update

The interface is updated using our higher order level set method [2]. The evolution of the level set is governed by the level set equation,

$$\frac{\partial \varphi_i}{\partial t} + \vec{v} \cdot \nabla \varphi_i = 0$$



(a) $t = 1$ (b) $t = 40$ (c) $t = 51$

(d) $t = 66$ (e) $t = 100$ (f) $t = 200$

Fig 3.1: Bubble coalescence. Several timesteps are shown here depicting bubble coalescence. Bubbles don't tend to coalesce easily in pure Stokes flow. For the simulation a term was added to the fluid solver to simulate the intermolecular forces at work that cause coalescence. We see that the bubbles tend to flatten out as they approach each other. A sudden joining occurs after they have been in close proximity for enough time. The joined bubble returns to a spherical shape due to surface tension.

3 Results

3.1 Implementation

We have an implementation of the above technique that runs on a Linux platform. The implementation consists of two main libraries: a meshing library, and a boundary element calculation library. These two sets of code are packaged in a graphical user interface where initial data can be input and physical parameters are defined. The package is also capable of outputting mesh data for analysis and rendering.

Figure 3.1 shows results of a simulation of two bubbles coalescing. The input data are two uniform spheres separated by a small distance. The viscosity ratio is $\lambda = 0.5$ and a small gravity field is enabled so that the bubbles rise slightly due to buoyant forces. There is no asymptotic flow in the suspending fluid. The simulation shows that bubbles tend to exist in a flattened state before intermolecular forces take over and the bubbles ultimately join.

Figure 3.2 shows two droplets deforming as they pass each other in a shearing flow given by $u^{\infty} = (\gamma z, 0, 0)$ where γ is a user defined parameter that controls the strength of the flow that can be used to

(a) $t = 1$ (b) $t = 20$ (c) $t = 30$

Fig 3.2: Bubbles deforming in shear flow. This simulation shows bubbles deforming as they move past each other in a shearing flow. The viscosity ratio for the simulation is 0.5 and a small gravity field is applied to make the bubbles rise slightly. The shear flow is keyframed to slowly dissipate. We observe the shearing and deformations as the bubbles interact and then return to their spherical shapes as the speed of the flow decreases.

control the strength of the field. For this animation we set $\gamma = 1$ and add a small gravity field. We observe the bubbles slowly rising and moving past each other. As the lower bubble rises it begins to move to the right as it moves to a height where the asymptotic flow is positive. The flow is keyframed to gradually dissipate over the coarse of the simulation and we see the bubbles slow down and return to their spherical shapes.

Within our simulations there are several parameters which may be adjusted to create a desired simulation based animation. The user can specify an initial configuration of droplets and a Stokes flow to embed the droplets in. This flow has parameters which control the speed and variation in the pressure gradient along the different spatial axes. In addition, the strength of the gravity field or other long range body forces may also be specified by the user and short range forces may be added to simulate intermolecular interactions. Adjustment of these parameters leads to very different results.

The viscosity ratio parameter is particularly important. As a special case we set $\lambda = 1$ in the boundary integral equation eliminating the integral over the double layer potential. The result is that there is no linear system to solve so interfacial velocities are computed simply by evaluating,

$$u_k(x_0) = \frac{2}{1+\lambda}(u_j^{\infty}(x_0) - \frac{1}{8\pi\mu_s}\int_{\Gamma} f(x)\hat{n}_i(x)G_{ij}(x,x_0)d\Gamma(x))$$

It should be noted that all phenomena of bubble interactions mentioned here can be observed in this special case, but the computational demands are much less than the $\lambda \neq 1$ solution.

Acknowledgment. This research was supported in part by NSF grants IIS-0325550, CNS-0540033 and NIH contracts P20-RR020647, R01-EB00487, R01-GM074258, R01-GM07308

References

[1] C. Bajaj, A. Chen, R. Hankins, and B. Sohn. A C^2 cubic B-Spline Level Set Approach to Simulating Deformable Interfaces. TICAM Report 08-xx, Texas Institute for Computational and Applied Mathematics, The University of Texas at Austin, 2008.

[2] C. Bajaj, G. Xu, and Q. Zhang. A higher order level set method with applications to smooth surface constructions. TICAM Report 06-18, Texas Institute for Computational and Applied Mathematics, The University of Texas at Austin, 2006.

[3] J. Ferziger D. Enright, R. Fedkiw and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comp. Phys.*, 183:83–116, 2002.

[4] N. Foster and R. Fedkiw. Practical animation of liquids. In *ACM Press/ACM SIGGRAPH, E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series*, pages 23–30, 2001.

[5] N. Foster and D. Metaxas. Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5):471–483, 1996.

[6] C.W. Hirt and B.D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *J. Comp. Phys.*, 39:201–255, 1981.

[7] J.-M. Hong and Kim C.-H. Animation of bubbles in liquid. In *In Proceedings of Eurographics 2003*, pages 253–262, 2003.

[8] E. Kita and N. Kamiya. A new adaptive boundary element refinement based on simple algorithm. *Mech Res Commun.*, 18(4):177–86, 1991.

[9] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comp. Phys*, 79:12–49, 1988.

[10] J. Stam. Stable fluids. In *In Proceedings of SIGGRAPH 99, ACM SIGGRAPH/Addison Wesley Longman, Computer Graphics Proceedings, Annual Conference Series, ACM*, pages 121–128, 1999.

[11] Y. Zhang and C. Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering (CMAME)*, 195:942–960, 2006.

[12] O.C. Zienkiewicz and J.Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Int. J. Numer. Methods. Engng.*, 24:337–57, 1987.

[13] O.C. Zienkiewicz, J.Z. Zhu, and N.G. Gong. Effective and practical h-p-version adaptive analysis procedures for the finite element method. *Int. J. Numer. Methods. Engng.*, 28:879–91, 1989.

[14] Z.A. Zinchenko, M.A. Rother, and R.H. Davis. A novel boundary integral algorithm for viscous interaction of deformable drops. *Phys. Fluids*, 9:1493–1511, 1996.