SPH for Weakly Compressible Fluids

Egor Larionov*
University of Waterloo

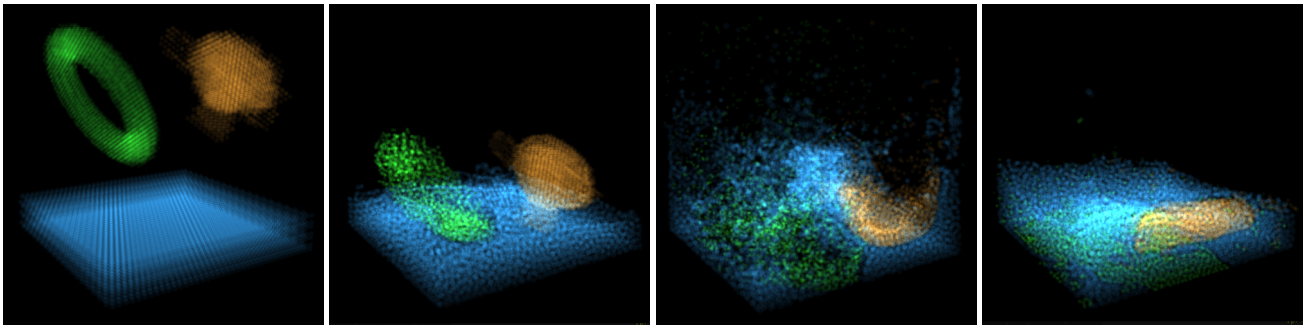

Figure 1: *Splash!*

Abstract

Modern fluid simulators have been able to reproduce the aesthetics of many naturally occurring fluids. Much research [Monaghan 2005; Ihmsen et al. 2014a] has been dedicated to smoothed particle hydrodynamics (SPH) for simulating fluids in computer graphics. This paper compares some of the fundamental state-of-the-art SPH techniques for simulating weakly compressible fluids. In particular, different methods to compute pressure, pressure forces, viscosity and surface tension are compared for efficiency and visual accuracy. In addition, various methods for handling boundary conditions are analysed. All findings are demonstrated with a robust SPH implementation, specifically designed for comparing various SPH techniques.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modelling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: physically-based animation, fluid simulation, smoothed particle hydrodynamics, viscosity, kernel optimization

1 Introduction

Fluids have fascinated humanity for centuries. The beauty in nature's flow patterns has captivated our imagination even predating any rigorous mathematical or scientific investigation of fluid flows from the seventeenth century. The work of Leonardo da Vinci, for instance, features the complexity and elegance of naturally occurring fluid flows [Ball 2011]. Moreover, people interact with various kinds of fluids on a daily basis, so it is only natural that fluid simulation is an important part of computer graphics.

*e-mail:elariono@uwaterloo.ca

Early fluid simulators can be categorized into particle-based and grid-based methods. Purely particle-based methods store and compute fluid quantities (density, viscosity, force, etc.) at discrete points (called particles), which follow the motion of the fluid through time. In contrast, purely grid-based methods generally compute the same quantities at points fixed in space, and often aligned into a hexahedral grid structure. There are advantages and disadvantages to both methods, so both are currently employed to simulate fluids in industry [Sid 2013], although neither in its pure form. For instance, particle-based methods rely on efficient neighbourhood search algorithms, which often utilize a regular grid. Grid-based methods, on the other hand, handle advection poorly, and are often augmented by marker particles to transport mass through the grid.

This work focuses on a particular particle-based method called smoothed particle hydrodynamics (SPH), which approximates the solutions of the equations of fluid dynamics by interpolating fluid quantities at discrete particles using smoothing kernels. The original kernel estimation technique was developed by statisticians to estimate probability density functions given a discrete set of random variables (see [Rosenblatt 1956] and [Parzen 1962] for details). In contrast, grid-based methods use finite difference schemes to interpolate quantities. There are a number of advantages SPH has over other methods, as elegantly outlined in [Monaghan 2005]:

- Being a particle-based method, SPH treats advection exactly. For instance, each particle carries a fixed mass, which follows the particle through the velocity field exactly.
- Multi-phase flow is trivially accomplished with SPH, while separate methods are needed to handle boundary conditions between neighbouring phases [Losasso et al. 2006] for grid-based methods.
- Naturally, particle methods like SPH approximate continuum fluid equations on a more fundamental level where the particles approximate the underlying molecular system of the fluid. This often allows SPH to handle complex physics easily. It also makes SPH particularly robust against large discontinuities (or shocks) in the underlying force fields.
- SPH can be extended to handle multiple resolutions dependent on time and space, making it suitable for simulating large bodies of fluid where lots of detail is needed on the surface, but not necessarily in the interior. See [Chaniotis et al. 2002] and [Feldman and Bonet 2007] for details on dynamic particle

refinement. Another multi-resolution method has been introduced in [Solenthaler and Gross 2011] where two distinct but coupled simulations evolve the SPH fluid on two different resolutions respectively.

- Computation happens only at particle locations, exactly where the fluid is. In contrast, grid-based methods compute quantities at fixed locations, which result in various grid artifacts. For instance, separated droplets of fluid may disappear in a grid-based method if the droplet is smaller than a grid cell.

Unfortunately solving the Navier-Stokes equations for weakly compressible or incompressible, free-surface fluids using SPH poses some difficulties. For instance, in standard SPH, particles experience pressure forces from neighbouring particles due to density fluctuations, thus in a body of fluid, surface particles may incorrectly experience lower pressures than their interior counterparts, due to the absence of surrounding air particles. A number of different approaches exist to resolve this issue including various equations of state (EOS) [Morris and Monaghan 1997; Monaghan 1994; Desbrun and Gascuel 1996; Adams et al. 2007], iterative EOS solvers including predictive corrective incompressible SPH (PCISPH) [Solenthaler and Pajarola 2009] and local Poisson SPH (LPSPH) [He et al. 2012]. There are methods, however, that iteratively solve the pressure Poisson equation (PPE) [Shao and Lo 2003], a technique common to grid-based methods. The current state-of-the-art solution to incompressibility in SPH is implicit incompressible SPH (IISPH), developed in [Ihmsen et al. 2014b].

Strategies for computing viscosity, surface tension and boundary forces also vary among different SPH implementations.

This paper discusses elements from a two key SPH implementations including

- [Müller et al. 2003], which first achieved interactive speeds for fluid simulation based on the Navier-Stokes equations;
- [Becker and Teschner 2007], which combined existing SPH ideas found in [Monaghan 2005], with a new surface tension term to achieve convincing results for weakly compressible SPH (WCSPH);

2 Related Work

Starting with Foster and Metaxas [1996; 1997], who introduced a finite-difference scheme to solve the Navier-Stokes equations, grid-based methods for fluid simulation have become increasingly popular in the computer graphics community. These methods use the Eulerian viewpoint to describe the motion of fluid on a fixed grid, and as a result, exhibit difficulties computing advection and conservation of mass and volume. Stam [1999] extended the grid-based approach to use semi-Lagrangian advection. Stam's method suffered from large numerical dissipation, which was alleviated by vorticity confinement, introduced in [Fedkiw et al. 2001] beautifully simulating smoke effects. Later, Fedkiw and Foster [2001] extended these ideas to incompressible fluids, and improved mass conservation, by introducing the level set method [1988] augmented by marker particles for surface tracking. Further notable developments of grid-based methods include improved boundary conditions [Rasmussen et al. 2004], two-way obstacle coupling [Carlson et al. 2004; Guendelman et al. 2005; Robinson-Mosher et al. 2008], multiple fluid interaction [Losasso et al. 2006] among others. Grid-based methods have a long history, tracing back to the work of Harlow and Welch [1962; 1965] who developed the particle-in-cell (PIC) and marker-and-cell (MAC) methods for compressible and incompressible flow respectively. The fluid-implicit-particle (FLIP) was later developed in [Brackbill and Ruppel 1986] to correct PIC for its large numerical dissipation. FLIP was later extended to handle incompressible flow [Kothe and Brackbill 1992].

Smoothed particle hydrodynamics was introduced by [Gingold and Monaghan 1977] and [Lucy 1977] independently. Since then, SPH has been thoroughly developed for various applications including fluid simulation.[Desbrun and Gascuel 1996] first introduced SPH into the computer graphics community to model deformable bodies. [Morris 2000] proposed a robust method for simulating surface tension which was later succeeded by [Akinci et al. 2013]. [Bonet and Kulasegaram 2002] described the application of the Shepard filter to SPH in order to improve density estimates near the boundaries; this idea is used extensively in SPH simulations. Shortly thereafter, SPH became the first method to achieve interactive speeds for fluid simulation using Navier-Stokes equations [Müller et al. 2003]. Since then SPH became a popular topic in computer graphics, and many adjustments have been introduced to improve weakly compressible and incompressible fluid simulation. For instance [Solenthaler and Pajarola 2008] proposed a correction for the multiphase behaviour of SPH fluids with largely density ratios. [Akinci et al. 2012] and [Schechter and Bridson 2012] introduced significant improvements to boundary handling. [Solenthaler and Gross 2011] uses coupled simulations to improve resolution in highly detailed areas in an SPH fluid. Naturally various optimizations and visualization methods [2003; 2011] have been introduced to aid in SPH development. For a detailed introduction of SPH, see [Monaghan 2005], and for state-of-the-art SPH developments in computer graphics, see [Ihmsen et al. 2014a].

3 Smoothed Particle Hydrodynamics

The context in SPH consists of a set of particles (indexed by i), at positions $\boldsymbol{x}_i$, with velocities $\boldsymbol{v}_i$, and masses m_i. Each particle also has an associated density ρ_i and pressure p_i. For a fluid, the motion of these particles are governed by the Navier-Stokes equations:

$$\frac{d\boldsymbol{v}}{dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{v} + \boldsymbol{f}^{other}$$
$$\nabla \cdot \boldsymbol{v} = \mathbf{0}$$

assuming constant viscosity ν. The first and second terms represent accelerations due to pressure and viscosity respectively. $\boldsymbol{f}^{other}$ encapsulates the force per unit area of extraneous forces including surface tension, gravity and collisions.

Suppose we want to compute a quantity A (density, pressure, forces, etc.) at a position $\boldsymbol{x}$ in the fluid. We can write $A(\boldsymbol{x})$ as an integral over the whole volume:

$$A(\boldsymbol{x}) = \int_V A(\boldsymbol{r})\delta(\boldsymbol{x}-\boldsymbol{r})d\boldsymbol{r}$$

where δ is the Dirac delta "function". We may approximate this integral by choosing a smooth distribution W, also called a smoothing kernel, to replace δ:

$$A(\boldsymbol{x}) \approx \int_V A(\boldsymbol{r})W(\boldsymbol{x}-\boldsymbol{r})d\boldsymbol{r}. \quad (1)$$

where W is normalized such that $\int_V W = 1$. Furthermore, we can discretize (1) by noting that the right-hand side can be written as

$$\int_V \frac{A(\boldsymbol{r})}{\rho(\boldsymbol{r})}W(\boldsymbol{x}-\boldsymbol{r})\rho(\boldsymbol{r})d\boldsymbol{r},$$

where $\rho(\boldsymbol{r})d\boldsymbol{r}$ is a mass element. Thus we can discretize this equation by taking the sum over the particles in the volume, where the

mass is known:

$$A(\boldsymbol{x}) \approx \sum_i \frac{A(\boldsymbol{x}_i)}{\rho_i} W(\boldsymbol{x} - \boldsymbol{x}_i) m_i. \tag{2}$$

Finally, choosing W to have compact support (say radius h) greatly reduces the number of terms in the sum. Thus we may compute the density of a particle i as follows:

$$\rho_i = \sum_j m_j W_{ij}, \tag{3}$$

where I use the notation $W_{ij} = W(\boldsymbol{x}_i - \boldsymbol{x}_j, h)$, so the size of the kernel is implied. For a thorough investigation of the errors involved in the described approximations, see [Monaghan 2005].

It is straight forward to compute spacial derivatives of SPH quantities:

$$\nabla A_i = \sum_j m_j \frac{A_j}{\rho_j} \nabla W_{ij}, \qquad \nabla^2 A_i = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W_{ij}$$

However, various approximations for spatial derivatives have been proposed to improve their accuracy. For instance in order to symmetrize forces (that is to conserve momentum) [Monaghan 1992], the gradient may be computed as follows:

$$\nabla A_i = \rho_i \sum_j m_j \left(\frac{A_i}{\rho_i^2} + \frac{A_j}{\rho_j^2} \right) \nabla W_{ij} \tag{4}$$

3.1 Smoothing Kernels

Many different smoothing kernels have been proposed in SPH literature. The most popular kernels are outlined in Appendix A, however the choice of kernel seldom has adverse effects.

3.2 Neighbourhood Search

The SPH interpolation sum (as in (2)) spans over particles within h distance of the target position, since other particles will have zero contribution to the sum. Searching for these nearby particles, can be efficiently achieved with a regular axis aligned grid structure containing cubic cells with side length h. Each cell in the grid contains a list of particles that occupy that cell.

Suppose that grid cells are ordered along the axes and the grid contains $\boldsymbol{N}$ cells (i.e. $\boldsymbol{N}_x$ marks the number of cells in the x direction). Further suppose $\boldsymbol{b}_{min}$ marks the boundary corner of the grid with the lowest co-ordinates. Then a particle at position x_i is inside the grid cell at index

$$idx(\boldsymbol{x}_i) = \text{clamp}_{vec}(\tfrac{1}{h}(\boldsymbol{x} - \boldsymbol{b}_{min}), \boldsymbol{0}, \boldsymbol{N} - \boldsymbol{1}),$$

where $\boldsymbol{0} = (0,0,0)$ and $\boldsymbol{1} = (1,1,1)$, and the clamp_{vec} function is a coordinate-wise clamp:

$$\text{clamp}(a,b,c) := \min(\max(a,b),c).$$

This ensures that an infinite domain is covered by the grid where the boundary cells are extended to infinity. This indexing is especially useful when testing boundary conditions, because particles passing through the boundary can be visually examined.

Now given a particle i, its neighbourhood particles can be found by iterating through the particles of each of the 9 cells neighbouring $idx(\boldsymbol{x}_i)$. Note that other cells cannot contain any neighbours since cell side length is h.

An important factor for performance in SPH implementations is the number of loops through all the particles needed to complete a time step. Each additional loop drastically hinders performance, so researchers try to minimize loops in their models as much as possible. Naturally, having multiple interacting fluids may suggest the need to loop for each fluid, however storing an identifier at a particle and determining which fluid it belongs to to gather physical parameter is significantly faster.

For additional neighbourhood search optimizations, see [Ihmsen et al. 2014a].

3.3 Incompressibility

As mentioned in the introduction, there are various methods to handle incompressibility in SPH fluids. I implemented two non-iterative EOS models, and I will describe in detail the method proposed in [Ihmsen et al. 2014b], although no examples are present.

Pressure in the fluid can be directly related to density via the ideal gas law equation: $p = c_s \rho$, where c_s is a compressibility constant, which physically corresponds to the speed of sound through the fluid. As suggested in [Desbrun and Gascuel 1996] and later used in [Müller et al. 2003], this equation is replaced by

$$p = c_s(\rho - \rho_0) \tag{5}$$

where ρ_0 is the rest density of the fluid. This formation is better suited for fluids, since particles will tend occupy a constant volume and be generally more evenly distributed through the fluid. Müller et al. [2003] used the following symmetrization of accelerations due to pressure forces:

$$\boldsymbol{a}_i^{\text{pres}} = -\frac{\nabla p_i}{\rho_i} = -\sum_j m_j \frac{p_i + p_j}{2\rho_i \rho_j} \nabla W_{ij}$$

Alternatively, Monaghan [1994] suggested using the Tait equation (also used in [Becker and Teschner 2007]), particularly,

$$p = \frac{\rho_0 c_s}{\gamma} \left(\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right) \tag{6}$$

where $\gamma = 7$ is often chosen. This formulation is best suited for weakly compressible fluids, where compression is penalized more aggressively. Becker and Teschner [2007] demonstrated a comparison of this EOS to (5), showing less compressibility when using (6). They used the symmetrization for acceleration as in (4):

$$\boldsymbol{a}_i^{\text{pres}} = -\frac{\nabla p_i}{\rho_i} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij} \right) \nabla W_{ij} \tag{7}$$

where Π_{ij} is a symmetric damping term discussed in Section 3.4 below.

In both cases, negative pressures must be clamped to 0. Negative pressures may be interpreted as surface tension, however the resulting attraction forces are rather exaggerated and can lead to large oscillations if ignored. Both equations of state above are implemented as described.

3.4 Viscosity

A canonical artificial viscosity, originally introduced in [Monaghan and Gingold 1983], is still widely used in fluid SPH simulations [Becker and Teschner 2007; Becker et al. 2009; Akinci et al. 2012].

In particular, equation (7) is damped by a viscous term Π_{ij}, given by when

$$\Pi_{ij} = \begin{cases} -\nu\left(\frac{\boldsymbol{v}_{ij}\cdot\boldsymbol{x}_{ij}}{|\boldsymbol{x}_{ij}|^2+0.01h^2}\right) & \boldsymbol{v}_{ij}\cdot\boldsymbol{x}_{ij} < 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\boldsymbol{x}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$, $\boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j$, and the viscous term

$$\nu = \frac{2\alpha h c_s}{\rho_i + \rho_j}.$$

Here α is the user controlled viscosity parameter.

Another popular viscosity is given in [Müller et al. 2003]:

$$\boldsymbol{a}_i^{\text{visc}} = \nu \sum_j m_j \frac{\boldsymbol{v}_j - \boldsymbol{v}_i}{\rho_i \rho_j} \nabla^2 W_{ij} \tag{8}$$

Both viscosities above are implemented as described.

3.5 Surface Tension

Surface tension effects are difficult to achieve in SPH fluids. This is because many models (e.g. [Morris 2000; Müller et al. 2003]) rely on curvature of the fluid surface, but computing curvature on a body of particles with no well defined surface is non-trivial.

Some SPH surface tension models like [Morris 2000; Müller et al. 2003] are derived from the continuum surface force method [Brackbill et al. 1992]. Here the surface tension minimizes surface area of the fluid. In particular, the force per unit area on a particle is given by

$$\boldsymbol{f}^{\text{surf}} = \sigma\kappa\hat{\boldsymbol{n}}, \tag{9}$$

where κ is the curvature, $\hat{\boldsymbol{n}}$ is the normal to the surface, and σ is the user controlled surface tension coefficient. In order to compute the curvature, we must first define a discrete field c over the particles, called the *colour field* in SPH literature. As proposed in [Müller et al. 2003], for a single fluid, we may define c to be 1 at particle positions and 0 everywhere else. Then the normal can be written as

$$\hat{\boldsymbol{n}} = \frac{\nabla c}{|\nabla c|}.$$

The idea is that the colour field changes rapidly at the interface. Thus curvature can be computed as

$$\kappa = -\nabla \cdot \hat{\boldsymbol{n}}$$

Additionally [Müller et al. 2003] proposes to multiply (9) by $|\nabla c|$ to ensure that tension is computed at the interface only. In addition, they suggest evaluating (9) when $|\nabla c|$ is large enough as to not cause any stability problems. Thus the force per unit mass is can be written as

$$\boldsymbol{a}^{\text{surf}} = -\frac{\sigma}{\rho}\nabla^2 c\,\hat{\boldsymbol{n}}$$

This equation can be discretized as usual.

One problem with the above model is that computing $\nabla^2 c$ is rather error prone, especially for low surface tensions, where particles may be scattered.

Following the work of [Tartakovsky and Meakin 2005], Becker and Teschner [2007] use a position based attractive force to model surface tension:

$$\boldsymbol{a}_i^{\text{surf}} = -\frac{\sigma}{m_i}\sum_j m_j(\boldsymbol{x}_i - \boldsymbol{x}_j)W_{ij} \tag{10}$$

This formulation is especially simple and gives visually convincing results.

Other models have been developed [Morris 2000; Hu and Adams 2006], however they focus on multi-phase flow and give poor results when applied to free-surface flows (see Figure 2 or [Becker and Teschner 2007] for comparison).

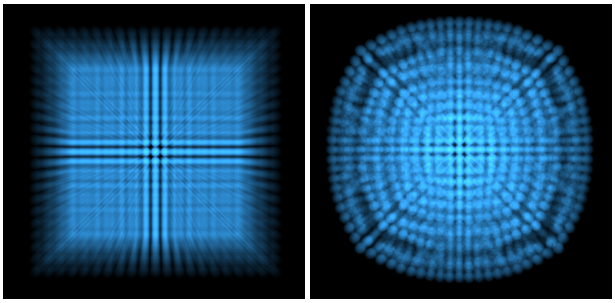

Figure 2: *A cube was evolved with high viscosity and surface tension similar to the test in Becker and Teschner [2007]. Note that the additive blending reveals low density areas near the surface in the right image (evolved state).*

3.6 Multi-phase Fluids

As mentioned in the introduction, SPH is naturally suited for multi-phase fluid flow. However, there are some issues with the standard SPH formulations, when it comes to fluid-fluid coupling with large density ratios. [Solenthaler and Pajarola 2008] has addressed this issue. In particular, the standard SPH computation of density given in equation (3) coupled with a typical EOS, causes large repulsion forces between neighbouring fluids with a large density ratio. If, however, the mass of the current particle was used to compute the density, this problem can be avoided. This is because on the boundary between two fluids, each fluid feels pressure as if it was due to its own particles. Thus, we compute $\rho_i = m_i \sum_j W_{ij}$ instead of (3). This idea is extended to other SPH quantities in [Solenthaler and Pajarola 2008], and demonstrated multi-phase fluids using the models of [Müller et al. 2003] and [Monaghan 1992]. These ideas are used in my implementation for multi-phase flow including pressure, viscosity and surface tension forces for the model of [Müller et al. 2003]. In order to model tension at the free surface I used the model in (10) instead of [Morris 2000] as otherwise suggested in [Solenthaler and Pajarola 2008]. Furthermore, my implementation of [Becker and Teschner 2007] uses a pressure force per unit mass given by

$$\boldsymbol{a}_i^{\text{pres}} = -\sum_j (V_i^2 p_i + V_j^2 p_j + m_j \Pi_{ij})\nabla W_{ij}$$

where $V_i = \frac{m_i}{\rho_i}$ is the volume. The artificial viscosity term left unchanged. Finally, surface tension is computed only between particles of the same phase.

3.7 Boundary Conditions

A variety of boundary conditions have been used in SPH. I will describe two particular strategies and make a note on the boundary model used in [Müller et al. 2003].

3.7.1 Impulse Response Boundaries

Müller et al. [2003] used impulse based boundary collisions. When a boundary-particle collision is detected, the velocity of the particle

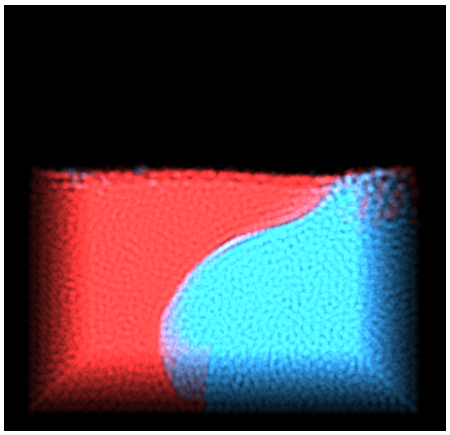

Figure 3: *A fluid with density 1000 kg/m^2 (blue) flows underneath a lighter fluid with density 900 kg/m^2 (red).*

is reflected along the normal at the boundary. In my implementation of [Müller et al. 2003], to simulate inelastic collisions with the boundary, the reflected speed was reduced to 10% of the speed before collision. To prevent particles from penetrating the boundary, if the particle passes the boundary during a simulation step, its position is projected to the boundary. To avoid sticking effects the particle is also pushed a distance of 0.001 away from the boundary in the direction of the normal.

3.7.2 Penalty Force Boundaries

Monaghan et al. [2004] noted that boundary force opposes the pressure gradient of the fluid, hence it is natural to use the gradient of the smoothing kernel to model boundary forces. In particular, the force per unit mass on fluid particle i due to boundary particle k is given by

$$\boldsymbol{f}_{ik} = \frac{m_k}{m_i + m_k}\Gamma(y)\chi(x)\boldsymbol{n}_k,$$

where $\boldsymbol{n}_k$ is the unit normal to the boundary at particle k, y is the normal distance of the fluid particle to the boundary particle, that is $y = \boldsymbol{x}_{ik} \cdot \boldsymbol{n}_k$, and x is the tangential distance $x = \sqrt{|\boldsymbol{x}_{ik}|^2 - y^2}$. Thus the total boundary force per unit mass on a fluid particle i is

$$\boldsymbol{a}_i^{\text{bdry}} = \sum_k \boldsymbol{f}_{ik}. \tag{11}$$

With boundary particle spacing Δ, the factor $\chi(x)$, written as

$$\chi(x) = \begin{cases} \left(1 - \frac{x}{\Delta}\right) & 0 \le x < \Delta \\ 0 & \text{otherwise,} \end{cases}$$

ensures that a particle moving parallel to the boundary feels the same force regardless of where it is between two boundary particles. Note that $\chi(x)$ is designed for 1D boundaries (2D SPH), a more sophisticated heuristic is needed for 2D boundaries. $\Gamma(y)$ takes the form of the kernel[1] gradient, but taking a constant value between its two local extrema:

$$\Gamma(y) = \beta \begin{cases} \frac{1}{3} & 0 \le q < \frac{1}{3} \\ 2q - 3q^2 & \frac{1}{3} \le q < \frac{1}{2} \\ (1-q)^2 & \frac{1}{2} \le q < 1 \\ 0 & \text{otherwise} \end{cases}$$

[1]The kernel is assumed to take the form of a cubic spline, see Appendix A for an explicit formula.

where $q = y/h$, and $\beta = 0.04c_s^2/y$ is chosen to estimate the maximum force per unit mass required to stop a particle moving at the estimated maximum speed. The factor $1/y$ ensures that a faster moving particle will also be stopped. [Monaghan et al. 2004] also observed that their choice of the boundary force made little difference in the final results.

A major flaw in this formulation is that the extra $1/y$ factor dominates $\boldsymbol{f}_{ak}$ as soon as the fluid particle comes to within half of the kernel radius of the boundary particle, where the penalty forces can be arbitrarily large. This greatly restricts time step size, and otherwise causes large spontaneous collision responses. This method models only free-slip conditions, additional tangential forces are required to model different slip conditions.

Becker and Teschner [2007] used very similar boundary conditions, however instead of using static boundary particles, they mirrored fluid particles on the boundary such that x is always zero, and thus avoiding a χ type correction in a 3D setting. A precise derivation for β isn't provided in SPH literature, so I provide a sample derivation on a simplified model in Appendix B for completeness.

3.7.3 Boundary Particles

Akinci et al. [2012] proposed a method for rigid-fluid coupling in SPH fluids. This method leverages the ideas of [Solenthaler and Pajarola 2008] for multi-phase flow mentioned in Section 3.6. Boundary particles are treated like fluid particles, which inherit their velocities from the rigid objects they represent. In particular, the total force per unit mass on a fluid particle i due to a boundary particle k is given by

$$\boldsymbol{a}_{ik}^{\text{bdry}} = \underbrace{-V_k\rho_0\left(\frac{p_i}{\rho_i^2}\right)\nabla W_{ik}}_{\text{pressure acceleration}} \underbrace{- V_k\rho_0\tilde{\Pi}_{ik}\nabla W_{ik}}_{\text{viscosity acceleration}}$$

where $\tilde{\Pi}_{ij}$ is the same as Π_{ij} from before, but with a new viscous factor given by

$$\nu = \frac{\alpha h c_s}{2\rho_i}.$$

Thus the total acceleration due to boundary forces is

$$\boldsymbol{a}_i^{\text{bdry}} = \sum_k \boldsymbol{a}_{ik}^{\text{bdry}}.$$

This method has been implemented to replace the penalty forces proposed by [Becker and Teschner 2007], and demonstrated significant improvements. In figure 5, penalty forces cause the fluid to bend away from the enclosing box boundary right-hand image, which does not occur with particle based boundaries. A closer comparison is shown in Figure 4

4 Time Integration

A standard leap-frog time integration scheme was used as described in [Desbrun and Gascuel 1996]. No other time integration schemes were tested or implemented.

5 Implementation and Visualization

An OpenGL interface was developed in C++11 with the help of Qt 5.2 [Dig 2013] in order to visualize the SPH particles. Various additional C++ libraries were used including `Libconfig` [Lib 2013], `Boost` [Boo 2013], `Assimp` [Ass 2013] and `Eigen` [Eig 2013]. In addition, the program generates output containing information about positions and velocities of an SPH fluid scene for each frame.

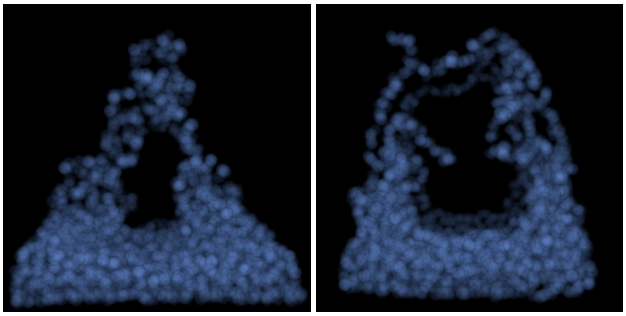

Figure 4: *Penalty forces push the splash into the middle (left image), while boundary particles maintain a more natural shape of the splash (right image).*

This output has the same format as the input for a particle skinner implementation from [Bhattacharya et al. 2011]. Thus, one may couple the two programs to generate triangular surfaces ready for rendering.

My implementation uses the libraries `Assimp` and `Libconfig` to let the user dynamically load geometry files describing the scene and dynamics to be used.

Figure 5: *A path traced image of a liquid ball splashing inside a cube box. This image is the result of my implementation of the model proposed by Becker and Teschner [2007]. Rendering was accomplished with Blender.*

6 Results and Conclusions

The focus of this project was to explore and learn about strengths and weaknesses of SPH used in fluid simulation. As a result, little attention was directed towards generating visually appealing imagery using one particular method. Instead, many methods were explored.

I have successfully implemented two SPH formulations presented in [Müller et al. 2003] and [Becker and Teschner 2007], although I could not establish a working version of an iterative pressure projection solver, such as the one found in [Ihmsen et al. 2014b]. In addition, I was able to simulate various boundary conditions and multi-phase fluid interactions. Currently, the code used to handle multi-phase interactions is able to simulate fluids implemented with drastically different methods, for instance a fluid governed by equations from [Müller et al. 2003] may interact with a fluid governed by the equations from [Becker and Teschner 2007]. This functionality is useful because different models are preferred for different types of fluid: EOS type models handle compressible fluids better, while PPE methods are better at simulating incompressibility. I have not rigorously tested this feature.

All simulations were done on a 2011 MacBook Air equipped with a 1.8 GHz Intel Core i7 CPU, 4GB of RAM and Intel HD Graphics 3000 (384 MB). The simulations shown in the video are read from cached files.

A particularly interesting drawback of both implemented models is that with a large amount of fluid, the surface of the fluid oscillates when at rest. This is possibly the result of poor viscosity models used, but most likely simply an EOS solver artifact.

Figures 1 and 3 demonstrate multiphase fluid interaction. In Figure 1 the orange shape has a lower density, and floats relative to the blue (higher density) fluid. Both figures show fluids governed by equations from [Becker and Teschner 2007].

The supported video contains a series of scenes (simulations). The first two simulations demonstrate two [Müller et al. 2003] fluids with varying densities. The second, has a higher compressibility constant and gives better results with less compression. The following two clips also demonstrate multi-phase flow using both implemented models. The next clip shows the surface test as in Figure 2, and the following clip demonstrates surface tension in a fluid subject to gravity. The final clip shows two breaking dams of different density fluids. Note the colour coded information given on the screen, it gives more precise information about each fluid on screen when being simulated. Also note that the video shows precomputed simulations, actual simulations may take up to 20 seconds per frame.

7 Future Work

Firstly, as hinted before, I would like to demonstrate a practical use of multi-phase interactions between different SPH models, where perhaps a gas-liquid coupling is effectively achieved. Furthermore, it would be particularly interesting to see how IISPH works with various viscosity, surface tension and multi-phase interaction models.

Acknowledgements

To Christopher Batty, for all the doughnuts.

References

ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively sampled particle fluids. In *ACM SIGGRAPH 2007 Papers*, ACM, New York, NY, USA, SIGGRAPH '07.

AKINCI, N., IHMSEN, M., AKINCI, G., SOLENTHALER, B., AND TESCHNER, M. 2012. Versatile rigid-fluid coupling for incompressible sph. *ACM Trans. Graph. 31*, 4 (July), 62:1–62:8.

AKINCI, N., AKINCI, G., AND TESCHNER, M. 2013. Versatile surface tension and adhesion for sph fluids. *ACM Trans. Graph. 32*, 6 (Nov.), 182:1–182:8.

2013. *Assimp: Open Asset Import Library*.

BALL, P. 2011. *Nature's patterns : a tapestry in three parts*. Oxford University Press, Oxford New York.

BECKER, M., AND TESCHNER, M. 2007. Weakly compressible sph for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '07, 209–217.

BECKER, M., TESSENDORF, H., AND TESCHNER, M. 2009. Direct forcing for lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics 15*, 3 (May), 493–503.

BHATTACHARYA, H., GAO, Y., AND BARGTEIL, A. W. 2011. A level-set method for skinning animated particle data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

BONET, J., AND KULASEGARAM, S. 2002. A simplified approach to enhance the performance of smooth particle hydrodynamics methods. *Applied Mathematics and Computation 126*, 23, 133 – 155.

2013. *Boost 1.55, general-purpose C++ library*.

BRACKBILL, J., AND RUPPEL, H. 1986. Flip: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics 65*, 2, 314 – 343.

BRACKBILL, J., KOTHE, D., AND ZEMACH, C. 1992. A continuum method for modeling surface tension. *Journal of Computational Physics 100*, 2, 335 – 354.

CARLSON, M., MUCHA, P. J., AND TURK, G. 2004. Rigid fluid: Animating the interplay between rigid bodies and fluid. In *ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, SIGGRAPH '04, 377–384.

CHANIOTIS, A., POULIKAKOS, D., AND KOUMOUTSAKOS, P. 2002. Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows. *Journal of Computational Physics 182*, 1, 67 – 90.

DESBRUN, M., AND GASCUEL, M.-P. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation 96*, R. Boulic and G. Hgron, Eds., Eurographics. Springer Vienna, 61–76.

DIGIA. 2013. *Qt 5.2 UI framework*.

2013. *Eigen, C++ linear algebra library*.

FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 15–22.

FELDMAN, J., AND BONET, J. 2007. Dynamic refinement and boundary contact forces in sph with applications in fluid flow problems. *International Journal for Numerical Methods in Engineering 72*, 3, 295–324.

FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '01, 23–30.

FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graphical models and image processing 58*, 5, 471–483.

FOSTER, N., AND METAXAS, D. 1997. Controlling fluid animation. In *Computer Graphics International, 1997. Proceedings*, IEEE, 178–188.

GINGOLD, R. A., AND MONAGHAN, J. J. 1977. Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly notices of the royal astronomical society 181*, 375–389.

GUENDELMAN, E., SELLE, A., LOSASSO, F., AND FEDKIW, R. 2005. Coupling water and smoke to thin deformable and rigid shells. In *ACM SIGGRAPH 2005 Papers*, ACM, New York, NY, USA, SIGGRAPH '05, 973–981.

HARLOW, F. H., AND WELCH, J. E. 1965. Numerical calculation of time?dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids (1958-1988) 8*, 12, 2182–2189.

HARLOW, F. H. 1962. *THE PARTICLE-IN-CELL METHOD FOR NUMERICAL SOLUTION OF PROBLEMS IN FLUID DYNAMICS*. Mar.

HE, X., LIU, N., LI, S., WANG, H., AND WANG, G. 2012. Local poisson sph for viscous incompressible fluids. *Computer Graphics Forum 31*, 6, 1948–1958.

HU, X., AND ADAMS, N. 2006. A multi-phase sph method for macroscopic and mesoscopic flows. *Journal of Computational Physics 213*, 2, 844 – 861.

IHMSEN, M., ORTHMANN, J., SOLENTHALER, B., KOLB, A., AND TESCHNER, M. 2014. Sph fluids in computer graphics. *State-of-the-Art Report, Eurographics*.

IHMSEN, M., CORNELIS, J., SOLENTHALER, B., HORVATH, C., AND TESCHNER, M. 2014. Implicit incompressible sph. *IEEE Transactions on Visualization and Computer Graphics 20*, 3, 426–435.

KOTHE, D., AND BRACKBILL, J. 1992. Flip-inc: a particle-in-cell method for incompressible flows. *Unpublished manuscript*.

2013. *Libconfig, C/C++ configuration file library*.

LOSASSO, F., SHINAR, T., SELLE, A., AND FEDKIW, R. 2006. Multiple interacting liquids. *ACM Trans. Graph. 25*, 3 (July), 812–819.

LUCY, L. B. 1977. A numerical approach to the testing of the fission hypothesis. *The astronomical journal 82*, 1013–1024.

MONAGHAN, J., AND GINGOLD, R. 1983. Shock simulation by the particle method sph. *Journal of Computational Physics 52*, 2, 374 – 389.

MONAGHAN, J., KOS, A., AND ISSA, N. 2004. Fluid motion generated by impact. *Journal of waterway, port, coastal, and ocean engineering 129*, 6, 250–259.

MONAGHAN, J. J. 1992. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics 30*, 543–574.

MONAGHAN, J. 1994. Simulating free surface flows with sph. *Journal of Computational Physics 110*, 2, 399 – 406.

MONAGHAN, J. J. 2005. Smoothed particle hydrodynamics. *Reports on progress in physics 68*, 8, 1703.

MORRIS, J., AND MONAGHAN, J. 1997. A switch to reduce {SPH} viscosity. *Journal of Computational Physics 136*, 1, 41 – 50.

MORRIS, J. P. 2000. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids 33*, 3, 333–353.

MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on*

Computer Animation, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '03, 154–159.

OSHER, S., AND SETHIAN, J. A. 1988. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics 79*, 1, 12–49.

PARZEN, E. 1962. On estimation of a probability density function and mode. *Annals of mathematical statistics 33*, 3, 1065–1076.

RASMUSSEN, N., ENRIGHT, D., NGUYEN, D., MARINO, S., SUMNER, N., GEIGER, W., HOON, S., AND FEDKIW, R. 2004. Directable photorealistic liquids. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '04, 193–202.

ROBINSON-MOSHER, A., SHINAR, T., GRETARSSON, J., SU, J., AND FEDKIW, R. 2008. Two-way coupling of fluids to rigid and deformable solids and shells. In *ACM SIGGRAPH 2008 Papers*, ACM, New York, NY, USA, SIGGRAPH '08, 46:1–46:9.

ROSENBLATT, M. 1956. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics 27*, 3, 832–837.

SCHECHTER, H., AND BRIDSON, R. 2012. Ghost sph for animating water. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 4.

SHAO, S., AND LO, E. Y. 2003. Incompressible {SPH} method for simulating newtonian and non-newtonian flows with a free surface. *Advances in Water Resources 26*, 7, 787 – 800.

SIDE EFFECTS INC. 2013. *Houdini 3D Animation Tool*.

SOLENTHALER, B., AND GROSS, M. 2011. Two-scale particle simulation. In *ACM SIGGRAPH 2011 Papers*, ACM, New York, NY, USA, SIGGRAPH '11, 81:1–81:8.

SOLENTHALER, B., AND PAJAROLA, R. 2008. Density contrast sph interfaces. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 211–218.

SOLENTHALER, B., AND PAJAROLA, R. 2009. Predictive-corrective incompressible sph. In *ACM SIGGRAPH 2009 Papers*, ACM, New York, NY, USA, SIGGRAPH '09, 40:1–40:6.

STAM, J. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 121–128.

TARTAKOVSKY, A., AND MEAKIN, P. 2005. Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Phys. Rev. E 72* (Aug), 026301.

A Smoothing Kernels

The following smoothing kernels were used in the examples presented. Note that $q = |\boldsymbol{x}|/h$. The main kernel used throughout literature as described in [Monaghan 2005] is the cubic spline kernel, given with its derivatives:

$$W_{spline}(\boldsymbol{x},h) := \frac{16}{\pi h^3}\begin{cases} 3q^3-3q^2+\frac{1}{2} & 0\le q<\frac{1}{2} \\ (1-q)^3 & \frac{1}{2}\le q<1 \\ 0 & q\ge 1 \end{cases}$$

$$\nabla W_{spline}(\boldsymbol{x},h) := \frac{48}{\pi h^5}\boldsymbol{x}\begin{cases} (3q-2) & 0\le q<\frac{1}{2} \\ -\frac{(1-q)^2}{q} & \frac{1}{2}\le q<1 \\ 0 & q\ge 1 \end{cases}$$

$$\nabla^2 W_{spline}(\boldsymbol{x},h) := \frac{96}{\pi h^5}\begin{cases} 3(2q-1) & 0\le q<\frac{1}{2} \\ (2q-1)\frac{(1-q)}{q} & \frac{1}{2}\le q<1 \\ 0 & q\ge 1 \end{cases}$$

Müller et al. [2003] proposed alternative kernels to improve performance and correct some side effects seen in standard SPH simulations. The following is a general purpose smoothing kernel used to replace the cubic spline above for performance reasons (no square root computations needed):

$$W_{poly6}(\boldsymbol{x},h) := \frac{315}{64\pi h^3}\begin{cases} (1-q^2)^3 & 0\le q<1 \\ 0 & q\ge 1 \end{cases}$$

$$\nabla W_{poly6}(\boldsymbol{x},h) := \frac{945}{32\pi h^5}\boldsymbol{x}\begin{cases} -(1-q^2)^2 & 0\le q<1 \\ 0 & q\ge 1 \end{cases}$$

$$\nabla^2 W_{poly6}(\boldsymbol{x},h) := \frac{945}{32\pi h^5}\begin{cases} (1-q^2)(7q^2-3) & 0\le q<1 \\ 0 & q\ge 1 \end{cases}$$

The following kernel was used in [Müller et al. 2003] for pressure force computations, to prevent clusters forming at high pressures:

$$W_{spiky}(\boldsymbol{x},h) := \frac{15}{\pi h^3}\begin{cases} (1-q)^3 & 0\le q<1 \\ 0 & q\ge 1 \end{cases}$$

$$\nabla W_{spiky}(\boldsymbol{x},h) := \frac{45}{\pi h^5}\boldsymbol{x}\begin{cases} -\frac{(1-q)^2}{q} & 0<q<1 \\ 0 & q\ge 1 \end{cases}$$

$$\nabla^2 W_{spiky}(\boldsymbol{x},h) := \frac{90}{\pi h^5}\boldsymbol{x}\begin{cases} (1-q)(2-\frac{1}{q}) & 0<q<1 \\ 0 & q\ge 1 \end{cases}$$

Mind the singularity at $\boldsymbol{x}=\boldsymbol{0}$ in ∇W_{spiky} and $\nabla^2 W_{spiky}$.

B Penalty-Force Boundary Conditions

Here I derive the penalty force function found similar to [Becker and Teschner 2007]. Suppose the fluid has an initial velocity $\boldsymbol{v}_0$ and is subject to gravitational acceleration $\boldsymbol{g}$. Further assume that the fluid is bound in an axis aligned box with dimensions $\boldsymbol{d} = (d_x, d_y, d_z)$. Then from conservation of energy, we can estimate the maximum speed of a fluid particle to be

$$v_f = \sqrt{2\boldsymbol{g}d_y + |\boldsymbol{v}_0|^2}.$$

Assume that boundary particles are arbitrarily more massive compared to fluid particles, and fluid particles interact with a single boundary particle inserted during collision resolution. Then the penalty force per unit mass on each fluid particle is modelled by

$$\frac{d\boldsymbol{v}_a}{dt} = \Gamma(y)\boldsymbol{n}$$

where $\boldsymbol{n}$ is the normal of the collision boundary. Then the work required to stop a particle moving at the maximum estimated speed is given by

$$\int_0^h \frac{d\boldsymbol{v}_a}{dt} \cdot \boldsymbol{n} dy = \frac{1}{2} v_f^2.$$

Solving this for β yields

$$\beta = \frac{27 v_f^2}{11 h}.$$

Adding the extra $1/y$ factor can ensure that boundaries are never penetrated given small enough time steps, however other measures are possible, such as clamping particle positions to the boundaries insuring non-penetration.