Bubbles and Foam

Benjamin Schroeder

November 20, 2007

Outline

We'll mostly concentrate on one paper, "Bubbling and Frothing Liquids" from SIGGRAPH 2007, and the Smoothed Particle Hydrodynamics technique it uses. I'll also say a few words about the approach of a paper from SCA that uses SPH for foam simulation.

1. Review: Smoothed Particle Hydrodynamics

2. Bubbling and Frothing Liquids
   Cleary et al, SIGGRAPH 2007

3. Real-time Simulation of Bubbles and Foam within a Shallow Water Framework
   Thürey et al, SCA 2007

Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics, also known as SPH, is a particle-based (Lagrangian) method for simulating fluid motion.

A fluid is represented as a set of discrete particles. When quantities such as the pressure at a given point are needed by the simulation, they are obtained by summing contributions from nearby particles.

We'll follow the treatment in

> Particle-based Fluid Simulation for Interactive Applications
> Müller et al, SCA 2003

Previous papers in graphics used the technique to model fire and highly-deformable surfaces.

Smoothed Particle Hydrodynamics

As usual in a particle system, we want to track each particle's position, velocity, and acceleration. The velocity and position can be determined by integrating the acceleration; the acceleration is a result of the forces acting on the particle.

For a particle *i*, we have

$$\mathbf{a}_i = \frac{\mathbf{f}_i}{\rho_i}$$

where

f_i is the sum of all forces acting on the particle, and
ρ_i is the density at the particle's location.

Smoothed Particle Hydrodynamics

We can derive this by considering a form of the equations usually used in Eulerian (grid-based) systems:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0;$$

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}$$

where

- ρ is the density field,
- **v** the velocity field,
- p the pressure field,
- **g** an external force field,
- and μ the fluid's viscosity constant.

Smoothed Particle Hydrodynamics

The first equation represents conservation of mass.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0;$$

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}$$

Since we have a constant number of particles, each with constant mass, we can ignore this for our computation.

Smoothed Particle Hydrodynamics

The remaining equation is a form of the Navier-Stokes equation for incompressible fluids.

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = -\nabla p + \rho\mathbf{g} + \mu\nabla^2\mathbf{v}$$

Note that the velocity term on the left-hand side is the *substantial derivative* of the velocity. This is the rate of change of the velocity of a fluid element as it moves through the fluid. Here, the particles move with the fluid and so the substantial derivative is simply the time derivative.

$$\rho\left(\frac{d\mathbf{v}}{dt}\right) = -\nabla p + \rho\mathbf{g} + \mu\nabla^2\mathbf{v}$$

Smoothed Particle Hydrodynamics

Rearranging terms, we get the acceleration equation from before.

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{-\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}}{\rho}$$

The force is made up of three components: *pressure*, *external forces*, and *viscosity*.

We will apply external forces, such as gravity, directly to each particle (they don't vary with the density). This leaves us needing to consider how to compute *pressure*, *viscosity*, and the *density* itself.

Smoothed Particle Hydrodynamics

We said earlier that the SPH rule was to compute quantities like the density or pressure by summing contributions from nearby particles.

It uses a kind of weighted sum to do this, multiplying the contribution of each particle by a smoothing function. In particular, for a scalar quantity A_S, we can sum the contribution of each particle *j* using

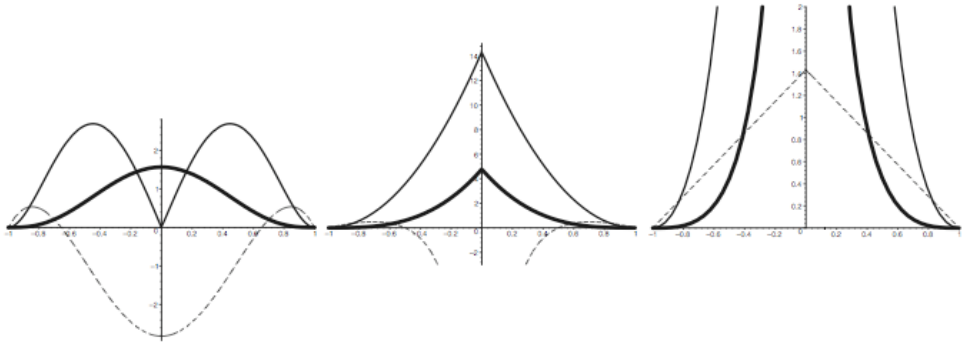
$$A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h)$$

where

m_j is the mass of particle *j*,
$\mathbf{r}_j$ is the position of particle *j*,
ρ_j is the density at the particle's position,
A_j is the scalar quantity for the particle, and
$W(\mathbf{r}, h)$ is the smoothing kernel with core radius *h*.

Smoothed Particle Hydrodynamics

The smoothing kernel here is a function something like a Gaussian, in that it should be large around zero and fall off with distance. Intuitively, this means that particles closer to a point contribute more. (Below, the bold lines are the kernels, the thin lines their gradients, and the dashed lines their Laplacians.)



For details on the choice of smoothing kernels for particular quantites, see the paper.

Smoothed Particle Hydrodynamics

One nice property of the SPH approach is that, when the gradient of a quantity is needed, it may be computed simply by using the gradient of the smoothing kernel.

$$\nabla A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h);$$

$$\nabla^2 A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h)$$

Smoothed Particle Hydrodynamics

Recall the quantities we wanted to compute: the pressure, the viscosity, and the density.

We will first consider the density, since it is used inside the computation of each other quantity. Subsituting $A_S = \rho_S$ into the SPH rule, we get

$$\rho_S(\mathbf{r}) = \sum_j m_j \frac{\rho_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h);$$

$$\rho_S(\mathbf{r}) = \sum_j m_j W(\mathbf{r} - \mathbf{r}_j, h)$$

The densities inside the sum cancel out, leaving us with a sort of smoothed sum of mass over the local area.

Smoothed Particle Hydrodynamics

For the pressure term, for a particle *i*, the SPH rule gives us

$$-\nabla p_S(\mathbf{r}_i) = -\sum_j m_j \frac{p_j}{\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h)$$

However, this is not symmetric between particles, which is a problem for us because it violates symmetry of forces. (The problem stems from the pressure at particle *i* being left out since the gradient of the kernel is zero at its center.) A simple way to make the forces symmetric is to average the local pressure with the pressure at each surrounding particle.

$$-\nabla p_S(\mathbf{r}_i) = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h)$$

Smoothed Particle Hydrodynamics

Of course, we still need to know how to compute each p_j.

$$-\nabla p_S(\mathbf{r}_i) = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h)$$

Given the density at the particle (from before), we can use the ideal gas state equation

$$p = k\rho$$

where k is a temperature-dependent constant.

(The paper actually uses the similar $p = k(\rho - \rho_0)$, where ρ_0 is a rest density, as this is equivalent for this computation and numerically more stable.)

Smoothed Particle Hydrodynamics

The computation of the viscosity term is similar:

$$\mu\nabla^2\mathbf{v}(\mathbf{r}_i) = \mu\sum_j m_j \frac{\mathbf{v}_j}{\rho_j}\nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h)$$

This is also asymmetric. We can use

$$\mu\nabla^2\mathbf{v}(\mathbf{r}_i) = \mu\sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j}\nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h)$$

to make it symmetric, since the viscosity depends only on the velocity difference between particles. As the paper notes, you can think of this as accelerating particle *i* in the direction of the relative speed of its environment.

Smoothed Particle Hydrodynamics

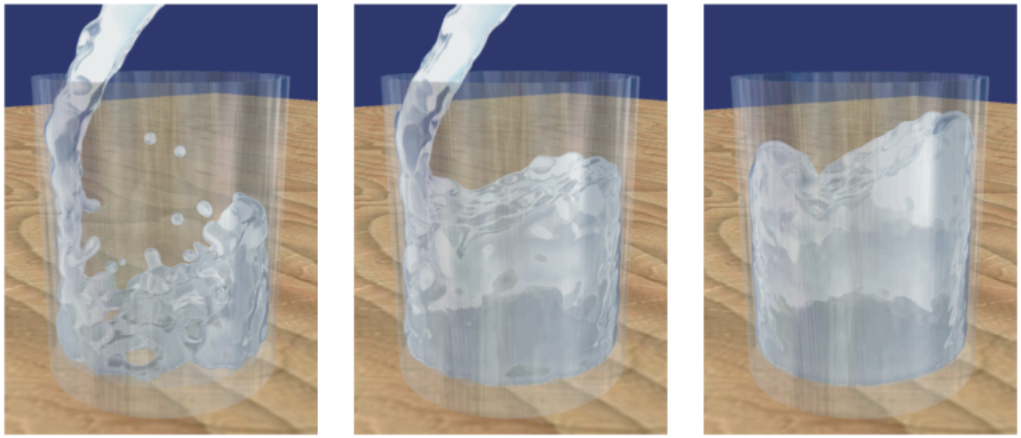
There are a couple of loose ends necessary for a full simulation.

Surface tension forces are modeled explicitly for particles near the surface using an additional SPH quantity. For details on the method used, see the paper.

Particles can *collide with their container*. When this happens, the particle is moved back into the body of the fluid and the component of the velocity normal to the container surface is negated.

Smoothed Particle Hydrodynamics

A principal advantage of this method is that it can run quickly. The authors, writing in 2003, report frame rates of 5 fps for their simulation of water pouring into a glass, using 3000 particles and marching cubes to render the results (shown here).



With a rougher rendering technique and only 1300 particles, they report achieving interactive frame rates of 25 fps.

Bubbling and Frothing Liquids

Cleary et al present a method for simulating bubbles in fizzy liquids, such as soda or beer, where gas is dissolved in the liquid. They consider the creation of bubbles, their transport through the liquid, the formation and flow of foam, and bubble dissipation.

The dissolved gas is modeled as an SPH quantity, and a bubble is formed at a nucleation site when enough gas has collected there. The bubbles then move as discrete particles, eventually gathering as foam.

Foam is also modeled as a set of discrete particles, but with different rules than for bubbles in the body of the liquid. Finally, bubbles in the foam may occasionally burst.

Bubbling and Frothing Liquids

As mentioned, the dissolved gas is modeled as an SPH quantity and tracked across the simulation. It is important to track this to provide realistic behavior; consider the difference between how much fizz is generated when a fresh drink hits a glass and later in the drink's life.

The rate of change of the dissolved gas for particle *i* is

$$\frac{dG_i}{dt} = D\sum_j \frac{2m_j}{\rho_i \rho_j} G_i \frac{\mathbf{r}_{ij} \cdot \nabla W(\mathbf{r}_{ij}, h)}{\mathbf{r}_{ij}^2 + \eta^2} - N_i - E_i$$

(Note that the equation here differs slightly from that in the paper. The subscripts have been changed to match the earlier discussion of SPH.)

Bubbling and Frothing Liquids

In

$$\frac{dG_i}{dt} = D\sum_j \frac{2m_j}{\rho_i \rho_j} G_i \frac{\mathbf{r}_{ij} \cdot \nabla W(\mathbf{r}_{ij}, h)}{\mathbf{r}_{ij}^2 + \eta^2} - N_i - E_i$$

D is the diffusivity of the gas in the fluid, a constant,
$\mathbf{r}_{ij}$ is ($\mathbf{r}_j$ - $\mathbf{r}_i$), the vector from one particle's position to the other, and
η is a small number added to smooth out the singularity when r_{ij} is zero.

The first term (the summation) models natural transport through diffusion of dissolved gas. The second and third terms model discrete losses due to creation of new bubbles ("nucleation") and expansion of existing bubbles.

Bubbling and Frothing Liquids

Bubbles are created when enough gas is present near a nucleation site. (How much is “enough” is a parameter to the simulation. A typical new bubble might be on the order of 0.5 mm in diameter.) When a bubble is created, a corresponding volume of gas is subtracted from the particle nearest to the nucleation site. (This is the N_i term in the rate-of-gas-change equation.)

Bubbles grow as they move through the liquid; a bubble removes some portion of the gas in nearby particles at each step. (This is the E_i term in the rate-of-gas-change equation.)

As we will see, larger bubbles are more buoyant and so rise faster. Therefore, this growth leads to a nice effect where bubbles accelerate and become more separated as they rise through the liquid.

Bubbling and Frothing Liquids

Each bubble is simulated as a discrete entity that may collide with other bubbles or the glass. SPH is not used to model the bubbles themselves, although SPH quantities from the surrounding liquid do come into play.

The forces that may act on a bubble in the liquid are *collision forces*, *buoyancy*, *drag* against the surrounding fluid, and (for bubbles above the surface) *cohesion* with surrounding bubbles.

The liquid may affect a bubble through buoyancy and drag, but we ignore any effect of the bubbles on the liquid's flow. (This is reasonable because the bubbles here are low-inertia compared to the liquid.)

Bubbling and Frothing Liquids

Collisions (both bubble-to-bubble and bubble-container) are modeled using a simple spring-damper system,

$$\mathbf{F}_n = -k\Delta x + Cv_n$$

where

k is the spring constant,
C is a damping constant,
Δx is the overlap between the two objects, and
v_n is the normal component of the relative velocity.

The normal is taken to be the vector between the two bubble's centers.

This force is only taken into account when Δx is positive (that is, when there is an active collision).

Bubbling and Frothing Liquids

Bubbles are much less dense than the liquid they displace, and so are buoyant. The buoyancy is given by

$$\mathbf{F}_b = \mathbf{g} V_{bub} (\rho_{bub} - \rho_{liq})$$

where

g is gravity,
V_{bub} is the volume of the bubble,
ρ_{bub} is the density of the bubble, and
ρ_{liq} is the density of the nearby liquid.

Bubbling and Frothing Liquids

A bubble's motion is impeded by a drag force from the surrounding liquid. In the simulation, this is intended to make the bubbles follow the flow patterns of the liquid.

The drag depends on the bubble's Reynolds number, a ratio of intertial forces to viscous forces.

$$\mathrm{Re}_{bub} = \frac{2\rho_{liq}\left\|\mathbf{v}_{bub} - \mathbf{v}_{liq}\right\| r_{bub}}{\mu_{liq}}$$

where

r_{bub} is the bubble's radius and
μ_{liq} is the liquid's viscosity.

Bubbling and Frothing Liquids

If Re_{bub} is less than 1.0, viscosity is high and Stokes' law applies:

$$\mathbf{F}_d = -6\pi r_{bub}\mu_{liq}(\mathbf{v}_{bub} - \mathbf{v}_{liq})$$

If Re_{bub} is greater than 1.0,

$$\mathbf{F}_d = -0.5C_d\rho_{bub}A_{bub}(\mathbf{v}_{bub} - \mathbf{v}_{liq})\left\|\mathbf{v}_{bub} - \mathbf{v}_{liq}\right\|$$

where

C_d is a constant drag coefficient, 0.2, and

A_{bub} is the surface area of the bubble.

Bubbling and Frothing Liquids

Foam that has risen above the surface of the main body of the liquid is called “dry foam”. Bubbles in dry foam are held together with a cohesive force.

This takes the same form as the collision force from before, but only applies when $\Delta x < 0$, that is, when bubbles are apart.

$$\mathbf{F}_{coh} = -k\Delta x + Cv_n$$

As with collisions, bubbles may also adhere to the walls of the container.

Bonds between dry bubbles are assigned a bond strength and may break apart to create foam flow and separate bodies of foam.

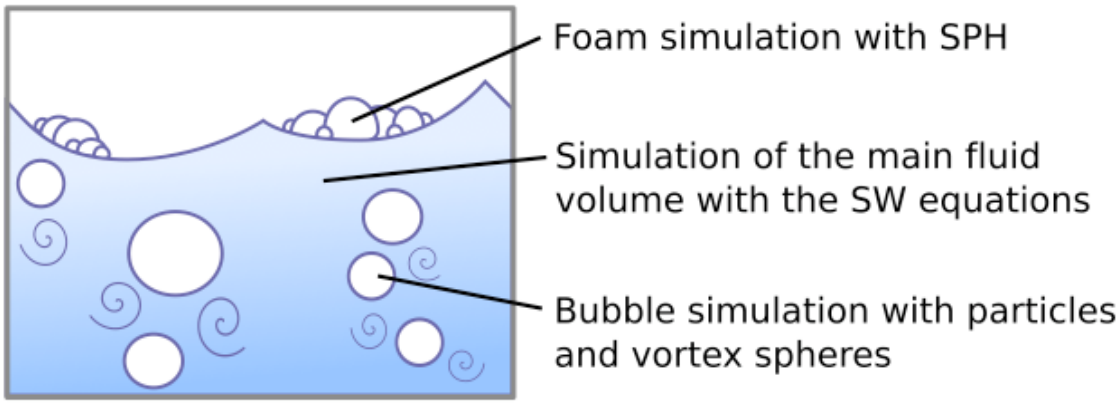
Bubbling and Frothing Liquids

Bubbles in dry foam may burst and disappear. This occurs probabilistically based on the bubble half-life, a parameter to the simulation.

Bubbles may also disappear or be formed due to coalescence or fragmentation during bubble collisions. These also occur probabilistically.

Shallow Water Framework

Thürey et al describe a technique for simulating bubbles in real time. Because a full SPH simulation for both bubbles and liquid might be computationally expensive, they combine a few different techniques.



Shallow Water Framework

The main body of the fluid is modeled as a 2D height field in the x-y plane. It is assumed that there is a constant pressure gradient from the top of the fluid to the bottom, and that the only forces present are pressure and gravity. (In particular, viscosity forces are ignored.)

This leads to the following simplified equations, which are then solved using a Stam-style solver.

$$\frac{\partial H}{dt} = -\mathbf{v} \cdot \nabla H - H\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right);$$

$$\frac{\partial v_x}{\partial t} = -\mathbf{v} \cdot \nabla v_x - g\frac{\partial H}{\partial x};$$

$$\frac{\partial v_y}{\partial t} = -\mathbf{v} \cdot \nabla v_y - g\frac{\partial H}{\partial y}$$

Shallow Water Framework

Bubbles in the fluid are simulated as discrete particles, each with a position, velocity, radius, and mass.

They are affected by *buoyancy* and by *velocity forces* arising from other elements in the simulation.

Shallow Water Framework

The buoyant force for a bubble *i* is computed as

$$\mathbf{f}_i^b = -\mathbf{g}\gamma_b m_i$$

where

g is gravity,

γ_b is a parameter denoting the density difference between the bubble and the fluid, and

m_i is the mass of bubble *i*.

Shallow Water Framework

The velocity force comes from two places: first, vortices created by nearby bubbles, and second, the velocity field of the shallow water simulation.

$$\mathbf{f}_j^v = \left[\mathbf{v}(\mathbf{p}_j) + \sum_i \mathbf{u}_h \left(\frac{\mathbf{d}_{ij}}{a_i}, \frac{|\mathbf{d}_{ij}|}{a_i} \right) f_v(|\mathbf{d}_{ij}|, a_i) \right] \frac{\Delta t}{m_j}$$

where

v is the velocity field of the shallow water simulation,

$\mathbf{u}_h$ is a function describing vorticity,

$\mathbf{d}_{ij}$ is a vector from the position of bubble j to the center of the vortex created by bubble i, and

a_i is the radius of the spherical vortex.

Formulations for u_h and a_i are given in the paper.

Note that here we are looking at the force for a primary bubble *j*, not a bubble *i* as in the discussion of buoyant force.

Shallow Water Framework

Submerged bubbles only affect the shallow water simulation by simple height displacement.

When a bubble breaks through the surface, it creates a small wave, modeled by adding to the height field of the shallow water simulation.

Shallow Water Framework

Surface (“dry”) foam is modeled using SPH. In this case, the bubbles themselves are the SPH particles. When a bubble breaks the surface, it adds a particle to the SPH simulation. (Strictly speaking, it might also burst; this is decided probabilistically.)

Bubbles in dry foam typically cluster, and this effect is achieved by adding surface tension forces to the SPH simulation.

Furthermore, a force based on gravity and the height field acts on each SPH particle. For a bubble at position **x**,

$$\mathbf{f}_s = \frac{\mathbf{g}}{|\mathbf{g}|}\left(H(\mathbf{x}) - (\mathbf{x}_z + \frac{r_i}{2}) \right) / \Delta t$$

These forces combine to create floating and spreading effects.

Shallow Water Framework

The authors, writing in 2007, report good frame rates for an interactive simulation. A table describing their results is reproduced below.

Table 1: *Simulation settings (SW simulation resolution, maximal number of bubble particles, maximal number of foam SPH particles) and performance measurements (lowest frame rate) for the different test cases.*

	SWS	Bubbles	Foam	FPS
Figure 7	40x66	81	133	161.2
Figure 8	50x50	142	131	144.1
Figure 9	50x80	501	466	34.3
Figure 10	50x80	654	1129	18.9