Harmonic Fluids

Changxi Zheng Doug L. James
Cornell University

Abstract

Fluid sounds, such as splashing and pouring, are ubiquitous and familiar but we lack physically based algorithms to synthesize them in computer animation or interactive virtual environments. We propose a practical method for automatic procedural synthesis of synchronized harmonic bubble-based sounds from 3D fluid animations. To avoid audio-rate time-stepping of compressible fluids, we acoustically augment existing incompressible fluid solvers with particle-based models for bubble creation, vibration, advection, and radiation. Sound radiation from harmonic fluid vibrations is modeled using a time-varying linear superposition of bubble oscillators. We weight each oscillator by its bubble-to-ear acoustic transfer function, which is modeled as a discrete Green's function of the Helmholtz equation. To solve potentially millions of 3D Helmholtz problems, we propose a fast dual-domain multipole boundary-integral solver, with cost linear in the complexity of the fluid domain's boundary. Enhancements are proposed for robust evaluation, noise elimination, acceleration, and parallelization. Examples are provided for water drops, pouring, babbling, and splashing phenomena, often with thousands of acoustic bubbles, and hundreds of thousands of transfer function solves.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; H.5.5 [Information Systems]: Info. Interfaces and Presentation—Sound and Music Computing

Keywords: Acoustic bubbles, sound synthesis, acoustic transfer

1 Introduction

Splash, splatter, babble, sploosh, drip, drop, bloop and ploop! Liquids are noisy and familiar sound sources. Yet, despite the enormous success of physically based fluid simulation in graphics, these simulations remain inherently silent movies. For most fluid applications, sound is an afterthought, added using stock recordings. While replaying "canned fluid sounds" is cheap and sometimes plausible, it can lack synchronization and physical consistency with observed dynamics, and may appear repetitive and perhaps irritating. Furthermore, while offline applications can rely on talented foley artists to "cook up" plausible sounds at their leisure, future interactive applications and virtual environments will demand algorithms for automatic procedural sound synthesis. Realistic physically based sound methods have appeared for vortex-based fluid sounds [Dobashi et al. 2003] and solid bodies [O'Brien et al. 2001; James et al. 2006], but we still do not know how to simulate synchronized physics-based sounds for familiar splashes and splatters.

What causes fluid sounds? Perhaps surprisingly, the majority of sound from a splashing droplet of water arises from harmonic vibrations resulting from the entrainment (creation) of millimeter-scale air bubbles (see Figure 1). Basically, the bubble oscillator stores potential energy as compressed air and surface tension, and kinetic energy as surrounding fluid vibrations. The important role of these tiny "acoustic bubbles" in water sound generation has been recognized for nearly a century since pioneering work by Minnaert [1933], and large texts have since been written about them [Leighton 1994]. Recently, van den Doel [2005] proposed bubbles as primitives for fluid sound synthesis, and synthesized compelling sounds using stochastically excited modal sound banks.

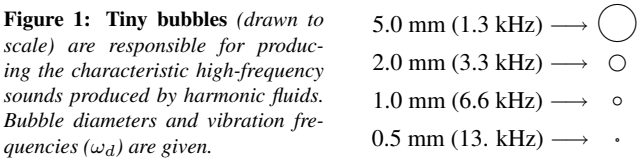


Figure 1: Tiny bubbles *(drawn to scale) are responsible for producing the characteristic high-frequency sounds produced by harmonic fluids. Bubble diameters and vibration frequencies (ω_d) are given.*

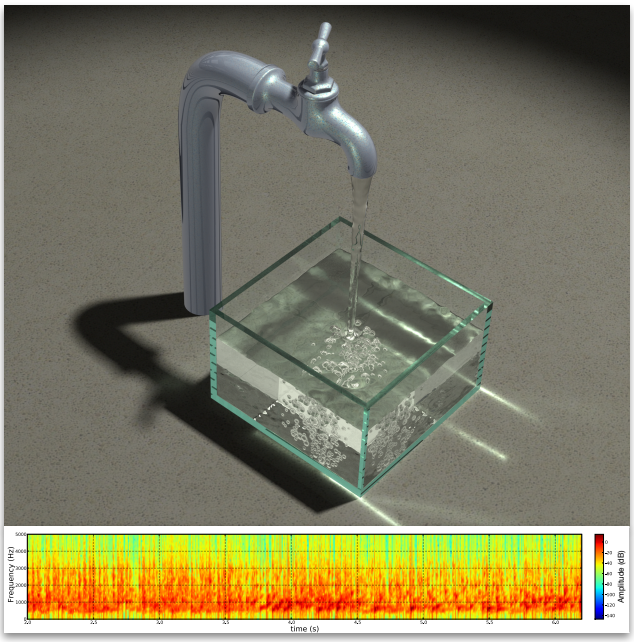


Figure 2: Synthesizing the sound of pouring water *via the linear superposition of acoustic radiation from 7900 vibrating acoustic bubbles.*

Exploiting multiple timescales: Ironically, the complex visible motion of the air-fluid interface causes relatively little sound, in part because visible surface motions are inefficient radiators of sound waves at audible frequencies [Bragg 1920]. Instead, the fluid shape vibrates harmonically at audio frequencies due to the microscopic oscillations induced by internal air bubbles, and acts like a shape-changing 3D loudspeaker. For example, consider visible fluid movements occurring at graphics rates: a water splash on a $15cm$-sized domain might occur over a 10^{-1} second timescale, i.e., a few frames, whereas enormous water sound speeds ($c_{water} \approx 1450m/s$) allow water sound waves to cross the $15cm$ domain in only 10^{-4} seconds. This thousand-fold difference in animation and sound wave timescales is why sound waves can propagate through small fluid bodies almost as if they were standing still. Therefore, we choose to model sound wave propagation and radiation in fluids by assuming they are a sequence of static problems. Given the harmonic nature of bubbles, we can efficiently model sound waves in the frequency-domain using the Helmholtz wave equation.

Our approach: We propose the first practical physically based method for synthesizing synchronized harmonic fluid sounds for computer animation (see Figure 2 for a preview). We model the creation of bubbles by air entrainment at the fluid surface; the advection of these bubbles with the fluid flow; the surface vibrations induced by the bubbles' vibrations; and the radiation of these vibrations into the air, producing sound (see Figure 3). Our method augments an existing incompressible fluid flow solver with a particle-based acoustic bubble model that models bubble entrainment, advection, vibration, and radiation. By avoiding audio-rate time-stepping of 3D compressible fluid sound waves (which are expensive, and difficult to parallelize), we can extend existing graphics fluid simulators with a pleasantly parallel sound model.

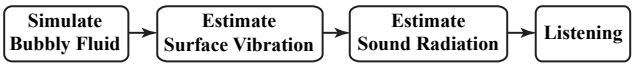


Figure 3: Overview: *(1) We first simulate an incompressible fluid flow with bubbles. For each vibrating bubble we (2) estimate the induced fluid-air surface vibration and (3) resulting air-domain sound pressure. (4) Finally, the linear superposition of bubble sound fields are rendered to the listener.*

Our main contribution is a parallel algorithm for estimating sound radiation. Spherical bubble vibrations induce harmonic vibrations of the fluid-air interface, which leads to acoustic radiation[1] which we approximate by a time-varying linear superposition of harmonic bubble contributions. The amplitude of each bubble oscillation is effectively multiplied by the bubble-to-ear acoustic transfer function, which we model in the frequency domain as the bubble-located Green's function of the Helmholtz wave equation for the instantaneous fluid geometry. These transfer functions can exhibit complex hundredfold variations which we believe are key to capturing the tonal character of harmonic fluids (see Figure 4). Enabling inexpensive Helmholtz Green's function evaluations is achieved by a novel dual-domain multipole approximation based on a two-stage fast linear-time boundary-integral solver. In the first stage, we solve a fluid-domain problem to estimate the normal velocity of the vibrating air-fluid interface. In the second stage, we estimate a multipole approximation of the air-domain acoustic radiation for sound rendering. Key benefits are that the transfer functions need only be updated at fluid simulation rates (or slower), and the only audio rate calculation required is the cheap integration of nonlinear bubble vibrations—thus subsequent sound synthesis can be achieved potentially in real time. We demonstrate harmonic fluid animations involving thousands of acoustic bubble sound sources, with parallelized sound computation times comparable to fluid simulation.

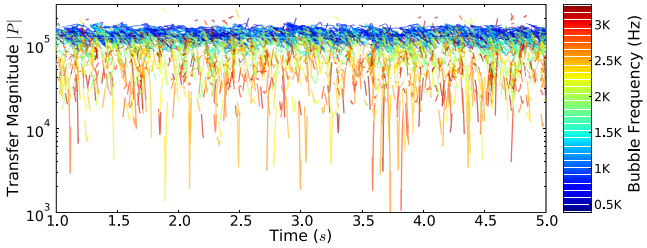


Figure 4: Observed transfer magnitudes $|P|$ *illustrate the complex bubble-dependent temporal structure, and significant hundred-fold variations in magnitude. Frequency colors illustrate that transfer magnitude is not just a function of frequency, but rather has other complex spatial and temporal dependencies. (Data: "water step" example.)*

[1] A graphics analogy: a point light source of specific frequency (the acoustic bubble) radiates light out of a tiny air-filled void (the water) into a highly refractive solid (the air) where it is observed (by the listener). Interestingly, because of the large difference in the speed of sound in water (1497m/s) and air (343m/s), the effective index of refraction is $\eta = 4.4$!

Other Related Work: Fluid sounds can arise via numerous mechanisms [Blake 1986] including harmonic bubble-based fluid sounds [Leighton 1994], vortex based sounds (e.g., whistling) [Howe 2002], shock waves (e.g., from explosions), and through fluid-solid coupling with vibrating solids [Howe 1998]. Perhaps the least familiar but most common, harmonic bubble-based fluid sounds come with almost all kinds of fluid movement: splashing or pouring water [Franz 1959], rain drops [Longuet-Higgins 1990], babbling brooks [Minnaert 1933], etc. Bubbles have received enormous attention due to vibration-based sound radiation and other exotic behaviors, such as cavitation (which can pit propellors) and even their ability to give off light via sonoluminescence! Bubble-related sounds have been studied for about a century, and people understand, albeit not entirely, the basic mechanisms for sound emission. It was realized nearly a century ago that it is hard for water to make any sound by itself [Bragg 1920], and Minnaert [1933] described the important role of harmonic acoustic bubbles. Continued work has revealed that most of the sound arises not from the initial impact of fluid but from small bubbles entrained from the resulting surface cavity [Pumphery et al. 1989; Longuet-Higgins 1990; Oguz and Prosperetti 1990]. The acoustics community has studied acoustic bubbles extensively because of their wide importance, e.g., in computational ocean acoustics [Jensen 1994], in estimating rainfall rates for climate models [Urick 1975], and understanding sounds from complex bubble plumes in breaking waves and surf [Deane 1997]. However, we still lack practical algorithms for synthesizing harmonic fluid sounds.

On the other hand, the computer graphics community has developed a sophisticated array of computational methods for simulating fluids in computer animation [Foster and Metaxas 1996; Stam 1999; Enright et al. 2002; Osher and Fedkiw 2003]. Because of their visual importance, numerous methods for animating bubbles and foam have appeared [Foster and Metaxas 1996; Greenwood and House 2004; Zheng et al. 2006; Cleary et al. 2007; Thuerey et al. 2007; Kim et al. 2007; Kim and Carlson 2007; Hong et al. 2008]. However, no methods currently address fluid sound generation.

Realistic sound rendering in computer graphics has addressed auralization of sound sources in virtual environments [Begault 1994; Kleiner et al. 1993; Vorlander 2007] especially for interactive virtual environments [Funkhouser et al. 1999; Tsingos et al. 2001; Tsingos et al. 2004], however, less work has addressed the physically based modeling of realistic sound sources. Important exceptions include vibrations of otherwise rigid objects, often using harmonic modal vibration methods [van den Doel and Pai 1996; van den Doel et al. 2001; O'Brien et al. 2002; Bonneel et al. 2008]. While fluid vibrations are locally harmonic, they exhibit long-time nonlinearities, and the fluid surface vibrations are themselves unknowns. We propose a method to estimate these vibrations, after which existing frequency-domain radiation solvers based on equivalent multipole sources, such as Precomputed Acoustic Transfer [James et al. 2006], could in principle be used. However, in practice significant complications arise due to temporal coherence, etc., and we instead propose an all-at-once dual-domain solver. For more general surface vibrations in computer animation, ray-based Rayleigh sound approximations (which ignore diffraction effects) have been used [O'Brien et al. 2001]. However, incompressible fluid solvers are incapable of producing predominantly harmonic sounds without the help of bubbles [Bragg 1920], and therefore applying ray-based renderers to existing fluid animations (at audio rates) will not produce correct fluid splashing sounds [Franz 1959].

To provide realism at minimal cost, recorded fluid sounds are often used to produce realistic sounds, e.g., of splashes and drops, by event-based sound synthesizers in interactive environments [Takala and Hahn 1992], and in offline animations by foley artists (c.f. [Carlson et al. 2004]). Sounds have also been used to sonify

CFD and other data for multi-modal data exploration and visualization purposes [McCabe and Rangwalla 1994; Childs 2001]. Recently Imura et al. [2007] proposed an *ad hoc* bubble-based fluid sound method that augmented an SPH fluid simulation with data-driven bubble sounds based on recordings of individual bubbles. Unfortunately such methods are not physically based, and can not capture time-varying spatial structure of 3D sound radiation.

Stochastic sound models and granular synthesis are often used to produce noise-like fluid sounds such as waterfalls, rainfall, and ocean waves [Cook 2002]. Continuous stochastic models can be constructed from input sound files, and can synthesize new sound textures, e.g., of babbling brooks [Miner and Caudell 2005]. Van den Doel proposed using harmonic bubble oscillator sound banks to produce stochastic bubble-based fluid sounds [van den Doel 2005]. Again, such methods are not physically based and/or lack integration with 3D fluid simulation and sound radiation models.

Finally, vortex-based fluid sounds, e.g., of whistling, have been synthesized by Dobashi and colleagues [2003; 2004] by a clever combination of Lighthill's theory of vortex-based sound [Howe 2002] and data-driven techniques. Since both vortex and harmonic fluid sounds can be present, our techniques are complementary.

2 Background: Incompressible Fluid Solver

Our acoustic bubble simulation is designed to augment existing incompressible liquid solvers familiar to the graphics community [Foster and Metaxas 1996; Stam 1999; Foster and Fedkiw 2001; Enright et al. 2002]. In this paper, we employ the Euler equations governing inviscid flow [Osher and Fedkiw 2003],

$$0 = \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho} \nabla p \qquad \text{subject to} \qquad 0 = \nabla \cdot \boldsymbol{u}, \tag{1}$$

which relate the liquid's velocity ($\boldsymbol{u}$), pressure (p) and density (ρ). Our approach does not depend critically on any particular fluid simulation method. However, in our implementation we use the FLIP/PIC method [Zhu and Bridson 2005], since its fluid particles are convenient markers to track bubble creation. We compute the level set function, $\phi(\boldsymbol{x})$ (negative in fluid, and positive in air), using the method proposed in Adams et al. [2007], with redistancing performed at each time step using a fast marching method [Osher and Fedkiw 2003]. We update the particle-based bubble simulation after each fluid time step using a one-way coupling approximation.

3 Modeling Acoustic Bubbles

Acoustic bubbles have received significant attention, and we refer the reader to the text by Leighton [1994] for a comprehensive introduction. Unfortunately their integration into 3D fluid simulators leads to a number of modeling details which need to be addressed (see Figure 5). We now summarize the acoustic bubble model used to implement Harmonic Fluids.

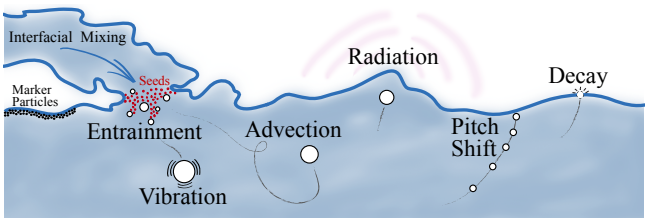


Figure 5: Life of an acoustic bubble

3.1 The Spherical Acoustic Bubble

Bubbles that generate audible sounds are typically quite small (≈ 1 mm), and in that limit, surface tension forces are strong enough to make the bubble essentially spherical. The spherical air bubble is an excellent oscillator, pulsating after an initial entrainment-related impulse. The simplest linear vibration model assumes an ideal, spherically pulsating, mono-frequency bubble, and was proposed originally by Minnaert [1933]. It models a spring-bob system where the restoring "spring" force is due to air pressure and surface tension, and inertia is due to the effective mass of the surrounding liquid. Consider a pulsating spherical bubble with radius

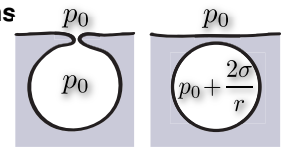
$$r(t) = r_0 + q(t), \tag{2}$$

where r_0 is the static radius (which may change slowly), and $q = q(t)$ is a small fluctuation ($|q| \ll r_0$) due to rapid spherical pulsations. An established linear model of the harmonic pulsations is the simple harmonic oscillator [Leighton 1994]:

$$\ddot{q} + 2\beta\dot{q} + \omega_0^2 q = F_b / m_b^{rad}, \tag{3}$$

where ω_0 is the bubble's resonant frequency (in radians/sec); β is the damping rate; F_b is the external forcing due to liquid pressure fluctuations and entrainment; and $m_b^{rad} = 4\pi r_0^3 \rho$ is the bubble's *effective radiative mass*. In practice, we hear the damped natural frequency $\omega_d = \sqrt{\omega_0^2 - \beta^2}$; sample values were given in Figure 1. Formulae for ω_0 and $\beta = \beta(\omega_0, r_0)$ are provided in Appendix A.

3.2 Exciting Bubble Vibrations



At the moment of bubble entrainment, the fluid-trapped air is subjected to an additional pressure: pressure jumps from just air pressure, p_0, to one also involving surface tension, $p_0 + p_\sigma$, where $p_\sigma = \frac{2\sigma}{r_0}$ is the extra surface tension ("Laplace") pressure [Leighton 1994]. For tiny acoustic bubbles, the surface tension pressure jump can be enormous. We model bubble vibration forcing using an initial pressure-jump impulse, and ignore later forcing. This corresponds to forcing the bubble vibration equation (3) with the right-hand side given by $\frac{2\sigma}{r_0 m_b^{rad}} \delta(t)$ (for $t = 0$ entrainment), and would yield the oscillator response, $q(t) = \frac{2\sigma}{r_0 m_b^{rad} \omega_d} e^{-\beta t} \sin \omega_d t$. Since the frequency and damping coefficients of (3) are time dependent, in practice we integrate the vibrations numerically using the midpoint method. To soften the attack (c.f. [van den Doel et al. 2001]), we smoothly blend the sound in over a $\Delta t = \frac{C}{\beta}$ window. We use $C = \ln(0.85)$ to blend until the amplitude decays to 0.85 of its initial amplitude; our blending function is given by (39) in Appendix A. By tuning these parameters via comparisons to recordings, our vibration responses appear plausible (see Figure 6).

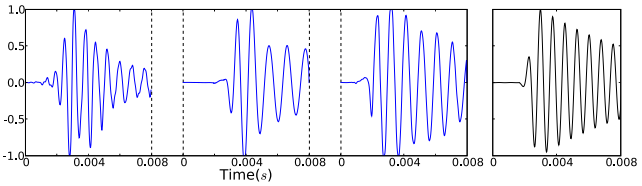


Figure 6: Comparison of bubble excitations: *(Left) Three recorded bubble sounds (in blue) illustrating typical bubble excitation responses and variations; and (Right) the response of our bubble model. Recordings were obtained from individual droplets falling 0.5m from a faucet (at ≈ 2 droplets/sec) into a water-filled container (roughly $30cm \times 30cm \times 15cm$).*

3.3 Particle-based Bubble Advection

Identical to previous works, we model bubbles as buoyant particles advected in the incompressible flow [Greenwood and House 2004; Cleary et al. 2007]. Each tiny bubble is advected independently, ignoring complex bubble-bubble interactions. We model the bubble

motion as a particle of effective mass $m_b = \frac{4}{3}\pi r_0^3 \rho$ (of the liquid hole), with applied pressure, gravity and drag forces,

$$\boldsymbol{f}_p = -K_p V_b \nabla p_i + m_b \boldsymbol{g} \quad (4)$$

$$\boldsymbol{f}_d = \frac{1}{2} C_d \rho A_b (\boldsymbol{u} - \boldsymbol{v}_b) \|\boldsymbol{u} - \boldsymbol{v}_b\| \quad (5)$$

where μ_f is fluid viscosity, $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}_b)$ is the fluid velocity at bubble's location, $\boldsymbol{x}_b$, the drag coefficient is $C_d = 0.2$, the bubble surface area is $A_b = 4\pi r_0^2$, its volume is $V_b = \frac{4}{3}\pi r_0^3$, and $K_p = 0.8$ in our examples. The drag force model is suitable for tiny acoustic bubbles which have Reynolds number, $Re = 2\rho\|\boldsymbol{u} - \boldsymbol{v}_b\| r_0 / \mu_f \gg 1$. After each fluid time step, we integrate the particle's motion using the mid-point method.

3.4 Time-dependent Bubble Frequency

The simple acoustic bubble model (3) uses a fixed frequency ω_0 (and damping β). However, a perceptually important feature of moving bubbles is that their frequency can vary significantly over time, with bubble sounds often having a characteristic rising pitch, e.g., the familiar "bloooop?" of a water drop (see Figure 7). Phenomenologically speaking, as the bubble approaches the fluid surface, the effective vibrational mass m_b^{rad} of the surrounding fluid decreases (since there is less of it to move), whereas the stiffness k_b (due to surface tension and air compression) is relatively unchanged, thus producing an increase in the resonant frequency, $\omega_0^2 = k_b / m_b^{rad}$. Models exist for rising bubble pitch as a function of distance to a planar liquid surface [Strasberg 1953], but can only provide about a $\sqrt{2}$ frequency multiplier—in our experiments, we often observed 2–3 times frequency multipliers likely due to complex local fluid geometry (see Figure 7).

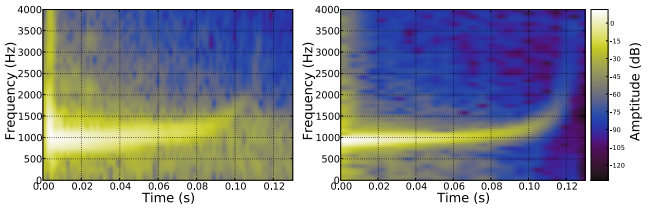


Figure 7: Water drop spectra: *(Left) A recording of a single-bubble water drop experiment; (Right) the spectra of a single-bubble fluid sound synthesized from a digital mockup. Each water droplet fell from a height of* ≈ 0.5 *meters into a pool of water. Both spectra exhibit qualitatively similar structures, with clear evidence of rising pitch.*

To support nonplanar interface geometry and larger pitch shifts[2] we propose an *ad hoc* model based on the bubble's level set values, ϕ. Our frequency-change model captures two behaviors: (1) the bubble frequency does not change until it approaches the surface, and (2) the faster the bubble travels to the surface, the faster its pitch changes. At each sound synthesis time step we increment the natural frequency, ω_0, by

$$\Delta\omega = K_\Delta \, \omega_d \, e^{-\eta\left(\frac{\phi}{\phi_0} - 1\right)} \, \Delta\phi, \quad (6)$$

where $\Delta\phi$ is the distance change since the last timestep, $\phi = \phi(t)$ is the (negative) distance to the fluid surface (from the fluid simulator); ϕ_0 is a distance parameter controlling how close to the interface the bubble must be to undergo pitch shift, and η controls the spatial rapidity of change; and K_Δ controls the magnitude of frequency changes. Following ω_0 modification, we update dependent parameters (such as ω_d and β). In our results, we always use $K_\Delta = 72.95$ and $\eta = 1.24$, but adjust ϕ_0 (between -0.008 and -0.025 meters). Our parameters (K_Δ and η) were tuned manually by performing numerous comparisons to real-world experiments. Qualitatively similar results can be obtained (see Figure 7). Finally, since ϕ is only evaluated at fluid time-stepping rates, for sound synthesis we temporally interpolate ϕ values to audio rates using a cubic spline; a low-pass filter is also used to remove temporal noise artifacts introduced by the fluid discretization.

[2] and to avoid estimating the bubble frequency as an eigenvalue of a fluid-bubble interaction problem [Ohayon 2004]

3.5 Modeling Acoustic Bubble Entrainment

To compute plausible fluid sounds, the entrainment of bubbles by estimated fluid-air mixing must be done so as to produce bubbles with appropriate distributions of radii (frequencies), amplitudes, and spatial and temporal structure. Once a bubble is created and an initial impulse applied, it can be simulated and sonified. Unfortunately, the bubble entrainment process is terribly complex [Leighton 1994] and computationally difficult to resolve spatially and temporally. Therefore we propose a simplified model of the acoustic bubble creation process. Similar to prior work [Greenwood and House 2004], we use marker particles to track where bubbles should be created, and our spherical acoustic bubbles are driven by one-way coupling to the fluid simulator. Primary differences in our work ("bubble seeds," bubble creation rates, and modeling of radii and spectra) result from our attempts to make the bubbles sound more plausible. We defer the interested reader to Appendix B for the details of our acoustic bubble entrainment model, and now proceed with sound radiation modeling.

4 Modeling Fluid Sounds

Sound radiation is modeled as a superposition of individual bubble sounds. For each harmonic bubble, we first estimate the induced fluid-air interface vibration, then next estimate the radiated air-domain sound waves that travel to the listener. Our multiple timescale approximation models these waves in the frequency domain. We now describe how to estimate the time-harmonic acoustic pressure field, $P(\boldsymbol{x}, t) = P(\boldsymbol{x}) e^{+i\omega t}$, where $P(\boldsymbol{x})$ is the (slowly time varying) spatial part satisfying the Helmholtz equation, and ω is the frequency of an acoustic bubble.

Listening to Helmholtz Green's functions: Given a bubble of frequency ω at position $\boldsymbol{x}_b$ in the fluid domain Ω_f, we use the harmonic Green's function $P(\boldsymbol{x}; \boldsymbol{x}_b) \in \mathbb{C}$ of the Helmholtz wave equation on the unbounded region comprised of both fluid and air domains, $\Omega = \Omega_f \bigcup \Omega_a$:

$$\left(\nabla^2 + k^2(\boldsymbol{x})\right) P(\boldsymbol{x}; \boldsymbol{x}_b) = S_b \, \delta(\boldsymbol{x} - \boldsymbol{x}_b), \quad \boldsymbol{x} \in \Omega, \quad (7)$$

where the spatially varying wavenumber is

$$k(\boldsymbol{x}) = \begin{cases} \omega / c_f, & \boldsymbol{x} \in \Omega_f \\ \omega / c_a, & \boldsymbol{x} \in \Omega_a \end{cases} \quad (8)$$

and c is the speed of sound (see Table 3). The Green's function is subject to an homogeneous Neumann boundary condition on the solid interface,

$$\partial_n P \equiv \frac{\partial P}{\partial n} = 0, \quad \boldsymbol{x} \in \Gamma_s \quad (9)$$

which corresponds to a "no vibration" boundary condition of zero surface normal velocity, $v_n = \mathbf{n} \cdot \mathbf{v}(\mathbf{x})$, since $\partial_n P \equiv -i\omega\rho \, v_n$; and the Sommerfeld radiation condition at infinity [Howe 1998]. The bubble's source strength S_b (for unit vibration amplitude) is

$$S_b = -4\pi\rho \, \omega^2 r_0^2 \quad (10)$$

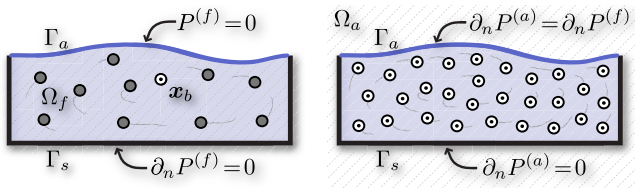
(see derivation in Appendix C). We will often refer to P as the bubble-to-ear acoustic transfer function, or simply "transfer." With this definition, we could approximate the sound contribution at the listening position, $\boldsymbol{x} \in \Omega_a$, due to a single bubble via

$$|P(\boldsymbol{x}; \boldsymbol{x}_b)| \, q(t), \quad \boldsymbol{x} \in \Omega_a \quad (11)$$

(or more sophisticated auralizations (see §6)). Unfortunately, in practice we desire to solve the Helmholtz PDE (7) *for every bubble in the scene at each fluid time step*. To make matters worse, since Ω is the unbounded region, efficient computation and evaluation of this function (for audio rendering) is a practical concern.

5 Dual-domain Multipole Radiation Solver

We now describe a novel Helmholtz boundary integral solver for rapid evaluation of the acoustic pressure $P(\boldsymbol{x};\boldsymbol{x}_b)$ to enable sound synthesis from harmonic fluids. We use an efficient two-stage approximation to any bubble's Helmholtz Green's function that exploits the common case wherein fluid vibrations are affected by the surrounding air only weakly. The solver is summarized in Figure 8. Readers wishing to skip this section's heavier mathematical details can proceed to §6 "Sound Synthesis Pipeline."



Fluid pressure $P^{(f)}$ problem Air pressure $P^{(a)}$ problem

Figure 8: Overview of dual-domain Helmholtz formulation: *(Left) Pass #1 solves the fluid-domain problem to estimate the fluid's acoustic pressure $P^{(f)}$ assuming that fluid on the solid boundary Γ_s can not vibrate ($\partial_n P^{(f)}=0$), and that fluid at the air interface is free to move ($P^{(f)}=0$). In addition to the singular bubble source at $\boldsymbol{x}_b$, numerous advected regular sources also contribute to $P^{(f)}$; their expansion coefficients $\mathbf{c}_f$ are least-squares estimated to match the boundary conditions. (Right) Pass #2 solves the air-domain problem to estimate the air's acoustic pressure $P^{(a)}$ assuming that air on the solid boundary Γ_s can not vibrate ($\partial_n P^{(a)}=0$), but that vibrations on the air interface Γ_a match the computed fluid vibrations ($\partial_n P^{(a)}=\partial_n P^{(f)}$). The air pressure $P^{(a)}$ is described by a larger number of advected singular multipole sources, whose expansion coefficients $\mathbf{c}_a$ are least-squares estimated to match the Neumann boundary conditions.*

5.1 Dual-domain Helmholtz Approximation

Let us now consider how to break the computation of the fluid-air Helmholtz Green's function $G(\boldsymbol{x};\boldsymbol{x}_b)$ into two Helmholtz problems with one-way coupling.

Pass #1: First, we compute a fluid-domain Green's function, $P^{(f)}=P^{(f)}(\boldsymbol{x};\boldsymbol{x}_b)$,

$$(\nabla^2+k_f^2)\,P^{(f)}=S_b\delta(\boldsymbol{x}-\boldsymbol{x}_b),\quad \boldsymbol{x}\in\Omega_f, \tag{12}$$

subject to the homogeneous boundary conditions on the fluid-air boundary, Γ_a, and the solid-air boundary, Γ_s,

$$P^{(f)}=0, \qquad \boldsymbol{x}\in\Gamma_a, \tag{13}$$

$$\partial_n P^{(f)}=0, \qquad \boldsymbol{x}\in\Gamma_s. \tag{14}$$

Pass #2: Second, given an approximation of the fluid-domain Green's function pressure, $P^{(f)}$, we can evaluate its normal derivative on the fluid-air interface (which specifies the surface normal velocity, v_n), and use that as an input to estimate the radiation into the surrounding air. The resulting air-domain Green's function, $P^{(a)}(\boldsymbol{x};\boldsymbol{x}_b)$ satisfies the unforced Helmholtz equation in air,

$$(\nabla^2+k_a^2)\,P^{(a)}(\boldsymbol{x};\boldsymbol{x}_b)=0,\quad \boldsymbol{x}\in\Omega_a, \tag{15}$$

subject to the Sommerfeld radiation condition at infinity, and

$$\partial_n P^{(a)}=\partial_n P^{(f)}, \qquad \boldsymbol{x}\in\Gamma_a, \tag{16}$$

$$\partial_n P^{(a)}=0, \qquad \boldsymbol{x}\in\Gamma_s, \tag{17}$$

which are derivative (velocity) boundary conditions, with the all important nonzero values on the vibrating fluid interface, Γ_a, and just zero values on any supporting rigid interface, Γ_s. Finally, our $P^{(a)}$ model is used to evaluate $P(\boldsymbol{x};\boldsymbol{x}_b)$ in Ω_a for sound rendering.

5.2 Pass #1: Interior Fluid-domain Solver

We approximate the Helmholtz problems using Trefftz-style equivalent source methods [Kita and Kamiya 1995; Ochmann 1995; James et al. 2006]. In each pass, the domain PDE is satisfied using a series expansion of fundamental solutions to the Helmholtz equation, and the boundary conditions are approximated in a least-squares sense to estimate expansion coefficients.

Pressure Expansion: To satisfy (12) for the fluid-domain Helmholtz Green's function, we introduce the pressure expansion

$$P^{(f)}(\boldsymbol{x};\boldsymbol{x}_b)=s(\boldsymbol{x};\boldsymbol{x}_b)+\mathbf{U}^{(f)}(\boldsymbol{x})\,\mathbf{c}_f, \tag{18}$$

where s is the singular free-space Helmholtz Green's function,

$$s(\boldsymbol{x};\boldsymbol{x}_b)=-\frac{e^{-ik_fR}}{4\pi R}S_b, \qquad (R=\|\boldsymbol{x}-\boldsymbol{x}_b\|) \tag{19}$$

satisfying the fluid Helmholtz equation

$$(\nabla^2+k_f^2)\,s(\boldsymbol{x};\boldsymbol{x}_b)=S_b\delta(\boldsymbol{x}-\boldsymbol{x}_b), \tag{20}$$

and the Sommerfeld radiation condition; the second part of (18) is a weighted combination of n_f nonsingular functions,

$$\mathbf{U}^{(f)}(\boldsymbol{x})\,\mathbf{c}_f=\left[\psi_1^{(f)}\;\psi_2^{(f)}\;\ldots\;\psi_{n_f}^{(f)}\right]\mathbf{c}_f \tag{21}$$

where $\mathbf{c}_f\in\mathbb{C}^{n_f}$ are weights, and $\mathbf{U}^{(f)}$ is a row matrix of functions, $\psi^{(f)}$, each satisfying the fluid Helmholtz equation (without regard for Γ boundary conditions)

$$(\nabla^2+k_f^2)\,\psi_j^{(f)}(\boldsymbol{x})=0,\quad \boldsymbol{x}\in\Omega_f. \tag{22}$$

Since the $P^{(f)}$ expansion (18) satisfies the Green's function PDE in (12), it remains only to select a sufficiently complete basis, $\mathbf{U}^{(f)}$, then find coefficients, $\mathbf{c}_f$, to satisfy the homogeneous boundary conditions (13-14). We propose using the *regular* spherical Helmholtz solutions [Gumerov and Duraiswami 2005] (other choices are possible),

$$\psi_j^{(f)}(\boldsymbol{x};\boldsymbol{x}_j^{(f)})=j_\ell(k_fR)\;Y_\ell^m(\theta,\phi), \tag{23}$$

where $\psi_j^{(f)}$ is positioned at $\boldsymbol{x}_j^{(f)}$ (we describe point-source selection later in §5.4), $R=\|\boldsymbol{x}-\boldsymbol{x}_j^{(f)}\|_2$, $Y_\ell^m\in\mathbb{C}$ are the spherical harmonics, and $j_\ell(k_fR)$ are spherical Bessel functions of the 1^{st} kind, e.g., $j_0(z)=\frac{\sin z}{z}$, $j_1(z)=\frac{\sin z}{z^2}-\frac{\cos z}{z}$, $j_2(z)=(\frac{3}{z^2}-1)\frac{\sin z}{z}-\frac{3\cos z}{z^2}$. In our implementation, we use basis functions up to and including quadrupoles ($\ell=0,1,2$), so our n-point multipole expansions have $n_f=9n$ unknown complex-valued coefficients.

Collocated Least-Squares Estimation: Given the homogeneous boundary conditions on $P^{(f)}$ (13-14), we collocate the boundary condition equations at N boundary points to obtain N equations involving the $\mathbf{c}_f\in\mathbb{C}^{n_f}$ unknowns, then estimate $\mathbf{c}_f$ using weighted least squares. Collocation points are chosen as mesh vertices (discussed in §5.5), and each sample point, $\boldsymbol{x}_i$, has normal, $\boldsymbol{n}_i$, and an effective area, Δa_i. The relevant equations for vertex point i are

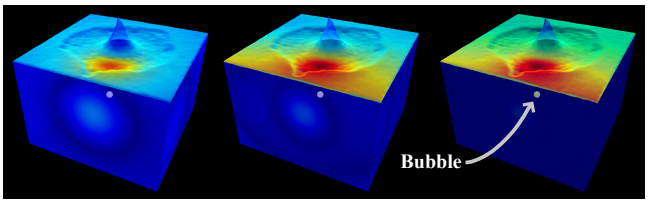
$$\mathbf{U}^{(f)}(\boldsymbol{x}_i)\,\mathbf{c}_f=-s(\boldsymbol{x}_i), \qquad \text{when } \boldsymbol{x}_i\in\Gamma_a \tag{24}$$

$$\partial_{\boldsymbol{n}_i}\mathbf{U}^{(f)}(\boldsymbol{x}_i)\,\mathbf{c}_f=-\partial_{\boldsymbol{n}_i}s(\boldsymbol{x}_i), \qquad \text{when } \boldsymbol{x}_i\in\Gamma_s \tag{25}$$

for $i=1\ldots N$. Each equation is weighted by $\sqrt{\Delta a_i}$, to assemble the N-by-n_f linear least-squares problem,

$$\mathbf{A}\,\mathbf{c}_f=\mathbf{b} \quad\Leftrightarrow\quad \begin{bmatrix}\mathbf{A}_a\\ \alpha\mathbf{A}_s\end{bmatrix}\mathbf{c}_f=\begin{bmatrix}\mathbf{b}_a\\ \alpha\mathbf{b}_s\end{bmatrix}. \tag{26}$$

The relative scaling parameter, α, balances the importance of pressure versus pressure derivative constraints; in our examples (with approximately unit-sized computational domains), we use the ratio of interfacial area, $\alpha=\text{Area}_s/\text{Area}_a$. After robust construction and least-squares solution of (26) for $\mathbf{c}_f\in\mathbb{C}^{n_f}$ (discussed in §5.6), we can estimate the harmonic fluid-surface vibrations (see Figure 9).



5 quadrupoles 10 quadrupoles 20 quadrupoles

Figure 9: Estimated surface velocity ($v_n \propto \partial_n P^{(f)}$) *computed from the fluid-domain solver (pass #1). Approximations are shown for differing numbers of regular quadrupole sources, and degrees of convergence.*

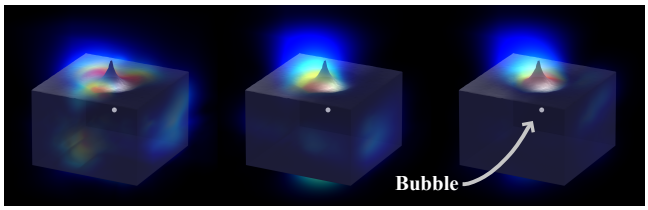
Discussion: The boundary integral equation associated with the least-squares problem (26) is related to their normal equations, and can be written as

$$\left[\int_{\Gamma_a} \mathbf{U}^H \mathbf{U}\, d\Gamma + \alpha^2 \int_{\Gamma_s} \mathbf{U}_n^H \mathbf{U}_n\, d\Gamma\right] \mathbf{c}_f = -\int_{\Gamma_a} \mathbf{U}^H s\, d\Gamma - \alpha^2 \int_{\Gamma_s} \mathbf{U}_n^H s_n\, d\Gamma$$

(where $\mathbf{U} = \mathbf{U}^{(f)}$, $\mathbf{U}_n = \partial_n \mathbf{U}^{(f)}$, $s_n = \partial_n s$). For reasons of solution efficiency and accuracy, we choose to work with the over-determined least-squares problem (26) instead of forming the normal equations associated with the matrix boundary integrals.

5.3 Pass #2: Exterior Air-domain Solver

The air-domain solver mirrors the fluid domain solver with a couple exceptions. Once we have $\mathbf{c}_f$, we can evaluate the $\partial_n P^{(f)}$ boundary condition on Γ_a (describing the surface velocity; see Figure 9), and then solve to get $P^{(a)}$ in the surrounding air (see Figure 10). We approximate the exterior radiation solution $P^{(a)}$ to (15) by a set of singular multipole sources, whose coefficients are estimated by fitting pressure derivative (normal velocity) data, which is zero except for fluid surface vibrations, $\partial_n P^{(f)}|_{\Gamma_a}$.



40 quadrupoles 60 quadrupoles 80 quadrupoles

Figure 10: Volume-rendered sound pressure, $|P|$ *estimated using the dual-domain solver. Varying quadrupole source counts for the air-domain solver help illustrate visual convergence of the method.*

Pressure Expansion: We again introduce a pressure expansion, but now use fundamental solutions of the air Helmholtz equation:

$$P^{(a)}(\boldsymbol{x}; \boldsymbol{x}_b) = \mathbf{U}^{(a)}(\boldsymbol{x})\mathbf{c}_a, \tag{27}$$

where $\mathbf{U}^{(a)}$ represents n_a singular multipole basis functions,

$$\mathbf{U}^{(a)}(\boldsymbol{x})\mathbf{c}_a = \left[\psi_1^{(a)}\ \psi_2^{(a)}\ \ldots\ \psi_{n_a}^{(a)}\right] \mathbf{c}_a \tag{28}$$

the $\mathbf{c}_a \in \mathbb{C}^{n_a}$ are weights, and each basis function $\psi_j^{(a)}$ satisfies the air Helmholtz equation (and Sommerfeld radiation condition),

$$(\nabla^2 + k_a^2)\, \psi_j^{(a)}(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Omega_a. \tag{29}$$

Since the $P^{(a)}$ expansion (27) satisfies the Green's function PDE in (15), it only remains to select $\mathbf{U}^{(a)}$ then find coefficients, $\mathbf{c}_a$, to satisfy the $\partial_n P$ boundary conditions. The appropriate basis functions here are *singular* multipole solutions to the free-space air Helmholtz equation (as in [James et al. 2006]),

$$\psi_j^{(a)}(\boldsymbol{x}; \boldsymbol{x}_j^{(a)}) = h_\ell^{(2)}(k_a R)\, Y_\ell^m(\theta, \phi), \tag{30}$$

where the source is positioned at $\boldsymbol{x}_j^{(a)}$, $R = \|\boldsymbol{x} - \boldsymbol{x}_j^{(a)}\|_2$, and where $h_\ell^{(2)}$ are spherical Hankel functions of the 2^{nd} kind; $h_\ell^{(2)}(z) = j_\ell(z) - i y_\ell(z) \in \mathbb{C}$, where j_ℓ and y_ℓ are real-valued spherical Bessel functions of the 1^{st} and 2^{nd} kind [Abramowitz and Stegun 1964]. Again we use quadrupole-order multipoles at each point, so an n-point multipole expansion will have $n_a = 9n$ unknown coefficients.

Collocated Least-Squares Estimation: We estimate the coefficients $\mathbf{c}_a$ by matching the boundary conditions (16-17) using weighted least squares. The equation for collocation sample i is

$$\partial_{\boldsymbol{n}_i} \mathbf{U}^{(a)}(\boldsymbol{x}_i)\, \mathbf{c}_a = -\partial_{\boldsymbol{n}_i} P^{(f)}(\boldsymbol{x}_i), \qquad \text{when } \boldsymbol{x}_i \in \Gamma_a \tag{31}$$

$$\partial_{\boldsymbol{n}_i} \mathbf{U}^{(a)}(\boldsymbol{x}_i)\, \mathbf{c}_a = 0, \qquad \text{when } \boldsymbol{x}_i \in \Gamma_s \tag{32}$$

which we then weight by $\sqrt{\Delta a_i}$ to obtain the over-determined N-by-n_a linear least-squares problem,

$$\tilde{\mathbf{A}}\, \mathbf{c}_a = \tilde{\mathbf{b}} \quad \Leftrightarrow \quad \begin{bmatrix} \tilde{\mathbf{A}}_a \\ \tilde{\mathbf{A}}_s \end{bmatrix} \mathbf{c}_a = \begin{bmatrix} \tilde{\mathbf{b}}_a \\ 0 \end{bmatrix}. \tag{33}$$

Note that no relative Neumann-vs-Dirichlet scaling parameter (α) is needed here, since only Neumann $\partial_n P$ constraints exist. Finally, we estimate $\mathbf{c}_a \in \mathbb{C}^{n_a}$ using the robust least-squares solver (§5.6).

5.4 Source Position Selection

Multipole placement affects the quality of the basis functions used in the solver. Traditional equivalent source methods often optimize source placement to increase accuracy [Ochmann 1995; James et al. 2006], however temporally incoherent source positions can ruin frame-to-frame coherence and lead to noise in synthesized sounds. Our numerical experiments indicate that a sufficient number of randomly selected point sources can achieve a plausible sound. To avoid discontinuities, we randomly select fluid particles as point-source locations when the bubble is created. To ensure both (a) temporally coherent basis functions ((23) and (30)) and (b) source positions (and singularities) that remain inside the complex splashing fluid, we advect source positions after each fluid time step.

5.5 Sampling Fluid Geometry

After each fluid time step, we extract an N-vertex triangle mesh of the fluid boundary using marching tetrahedra [Chan and Purisima 1998]; in our examples, mesh resolutions match that of the fluid grid. Each mesh vertex is used as fluid boundary sample at which to impose boundary condition constraints; for vertex $i = 1 \ldots N$ we evaluate and cache the position $\boldsymbol{x}_i$, normal $\boldsymbol{n}_i$, and effective area Δa_i. Sampling fluid geometry can also introduce temporal artifacts in estimated transfer, but these are addressed by temporal filtering/interpolation during the sound rendering process.

One computational difficulty arises when bubbles (or $\psi^{(a)}$) are very close to the fluid boundary, since this can lead to singularities in (19) and (30). Note that singularities are intrinsic to the problem formulation, since bubbles will always rise to the water surface. In practice we choose to expand the fluid surface slightly to regularize such singularities. In our examples, the boundary isosurface is expanded by one fluid-voxel width by extrapolating the level-set isosurface using the fast marching algorithm. While the accuracy is sacrificed slightly, it is more robust numerically, and we found the sound changes imperceptible. The latter point is perhaps unsurprising since vibrations often decay significantly by the time bubbles reach the surface.

5.6 Temporally Coherent Least-Squares Estimation

The under-determined linear systems (26) and (33) can be nearly singular, and must be solved using a robust least-squares solver.

However, common solvers based on the Truncated Singular Value Decomposition (TSVD) should not be used since they can introduce temporal coherence problems: small changes in rank between two time-steps can lead to large magnitude differences in the solution, $\mathbf{c}$ (since the problem is ill-posed). Instead, we use a ridge regression technique with a QR solver (see §12.1 of Golub and Van Loan [1996]). For example, given our N-by-m linear system, $\mathbf{Ac} = \mathbf{b}$, the normal equations solution is $\mathbf{c} = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{Ab}$ but $\mathbf{A}^H\mathbf{A}$ may be near rank deficient. The ridge-regression solution is obtained as $\mathbf{c} = (\mathbf{A}^H\mathbf{A} + \varepsilon^2\mathbf{I})^{-1}\mathbf{Ab}$ for a small $\varepsilon > 0$. Unfortunately explicitly forming $\mathbf{A}^H\mathbf{A}$ can lead to a loss of accuracy (c.f. §5.2 *Discussion*). We instead compute $\mathbf{c}$ by solving the related $(N+m)$-by-m least-squares problem,

$$\begin{bmatrix}\mathbf{A}\\ \varepsilon\mathbf{I}\end{bmatrix}\mathbf{c} = \begin{bmatrix}\mathbf{b}\\ \mathbf{0}\end{bmatrix}, \tag{34}$$

using LAPACK's double-precision QR-based least-squares solver (`zgels`). The resulting $\mathbf{c}$ values (and thus the acoustic transfer pressure values) are more temporally coherent, provided that the same ε value is used; we always use $\varepsilon = 10^{-8}\|\mathbf{A}\|_F$.

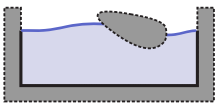
Linear-time Cost: Since the least-squares solver has complexity $O(m^2N)$, the total dual-domain multipole solver cost is $O(n_f^2N + n_a^2N)$, which is linear in the number of boundary samples, N. In our examples, $n_f < n_a \ll N$, and the dual-domain solves required only 1–4 sec/bubble.

5.7 Optimizations and Extensions

Parallelization: Evaluating independent bubble sound sources is a pleasantly parallel computation. In our fluid preprocess, we implemented the dual-domain multipole radiation solver as a service running on an 80-core Xeon cluster. After each fluid time step, the fluid geometry is updated, the cluster computes every active bubble's transfer function coefficients, $\mathbf{c}_a$, using the radiation solver. Since the simulation of fluids and bubbles are not dependent on radiation calculations, the fluid simulator can advance to the next time step while the acoustic transfer is evaluated.

Adaptive Transfer Evaluation: Evaluating transfer coefficients for each bubble at every time step can be a bottleneck when thousands of bubbles exist. Some simple observations can reduce these bottlenecks without compromising accuracy:

1. *Avoid transfer computations for inaudible bubbles:* Our entrainment-forced acoustic bubble exhibits exponentially decaying vibrations which quickly become inaudible especially in the presence of other bubble entrainment events. In practice, we stop the radiation solve for a bubble after its amplitude decays to $1/1000$ of its initial amplitude, e.g., after approximately $T = -\ln 0.001/\beta$.
2. *Temporally adaptive transfer evaluation* avoids computing transfer for bubbles at every timestep. When a bubble's amplitude decays (roughly as $e^{-\beta t}$), we also decrease transfer sampling rates. In our implementation, we use a frequency-dependent sampling rate which roughly gives the sample step size as $\Delta t_{sample} = \Delta t_{fluid}\, e^{\beta t}$, where Δt_{fluid} is the average fluid time-step size. See Figure 11.



Triple-Domain Problem: We have considered a dual-domain fluid-air problem where solid objects are abstracted as a thin mathematical interface, Γ_s. However, the sound radiation model could also include nontrivial solid objects, e.g., for splashing objects (see "Splash" example) or a container of finite thickness. In such cases, the interior fluid-domain solve is identical except for the modified fluid-solid boundary Γ_s. The exterior air-domain problem must be modified to use the larger air-(fluid/solid) interface, and rigid-object scattering can be modeled by adding fixed singular sources $\psi_j^{(a)}$ in the solid region.

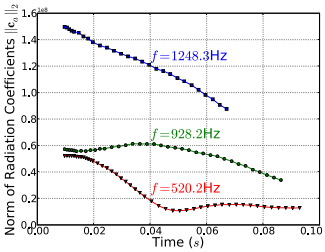


Figure 11: Adaptive transfer evaluation *for three bubbles of different frequency. Bubble lifetimes reflect whether they reached the surface, or became inaudible. We extract a conservative 3× speedup; however, coarser samplings result in greater speedups.*

6 Sound Synthesis Pipeline

We use a two-pass implementation with (1) a fluid and transfer preprocess followed by (2) a sound synthesis phase.

Fluid Preprocess: Algorithm 1 summarizes the main Harmonic Fluids preprocess. After each fluid timestep (line 4), we advect existing bubbles, $\mathcal{B}$ (line 5) and any multipole-solver source points $\mathcal{P}$ for $\psi^{(f)}$ and $\psi^{(a)}$ (line 6). In line 7 we create new bubbles (updating marker positions, bubble seeds, etc., as described in Appendix B), then (line 8) randomly sample new multipole source points for any new bubbles. Level set values ϕ are recorded (line 9) to model frequency variations (§3.4) during sound synthesis.

Parallel transfer computations are then initiated, but only when jobs from the last timestep have completed (line 10). We first mesh the fluid's slightly expanded boundary using marching tetrahedra (§5.5), then extract the vector of mesh vertex positions, normals and effective areas, $(\mathbf{x}, \mathbf{n}, \mathbf{a})$. After initializing the remote-procedure-call (RPC) service (line 13), we launch transfer computation jobs on the remote compute nodes using RPC, and send (line 15) each bubble's parameters $(\omega_d, \xi, \ldots)$, multipole-solver source points $(\mathcal{P}_{bub})$, and surface samples $(\mathbf{x}, \mathbf{n}, \mathbf{a})$. Each bubble's transfer job invokes the dual-domain multipole solver (§5), first solving for $\mathbf{c}_f$ using (26), then solving for $\mathbf{c}_a$ using (33); however, only the small vector $\mathbf{c}_a$ of multipole expansion coefficients are recorded. Adaptive transfer computation (§5.7) allows processing only a subset of bubbles (line 14). Finally, once all bubbles have been scheduled for parallel computation, we proceed with the next fluid time step. In our implementation, bubble vibrations and frequency shift (§3.4) are not evaluated in the fluid/transfer preprocess.

Algorithm 1: FluidPreprocess()

```
1  begin
2      while simulating do
3          t ← t + Δt;
4          timestep_fluid ();
5          advect_bubbles (B);
6          advect_source_points (P);
7          CreateBubbles (B, M, S, t); // (see Appendix B)
8          create_new_source_points (P);
9          record_bubble_φ_values (B);
10         if bubbles exist then
11             mesh ← mesh_fluid_boundary ();
12             (x, n, a) ← pointsNormalsAreas (mesh);
13             for bub ∈ bubblesNeedingTransfer(B) do
14                 eval_transfer (bub, P_bub, (x, n, a));
15 end
```

Example	Fluid & Bubble Simulation					Dual-domain Radiation Solve					
	time	Scale (cm)	Voxels	# of Fluid Particles	# of Bubbles	# of Solves	Frequency range (Hz)	min–max #sources Fluid	min–max #sources Air	<Fit Error> Fluid / Air	max kL Fluid / Air
Droplet	6.4s	14×18×14	70×90×70	1965886	14	2280	500–4K	30–60	80–120	0.06 / 0.18	0.5 / 2.1
Splash	1.5s	45×50×45	90×100×90	3717120	127	25472	300–6K	30–60	80–120	0.08 / 0.24	1.9 / 8.4
Pouring	5.0s	25×40×25	50×80×50	668640	7896	363457	300–6K	30–60	50–80	0.10 / 0.32	1.2 / 5.3
Water Step	8.6s	120×36×72	100×30×60	393376	26657	616846	300–5K	25–60	40–80	0.08 / 0.22	2.0 / 8.7

Table 1: Example Statistics *including temporal duration, grid dimensions, voxel resolutions, the number of FLIP fluid particles and bubbles. Ironically "Water Step" has the fewest fluid particles but the longest fluid simulation time (see Table 2); note that particles are "recycled" at the inlet when they exit the computational cell. "Pouring" and "Water Step" have the most bubbles and transfer solves. Frequencies range from about 300 Hz to 6000 Hz. The highest frequency radiation problems are harder to approximate, since for the same domain lengthscale, L, they span more wavelengths per domain, i.e., have higher kL values. We use roughly twice as many quadrupole sources for the highest frequency than the lowest (and linearly interpolate the rest). Similarly, the air-domain problem's smaller wavelengths make it harder to approximate than the fluid-domain problem, i.e., $k_a L \approx 4.4 k_f L$, and therefore we use more sources for the air domain than the fluid domain. Nevertheless, fitting errors for the least-squares problem (average relative residual error, $\|\mathbf{Ac}-\mathbf{b}\|_2/\|\mathbf{b}\|_2$) were always larger in the air domain. Maximum kL values quantify the difficulty of the highest-frequency Helmholtz approximation problems.*

Sound Synthesis: The sound synthesis stage is much simpler and faster than the fluid preprocess. First, serialized time-series data from the fluid preprocess is loaded, which includes each bubble's trajectory, sampled level-set ϕ values, and multipole expansion coefficients $\mathbf{c}_a$, etc. Given the ear trajectory, the bubble-to-ear transfer functions can be quickly evaluated (in parallel) at the listening position for times when $\mathbf{c}_a$ are available. At each audio-rate time step (of size $\delta t = 1/44100$ seconds), the active set of created/deleted bubbles is updated using loaded data, bubble vibrations are time-stepped (including frequency shifts (§3.4)), and the ear position determined. Each bubble's sound contribution is accumulated, which involves interpolating/filtering its bubble-to-ear transfer function (to the current time), multiplying by its complex-valued oscillator value $\tilde{q}^{(t)}$ (such that q is the real part of $\tilde{q}$), and applying any head-related transfer function (HRTF) [Vorlander 2007]. In our implementation, amplitude filters are used to smoothly blend bubble sound contributions in and out of the sound track since small artifacts can contribute to noise artifacts, especially when thousands of bubbles are present. We synthesize stereo sounds, and use an HRTF model [Brown and Duda 1998] (instead of the using the transfer modulus as in (11)) to exploit the bubble-to-ear transfer function phase for stereo sound:

$$sound(t) = \sum_{b \in \mathcal{B}} \text{HRTF}(P_b^{(t)}\, \tilde{q}_b^{(t)};\ \boldsymbol{x}_b^{(t)} - \boldsymbol{x}_{ear}^{(t)},\ \omega_b^{(t)}) \qquad (35)$$

where the bubble position and frequency parameterize the HRTF.

7 Results

We describe results for four different water sounds: (1) falling water drops, (2) water pouring from a faucet, (3) water splashing from a falling rigid object, and (4) a babbling water step. Please see our accompanying video for all animation and sound results. Statistics are in Table 1, timings in Table 2, and constants in Table 3.

Parallel Implementation: For all our examples, fluid and bubble simulations run on a 16-core 2.4 GHz Xeon node using C++ code. The sound radiation code is compiled into an independent RPC service, and is run on eight 8-core 2.66 GHz Xeon and one 16-core 2.4 GHz Xeon Linux machine. These two parts run in a parallel producer-consumer mode. The fluid simulation generates bubbles and samples surface boundaries as it advances, and launches parallel dual-domain radiation solves using RPC. In our examples, parallel radiation solves complete in less time than each fluid time step, so that parallel sound synthesis adds no additional wall-clock time to fluid simulation. As shown in Table 2, even for simulations with tens of thousands of bubbles, the bottleneck is our fluid simulation.

EXAMPLE (Falling Water Drops): We simulated three large droplets falling from a faucet into a small tank of water (see Figure 12). As in all our examples, transfer is computed for an isolated fluid source; here we ignore surrounding faucet and floor geometry. Since only 14 bubbles were generated, computing costs are dominated by fluid simulation (see Table 2). For convenience, we also provide a "wet" sound using a simple reverberation filter. Recordings of individual bubble sounds were used to originally tune our bubble entrainment model's parameters. See Figure 7 for qualitatively similar spectrograms of a recorded droplet sound and our digital mockup. A convergence analysis is provided in Figure 13 for the fluid-domain and air-domain solvers.

Example	Computation Time (in hours)			
	Fluid	ϕ Update	Radiation	Synthesis
Droplet	0.53 (32%)	1.08 (65%)	0.05 (3%)	0.004 (0.2%)
Splash	0.91 (26%)	2.38 (68%)	0.12 (6%)	0.009 (0.3%)
Pouring	2.57 (29%)	4.34 (49%)	1.86 (21%)	0.044 (0.5%)
Water Step	2.85 (21%)	6.38 (47%)	4.21 (31%)	0.054 (0.4%)

Table 2: Performance Timings: *The parallelized fluid solver (Fluid) and non-parallelized level-set update (ϕ Update) are always the bottleneck in our implementation. Parallelized dual-domain radiation solves (Radiation) are less expensive. Sound synthesis is relatively trivial, and (Synthesis) timings consist primarily of nonoptimized gigabyte file I/O. Overall, transient few-bubble sounds ("Droplet" and "Splash") are significantly less expensive than continuous many-bubble sounds ("Pouring" and "Water Step").*

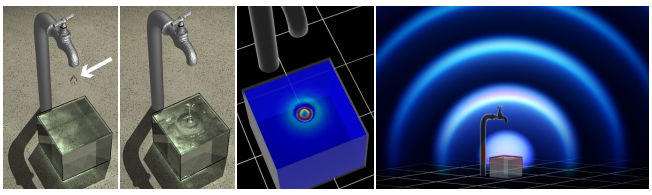


Figure 12: Falling water droplet *splashing and entraining bubbles. The estimated surface normal velocity ($|v_n|^2$) is shown at the time of impact. Resulting pressure waves are volume rendered for illustrative purposes only.*

EXAMPLE (Pouring Water): This example (see Figure 2) is geometrically similar to "water drops," but generated 7896 bubbles (564× more) and required 363,457 transfer solves. Characteristic bubble "chirps" can be heard here and in "water drops."

EXAMPLE (Splashing Water): We simulated a small rigid sphere splashing into a water tank (see Figure 14) using a technique similar to [Carlson et al. 2004]. This example is an instance of the "Triple-Domain Problem" (§5.7), and we place a quadrupole sound source inside the rigid sphere in the air-domain radiation solver. The radia-

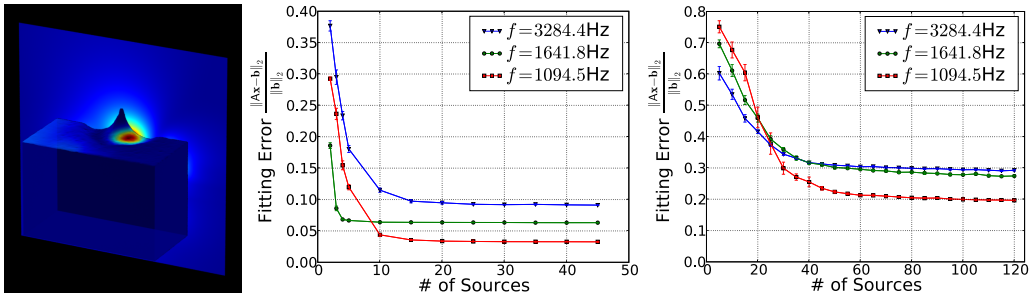


Figure 13: Dual-domain approximation results *for (Left) a single bubble inside a fluid volume deformed after droplet impact: (Middle) fluid-domain and (Right) air-domain convergence rates (with error bars for 95% confidence interval) for randomly distributed quadrupole sources, but fixed geometry and $\boldsymbol{x}_b$. Both curves indicate quick decay to a nominal accuracy suitable for plausible sound rendering.*

tion computation was relatively cheap for this short transient sound.

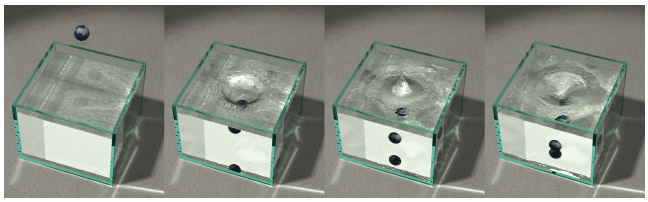


Figure 14: Splash example

EXAMPLE (Babbling Water Step): Our most computationally intensive example is water flowing over an horizontal surface with a small downward step (see Figure 15). The example produces characteristic babbling and chirping sounds.

Fixed sources: Unlike other examples where multipole sources are advected, in this example we fix sources within the water domain to avoid them entering/leaving the domain. Bubbles that reach the interface (or otherwise exit) have their transfer function value frozen at the last computed value.

COMPARISON (to unit transfer): To evaluate the significance of including acoustic transfer effects, we also synthesized pouring sounds with "unit transfer" ($P = 1$). The resulting sound is harsh and unrealistic, which is perhaps not surprising given the complex structure of transfer values (see Figure 4).

COMPARISON (constant vs. changing bubble frequency): We synthesized pouring and "water step" sounds with and without bubble frequency changes (§3.4) to demonstrate their subtle but perceptually important effect. Transfer functions were unchanged. The constant-frequency sounds tend to sound more like computer-generated noise, whereas the nonconstant-frequency sounds have richer variations and exhibit more chirping and babbling sounds.

COMPARISON (to real-world splashing): To compare against an actual splashing sound with constant visual stimulus, we replaced the sound track with recordings of real-world splash mock-ups. We provide a single comparison, with mono-phonic sound. Although the sounds are qualitatively similar, the real sound has more complex tonal variations during the latter splashing phase.

COMPARISON (different radiation solver errors): A strength of our radiation solver is that it can exploit the relatively low boundary-condition accuracy (recall Figure 13) needed to produce plausible fluid sounds in the listener's far-field location. To evaluate the impact of larger radiation solver errors, the video compares "water step" animations with different boundary-condition errors in the fluid/air domain radiation solvers (18%/40%, 12%/35%, 8%/22%). Although the sounds are qualitatively similar, the low-accuracy radiation coefficients tend to exhibit greater temporal variations (likely due to the ill-posedness of the least-squares approximation) resulting in greater noise in the synthesized sound. Some listeners also perceived localization errors in low-frequency bubbles, possibly due to left/right-ear transfer errors in phase and/or amplitude. We recommend using higher-accuracy approximations when possible to minimize artifacts.

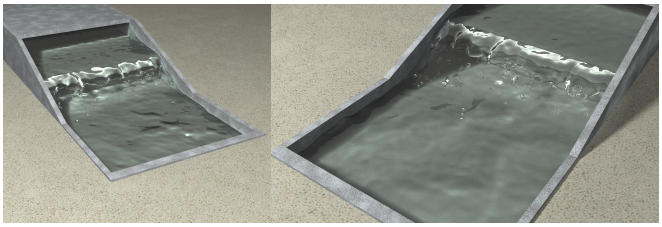


Figure 15: Water "babbles" as it flows over a small step

8 Limitations and Future Work

Fluid sound synthesis is a new area, and significant challenges remain. Our proposed model enables physically based sound rendering for harmonic fluid phenomena, however its physical simplifications and limitations provide many avenues for future work.

The mono-frequency acoustic bubble provides a good starting point for modeling sound radiation, but is rather simplistic. It neglects higher-order linear vibration modes, which is often justified by the fact that higher-order linear modes radiate less well than monopoles. However more complex nonlinear vibration modes also exist, and can contribute to far-field radiation [Leighton 1994]. Both linear and nonlinear bubble vibrations can also lead to significant inter-bubble coupling effects; dense bubble concentrations, such as in foam or plumes, pose particular nonlinear challenges, especially for radiation modeling [Deane 1997]. Very large bubbles can be important, and demand special attention given the complexities of nonlinear vibrations and acoustic radiation. Bubbles approaching the interface can lead to singularities in our boundary integral solver, and a better model of nonspherical acoustic bubbles at the interface is needed. Bubble popping and merging are missing interfacial phenomena, as are boiling and fizzing.

Bubble forcing could be improved. We only considered an initial entrainment-related pressure impulse, but later pressure forces can be important, especially for larger bubbles [Leighton 1994], e.g., consider large bubbles rising from a scuba diver. Unfortunately audio-rate pressure forcing can be expensive to evaluate accurately.

Our bubble entrainment model is stochastic, but actual entrainment statistics are more complex [Leighton 1994]. Our model also lacks dependence on pressure, which can be important for impact and splashing, especially at high velocities [Franz 1959]. Future models should reduce parameter tuning needed to achieve realistic bubble distributions and spectra.

Our dual-domain multipole solver can be a good approximation for

compact sound sources (with modest kL values), however it is less well-suited to larger sources, such as a swimming pool. Similarly, we have not considered underwater listeners, which could avoid air-domain solves but would be complicated by large fluid domains. We have modeled harmonic fluid sound sources, but it still remains to integrate these sound models into larger acoustic environments. Including scattering effects of surrounding geometry, especially for larger sound sources, remains a challenge. Low-error approximations may necessitate more sophisticated frequency-domain solvers [Gumerov and Duraiswami 2005]. However, reviewers pointed out that analytical solutions for simplified planar fluid-interface geometry may suffice for some applications.

Splashing sounds produced by an impacting elastic object can also include significant elastic object sound contributions [Franz 1959]. In general, fluid-solid-air coupling methods are needed to capture the effects of vibrating solid objects, e.g., when pouring water into a plastic cup or metal sink.

Opportunities exist for accelerating sound synthesis, and real-time Harmonic Fluid sound sources appears feasible. The frequency-domain radiation preprocess is pleasantly parallel, but numerous bubble sound sources may become a bottleneck. Opportunities clearly exist for perceptually based sound rendering by using degraded sound quality and exploiting perceptual masking, etc. Time-domain solvers for the wave equation may also be a viable methods for integrating the contributions of many bubbles. Finally, physically based sound rendering might be combined with data-driven and stochastic methods to exploit complementary advantages for more complex and noise-like phenomena, e.g., Niagara Falls.

A Acoustic Bubble Formulae

The bubble's undamped natural frequency is [Leighton 1994]

$$\omega_0 = \sqrt{3\gamma p_0 - 2\sigma/r_0}/\left(r_0\sqrt{\rho}\right), \tag{36}$$

and its damping rate is given by

$$\beta = \omega_0\delta/\sqrt{\delta^2+4} \tag{37}$$

where $\delta = \delta(\omega_0, r_0) = \delta_{rad}+\delta_{vis}+\delta_{th}$ is a dimensionless damping value describing damping due to wave radiation (rad), fluid viscosity (vis), and thermal conductivity (th):

$$\delta_{rad} = \frac{\omega_0 r_0}{c_f}, \quad \delta_{vis} = \frac{4\mu_f}{\rho\omega_0 r_0^2}, \quad \delta_{th} = 2\frac{\sqrt{\psi-3} - \frac{3\gamma-1}{3(\gamma-1)}}{\psi-4}, \tag{38}$$

with $\psi = \frac{16}{9(\gamma-1)^2}\frac{G_{th}g}{\omega_0}$. The numerous parameters are as follows (values given in Table 3): c_f is the fluid's speed of sound; p_0 is the hydrostatic pressure of the liquid (which we always approximate as 1 atm in our simulations); γ is the gas's heat capacity ratio (or adiabatic index); μ_f is the liquid's shear viscosity; σ is the fluid surface tension coefficient; $G_{th} = \frac{3\gamma p_0}{4\pi\rho D_g}$ is the thermal damping constant at resonance; and D_g is the gas's thermal diffusivity.

Our *ad hoc* entrainment-related blending function is:

$$q_{blend}(t) = \begin{cases} q(t)e^{-\frac{(e^{-\beta t}-0.85)^2}{0.0028125}}, & e^{-\beta t} \geq 0.85 \\ q(t), & e^{-\beta t} < 0.85 \end{cases} \tag{39}$$

B A Stochastic Model of Bubble Entrainment

Complex multi-scale interfacial mixing processes are responsible for bubble formation, but we desire a simplified computational model. We track mixing via the rapid movement of interfacial fluid material into the fluid volume by monitoring rapid changes in ϕ values of fluid material from a value near zero, to a value revealing it is now deep in the fluid. We place *markers* on a layer of particles near the surface: fluid particle i gets a marker if $\phi_\varepsilon < \phi_i < 0$, where ϕ_ε is a constant specifying the thickness of the marker layer (we use $\phi_\varepsilon = -2h$). At each time step, we track each marker's isosurface value. Dramatic decreases in marker ϕ_i values indicate the potential for bubble creation at the marker's position. When a sufficient ϕ_i decrease is detected, we call that marker a *bubble seed*—we use these in our bubble creation process. Markers and bubble seeds are illustrated in Figure 16.

Parameter	Value	Description
g	9.8 m/s^2	gravitational acceleration
ρ	1000 kg/m^3	water density
p_0	101.325 kPa	atmospheric pressure
γ	1.4	specific heat ratio of air
σ	0.0726 N/m	surface tension coefficient of water
D_g	2.122e-5 m^2/s	thermal diffusivity of gas
c_f	1497 m/s	sound speed in water
c_a	343 m/s	sound speed in air
μ_f	8.9e-4 $Pa \cdot s$	shear viscosity of water
G_{th}	1.60 $\times 10^6$ s/m	thermal damping constant

Table 3: Physical constants used in our simulations

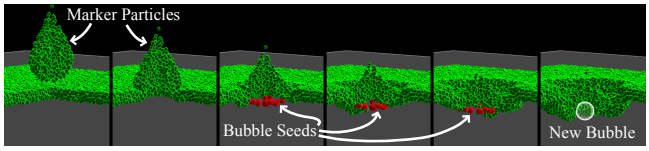


Figure 16: Bubble entrainment by a falling water drop *(cut-away view)*

Unfortunately, the reconstructed isosurface field can be noisy, so that simply detecting rapid decreases in ϕ_i values is not robust. Therefore, we use linear regression to estimate the slope, $\frac{d\phi_i}{dt}$, by maintaining a sliding window (of width between $0.006sec$ and $0.01sec$) for each marker's ϕ_i values. The moment the slope exceeds a threshold (between $-0.9m/s$ and $-2.2m/s$), the marker becomes a bubble seed.

Bubble seed TTL and strength: Each bubble seed has (a) a creation time, t_0, (b) a time-to-live (TTL) value, T_{ttl}, after which the seed dies, and (c) a "bubble creation strength" value, $w_s(t)$, which is 1 initially and decays thereafter. Given a seed created at time t_0, we model the bubble seed's strength by the cubic spline:

$$w_s(t) = \begin{cases} 1-4\tau^3, & 0 \leq \tau \leq 0.5 \\ 4(1-\tau)^3, & 0.5 < \tau \leq 1 \end{cases} \quad \text{where} \quad \tau = \frac{t-t_0}{T_{ttl}}.$$

This distribution of strength-weighted bubble seeds provides clues for creating bubbles at seed positions. We model the number of bubble creation attempts (per time step) as proportional to the sum total of seed strengths:

$$N_{bub} = \kappa h^2 \Delta t \sum_{\text{s is seed}} w_s(t), \tag{40}$$

where κ is a parameter controlling the bubbliness of the flow; and to try to make the bubble generation rate independent of spatial and temporal discretizations we scale by the interface fluid-grid resolution, h^2, and the time step size, Δt.

Bubble radius and spectra: The radii of created bubbles strongly affects the spectrum of the generated sound. In order to approximate the spectra of real fluid sounds, we use a probability distribution function to randomly sample bubble radii. In principle, by selecting a proper bubble radius distribution, we can match the sound spectrum to real sound cases—although not a sufficient condition for realistic sounds. Similar to [Greenwood and House 2004], we use a *Gaussian* distribution: mean and deviation were calibrated by matching the characteristic pitch to typical recorded sounds.

Radius rejection sampling: For each bubble created at a time step, we randomly select a seed as the bubble's initial position. This

provides a density-based sampling, so that well-seeded regions are more likely to create bubbles. Given a randomly sampled bubble radius and position, to avoid placing unrealistically large bubbles in small regions, our check to determine if enough local seeds s are inside the bubble is:

$$r_{par}^2 \sum_{\boldsymbol{x}_s \in Bubble} w_s(t) \quad < \quad \tau_{rej}\, r_0^2 \tag{41}$$

where r_{par} is the fluid particle radius ($0.22h$ in our simulations), and τ_{rej} controls bubble sizes (our examples use τ_{ref} values between 0.9 and 2). Otherwise we create a bubble, and the seeds inside a sphere of radius $1.5r_0$ are removed.

Algorithm: Our bubble creation method is summarized in Algorithm 2. In reality, bubbles are generated at very high rates, so that sounds from splashing or pouring appear continuous. To avoid discretization artifacts here, bubble creation times are uniformly distributed during the time step. The bubbles' initial positions and velocities are interpolated from bubble seeds.

Algorithm 2: CreateBubbles($\mathcal{B}$, $\mathcal{M}$, $\mathcal{S}$, t)

Data: The set of current bubbles $\mathcal{B}$, seeds $\mathcal{S}$, current markers $\mathcal{M}$, and current time t

```
begin
    update_markers(M);
    sample_isosurface_value(M);
    create_seeds(M, S);
    update_seed_strengths(S);
    N_bub ← num_bubble_creation_attempts(S);
    for i = 1 ... N_bub do
        seed ← random_select_seed(S);
        r ← random_select_radius;
        if not reject_bubble(seed, r, S) then
            create_bubble(seed.pos, r);
            remove_seeds(seed, r, S);
end
```

Parameter Tuning: Model parameters can be tuned manually for best results. We first adjust the bubbly flow to get a plausible number of bubbles by tuning κ in (40) and τ_{rej} in (41), with unit values being good initial guesses (see Table 4). In the second pass, we can adjust the (Gaussian) distribution for the bubbles' radius (and thus frequency), e.g., to approximate spectra of recorded fluid sounds.

	Drop	Splash	Pour	WStep	Description	Eqn
κ	3.2	1.3	1.0	1.8	bubbliness	(40)
τ_{rej}	2.0	0.9	1.4	1.5	radius limiter	(41)

Table 4: User-specified entrainment parameters *are roughly of unit size.*

C Derivation of Source Strength, S_b

We estimate the delta-function source strength, S_b, of a point-like bubble from its "divergence sourcing" strength (c.f. [Kim et al. 2007]). First, we take the divergence of the relationship between harmonic acoustic pressure $p(\boldsymbol{x})$ and acoustic velocity $\boldsymbol{v}(\boldsymbol{x})$,

$$\nabla p = -i\omega\rho \boldsymbol{v} \quad \Rightarrow \quad \nabla^2 p = -i\omega\rho\,(\nabla\cdot\boldsymbol{v})\,. \tag{42}$$

Given our divergence singularity of the form, $\nabla^2 p = S_b\delta(\boldsymbol{x}-\boldsymbol{x}_b)$, we can estimate S_b by integrating over a small domain Ω_b containing the tiny bubble (so that $\int_{\Omega_b}\delta(\boldsymbol{x}-\boldsymbol{x}_b)d\Omega = 1$):

$$S_b = -i\omega\rho \int_{\Omega_b} (\nabla\cdot\boldsymbol{v})\, d\Omega. \tag{43}$$

The divergence theorem, and the rate of fluid expulsion from the volume Ω_b due to ε-amplitude pulsations, $r = r_0 + \varepsilon e^{+i\omega t}$, yields

$$\int_{\Omega_b} \left(\nabla\cdot\boldsymbol{v}e^{+i\omega t}\right) d\Omega = -\frac{d\,V_b}{dt} = -4\pi i\omega r^2 \varepsilon e^{+i\omega t}. \tag{44}$$

It follows that S_b is given by (10).

Acknowledgements: We would like to thank the anonymous reviewers for helpful feedback. This work was supported in part by the National Science Foundation (CAREER-0430528, HCC-0905506), the Alfred P. Sloan Foundation, Pixar, Intel and Autodesk. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

ABRAMOWITZ, M., AND STEGUN, I. A. 1964. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York.

ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively Sampled Particle Fluids. In *Proc. ACM SIGGRAPH*.

BEGAULT, D. 1994. *3-D sound for virtual reality and multimedia*. Academic Press Professional, Inc. San Diego, CA, USA.

BLAKE, W. 1986. *Mechanics of Flow-Induced Sound and Vibration*. Academic Press.

BONNEEL, N., DRETTAKIS, G., TSINGOS, N., VIAUD-DELMON, I., AND JAMES, D. 2008. Fast modal sounds with scalable frequency-domain synthesis. *ACM Trans. on Graphics 27*, 3 (Aug.), 24:1–24:9.

BRAGG, S. 1920. *The World of Sound*. G. Bell and Sons Ltd., London.

BROWN, C. P., AND DUDA, R. O. 1998. A Structural Model for Binaural Sound Synthesis. *IEEE Trans. on Speech and Audio Processing 6*, 5.

CARLSON, M., MUCHA, P. J., AND TURK, G. 2004. Rigid Fluid: Animating the interplay between rigid bodies and fluid. *ACM Trans. on Graphics 23*, 3 (Aug.), 377–384.

CHAN, S., AND PURISIMA, E. 1998. A new tetrahedral tesselation scheme for isosurface generation. *Computers & Graphics 22*, 1, 83–90.

CHILDS, E. 2001. The Sonification of Numerical Fluid Flow Simulations. In *Intl. Conf. on Auditory Display (ICAD 2001)*.

CLEARY, P. W., PYO, S. H., PRAKASH, M., AND KOO, B. K. 2007. Bubbling and Frothing Liquids. *Proc. ACM SIGGRAPH*.

COOK, P. 2002. *Real Sound Synthesis for Interactive Applications*. AK Peters, Ltd.

DEANE, G. B. 1997. Sound generation and air entrainment by breaking waves in the surf zone. *The Journal of the Acoustical Society of America 102* (November), 2671–2689.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2003. Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. *ACM Trans. on Graphics 22*, 3 (July), 732–740.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2004. Synthesizing sound from turbulent field using sound textures for interactive fluid simulation. *Computer Graphics Forum 23*, 3 (Sept.), 539–545.

ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and rendering of complex water surface. *ACM Trans. on Graphics 22*, 3, 736–744.

FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. *Proc. ACM SIGGRAPH*, 23–30.

FOSTER, N., AND METAXAS, D. 1996. Realistic Animation of Liquids. *Graphical Models and Image Processing 58*, 5, 471–483.

FRANZ, G. J. 1959. Splashes as Sources of Sound in Liquids. *Journal of the Acoustical Society of America 31* (Aug), 1080–1096.

FUNKHOUSER, T. A., MIN, P., AND CARLBOM, I. 1999. Real-time acoustic modeling for distributed virtual environments. In *Proc. of SIGGRAPH 99*, 365–374.

GOLUB, G., AND VAN LOAN, C. 1996. *Matrix computations*, third ed. Johns Hopkins University Press.

GREENWOOD, S., AND HOUSE, D. 2004. Better with Bubbles: Enhancing the Visual Realism of Simulated Fluid. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*.

GUMEROV, N., AND DURAISWAMI, R. 2005. *Fast multipole methods for the Helmholtz equation in three dimensions*. Elsevier.

HONG, J.-M., LEE, H.-Y., YOON, J.-C., AND KIM, C.-H. 2008. Bubbles alive. *ACM Trans. on Graphics 27*, 3 (Aug.), 48:1–48:4.

HOWE, M. S. 1998. *Acoustics of Fluid-Structure Interactions*. Cambridge Press.

HOWE, M. S. 2002. *Theory of Vortex Sound*. Cambridge Press.

IMURA, M., NAKANO, Y., YASUMURO, Y., MANABE, Y., AND CHIHARA, K. 2007. Real-time generation of CG and sound of liquid with bubble. In *ACM SIGGRAPH 2007 Posters*.

JAMES, D. L., BARBIC, J., AND PAI, D. K. 2006. Precomputed Acoustic Transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Trans. on Graphics 25*, 3 (July), 987–995.

JENSEN, F. 1994. *Computational Ocean Acoustics*. American Institute of Physics.

KIM, T., AND CARLSON, M. 2007. A simple boiling module. In *Proc. of Symp. on Computer Animation (SCA)*.

KIM, B., LIU, Y., LLAMAS, I., JIAO, X., AND ROSSIGNAC, J. 2007. Simulation of bubbles in foam with the volume control method. *ACM Trans. on Graphics 26*, 3 (July), 98:1–98:10.

KITA, E., AND KAMIYA, N. 1995. Trefftz method: An overview. *Advances in Engineering Software 24*, 89–96.

KLEINER, M., DALENBAECK, B., AND SVENSSON, P. 1993. Auralization-An Overview. *Journal-Audio Engineering Society 41*, 861–861.

LEIGHTON, T. 1994. *The Acoustic Bubble*. Academic Press.

LONGUET-HIGGINS, M. S. 1990. An analytic model of sound production by raindrops. *J. Fluid Mech. 214*, 395–410.

MCCABE, R. K., AND RANGWALLA, A. A. 1994. Auditory display of computational fluid dynamics data. In *Auditory Display: Sonication, Audication, and Auditory Interfaces; Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVIII*, Addison Wesley, G. Kramer, Ed., 321–340.

MINER, N. E., AND CAUDELL, T. P. 2005. Using wavelets to synthesize stochastic-based sounds for immersive virtual environments. *ACM Trans. on Applied Perception 2*, 4, 521–528.

MINNAERT, M. 1933. On musical air-bubbles and sounds of running water. *Phil Mag 16*, 235–248.

O'BRIEN, J. F., COOK, P. R., AND ESSL, G. 2001. Synthesizing sounds from physically based motion. In *Proc. of ACM SIGGRAPH 2001*, 529–536.

O'BRIEN, J. F., SHEN, C., AND GATCHALIAN, C. M. 2002. Synthesizing sounds from rigid-body simulations. In *ACM SIGGRAPH Symposium on Computer Animation (SCA)*, 175–181.

OCHMANN, M. 1995. The Source Simulation Technique for Acoustic Radiation Problems. *Acustica 81*.

OGUZ, H., AND PROSPERETTI, A. 1990. Bubble entrainment by the impact of drops on liquid surfaces. *J. Fluid Mech. 219*, 143–179.

OHAYON, R. 2004. Reduced models for fluid-structure interaction problems. *Int. J. Numer. Meth. Engng 60*, 1, 139–152.

OSHER, S., AND FEDKIW, R. 2003. *Level Set Methods and Dynamic Implicit Surfaces*. Springer.

PUMPHERY, H., CRUM, L., AND BJØRNØ, L. 1989. Underwater sound produced by individual drop impacts and rainfall. *J. Acoust. Soc. Am. 85*, 1518–1526.

STAM, J. 1999. Stable fluids. In *Proc. ACM SIGGRAPH*, 121–128.

STRASBERG, M. 1953. The pulsation frequency of non-spherical gas bubbles in liquids. *J. Acoust. Soc. Am. 25*, 536–537.

TAKALA, T., AND HAHN, J. 1992. Sound rendering. In *Computer Graphics (Proc. of SIGGRAPH 92)*, 211–220.

THUEREY, N., SADLO, F., SCHIRM, S., AND M. MÜLLER, M. G. 2007. Real-time simulations of bubbles and foam within a shallow-water framework. In *Proc. of Symp. on Computer Animation (SCA)*.

TSINGOS, N., FUNKHOUSER, T., NGAN, A., AND CARLBOM, I. 2001. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proc. of ACM SIGGRAPH 2001*, 545–552.

TSINGOS, N., GALLO, E., AND DRETTAKIS, G. 2004. Perceptual audio rendering of complex virtual environments. *ACM Trans. on Graphics 23*, 3 (Aug.), 249–258.

URICK, R. 1975. *Principles of Underwater Sound*. McGraw-Hill.

VAN DEN DOEL, K., AND PAI, D. K. 1996. Synthesis of shape dependent sounds with physical modeling. In *Intl. Conf. on Auditory Display (ICAD 96)*.

VAN DEN DOEL, K., KRY, P. G., AND PAI, D. K. 2001. FoleyAutomatic: Physically-Based Sound Effects for Interactive Simulation and Animation. In *Proc. of ACM SIGGRAPH 2001*, 537–544.

VAN DEN DOEL, K. 2005. Physically based models for liquid sounds. *ACM Trans. on Applied Perception 2*, 4, 534–546.

VORLANDER, M. 2007. *Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality*. Springer Verlag.

ZHENG, W., YONG, J.-H., AND PAUL, J.-C. 2006. Simulation of bubbles. In *Proc. of Symp. on Computer Animation (SCA)*, 325–333.

ZHU, Y., AND BRIDSON, R. 2005. Animating sand as a fluid. *ACM Trans. on Graphics 24*, 3 (Aug.), 965–972.