Rotation Project : Simulating fluids with surface tension

Manas Bhargava
manas.bhargava@ist.ac.at
Institute of Science and Technology, Austria

Supervised by - Prof. Scott Waitukaitis
scott.waitukaitis@ist.ac.at
Institute of Science and Technology, Austria

Figure 1: A spherical droplet of radius 4 cm is dropped onto the ground.

1 INTRODUCTION

1.1 Motivation

Researchers in computer graphics and physics often take a different approach while tackling a problem. Computer graphics researchers are more interested in simulating cool physical phenomenon, having control over the simulation to add artistic effects, and most importantly being computationally efficient so that the simulation can be scaled for production in the animation and gaming industry. On the other hand, physicists are more interested in making the simulation match with real-world behavior so that they have a better understanding of the physical model governing the behavior of the particular physical phenomenon. In this rotation project, we try to bridge the gap between the two disciplines and try to utilize the recent development in fluid simulation to simulate fluid close to the real-world behavior. We implemented the recent Affine Particle in Fluid (APIC) [Jiang et al. 2015] method to simulate fluid motion. We also introduce the sign distance field computation to reconstruct the surface mesh from the simulation. The sign distance field formulation was later used to augment our fluid simulation with a physically motivated surface tension model.

2 BACKGROUND

2.1 Fluid simulation

Fluid simulation is an actively researched field in both computation fluid dynamics and computer graphics. The seminal work of Harlow and Welch [Harlow and Welch 1965] introduced an efficient way of simulating fluids using staggered grids in the marker and cell method. Stable fluids [Stam 1999] introduced operator splitting to solve the Navier Stoke's equation thereby providing an efficient method of fluid simulation to the computer graphics community. This opened up the possibility of improving this method by making it more stable, less dissipative, and incorporating different physical models. Recently in 2015, Jiang et.al [Jiang et al. 2015] introduced the affine particle in cell (APIC) method for fluid simulation. Figure 2 showcases how APIC outperforms the previous state of the art

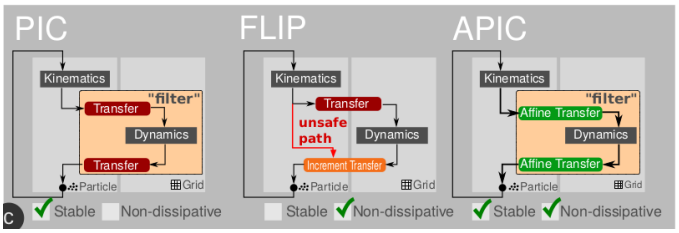

Figure 2: The figure showcases the benefits of using APIC over PIC and FLIP methods. Source - [Jiang et al. 2015]

methods particle in cell method (PIC) and fluid implicit particle in cell method (FLIP) by being both stable and non-dissipative.

2.2 Sign Distance Field

In simple terms, the sign distance field represents the distance of every point on the grid to its closest point on the surface. Thus for a given closed set of points P on the surface, the signed distance function $\phi_s(\vec{x})$is given by -

$$distance_s(\vec{x}) = \min_{p \in P} ||\vec{x} - \vec{p}|| \tag{1}$$

$$\phi_s(\vec{x}) = \begin{cases} distance_s(\vec{x}) & : \vec{x} \text{ is outside} \\ -distance_s(\vec{x}) & : \vec{x} \text{ is inside} \end{cases} \tag{2}$$

The sign distance field has several applications in fluid simulation. It helps in reconstructing mesh surfaces thus providing realistic animations. It improves the pressure projection solve discussed in Section 5. The sign distance function posses some interesting properties and helps in computing normal vector and the mean curvature values at the surface. These properties play an important role in formulating the surface tension force in the fluid.

2.3 Surface Tension

The initial work in the graphics community was focused on simulating water drops. They used particle-systems or spring-mass system

to model the droplets. These methods worked efficiently but lacked the physical motivations behind the simulations and were thus limited in their capabilities to simulate diverse physical phenomena. After the introduction of stable fluids in 1999 by Stam, researchers started augmenting surface tension models into the fluid simulators. Enright [Enright et al. 2003] and Lossaso [Losasso et al. 2004] used Dirichlet pressure boundary conditions on air boundary cells by estimating the mean curvature using the surface's signed distance function to model surface tension which generated promising results. Wang [Wang et al. 2005] later upgraded this method and presented a physically motivated surface tension model for the 3D interfacial surface tensions on arbitrarily curved solid boundaries by enforcing the contact angles.

3 ALGORITHM OVERVIEW

In this report, we will discuss the fluid solver as described in Bridson's book - Fluid Simulation for Computer Graphics [Bridson 2015]. We then present the extension of this work by incorporating the APIC method described in [Jiang et al. 2015] for fluids. We then discuss the implementation of the sign distance field for our surface and use it for better rendering and improved non-voxelized pressure projection solve. Later, we present the surface tension model as introduced by Kang et.al [Kang et al. 2000] and further improved by Enright [Enright et al. 2003]. Finally, we motivate the use of sign distance function to approximate the normal vector and curvature at the surface. We use the ghost air cell method to formulate the forces arising due to surface tension and incorporate it into our fluid simulation.

4 APIC - FLUID SIMULATOR

Our fluid simulator is governed by the Navier Stokes equation given by -

$$\frac{D\vec{u}}{Dt} + \frac{1}{\rho}\nabla P = \vec{F_{ext}} \tag{3}$$

$$\nabla.\vec{u} = 0 \tag{4}$$

Here, u represents the velocity field of the fluid, P is the scalar quantity pressure inside the fluid, ρ is the density of the fluid and F_{ext} represents the external force that is applied on the fluid - gravity, surface tension, and user applied input force.

We use the hybrid particle-grid based method to solve the Navier Stokes equation (without the viscosity term). Instead of solving the entire equation in one step, we perform operator splitting. This allows us to divide the problem into small steps and solve them independently and efficiently. The first term in equation 3 i.e. the material derivative of the velocity field represents the advection step of our simulation. The next step is to apply the external forces and update the velocity field. After this we apply the pressure term (in equation 3) along with the incompressibility condition (equation 4) and solve a Poisson's equation given by -

$$\frac{\Delta t}{\rho}\nabla.\nabla P = \nabla.\vec{u} \tag{5}$$

This step is also referred to as the pressure projection step.

Thus overall our fluid simulator has three main steps - advection, velocity update, and pressure projection solve. Instead of solving this equation in the continuum domain, we discretize our Navier Stokes equation. Traditionally researchers have used either Eulerian (grid-based methods) or Lagrangian (particle-based methods) to solve the discretized Navier Stokes equation. Both of them have their advantages and disadvantages. The Eulerian method performs well in velocity update and pressure projection solve. But the advection step is fairly complicated and requires higher-order integration schemes like Runge-Kutta method or backward Euler methods to better approximate the solution of the differential equation. This still does not solve the problem properly as Runge-Kutta methods are known to be slightly unstable in certain regimes while the backward Euler method damps the entire fluid simulation significantly. On the other hand, the Lagrangian method has a fairly straightforward implementation for advection and external force update but implementing the pressure projection step is fairly complicated and introduces approximation error. To counter this, we use a hybrid particle-grid based method for our simulation as motivated by [Brackbill et al. 1988]. To ensure stability and avoiding dissipation while transferring information between particles and grid we use affine transfers as described in the APIC paper [Jiang et al. 2015].

In this method, the particle represents the principle information about the fluid, i.e. its position and velocity. The background staggered grid is used as a scratch-pad to perform further computations. The grid stores the velocities at its faces while the pressure values are stored at the grid centers. This way of storing values at faces and cell centers was first introduced by [Harlow and Welch 1965] and allows for robust and efficient computation of the discretized gradients in the differential equation.

As the first step, the particle velocities are interpolated onto the grid using affine transfer matrices. Once we have the velocities on the grid, we apply the external force and update our grid velocities. We then apply the pressure projection step, to calculate the pressure values inside the fluid and update the grid velocity to make them divergence-free. To solve the Poisson's equation in the pressure projection step, we use the Modified Incomplete Cholesky Conjugate Gradient (MICCG) method as described in Bridson's book [Bridson 2015]. The divergence-free grid velocity field is thus finally interpolated back to the particles. At this step, we perform the advection step and update the position of the particles using the updated particle velocity. The overall algorithm for the fluid simulation is summarised in figure 3.

5 THE SIGN DISTANCE FIELD

To get an estimate of the surface, we need to implement the sign distance field $\phi(s)$. We create a sphere of radius (r = $2\sqrt{3}$) times the grid spacing around every particle. This radius allows the particle to cover all the grid cells that it affects while transferring the velocities from particle to the grid nodes. Then for every grid node s and set of particles P, we compute $\phi_{temp}(s)$ given by -

$$\phi_{temp}(s) = \min_{p \in P} ||\vec{s} - \vec{p}|| - r \tag{6}$$

All the points that are inside the surface have a negative value of $\phi_{temp}(s)$ while all the outside points have a positive value. Using this information of $\phi_{temp}(s)$ we calculate the boundary points of the fluids whose neighbors lie outside the surface. For these boundary points we linearly interpolate the $\phi_{temp}(s)$ value to the surface which has $\phi(s)$ value equal zero. This allows us to compute

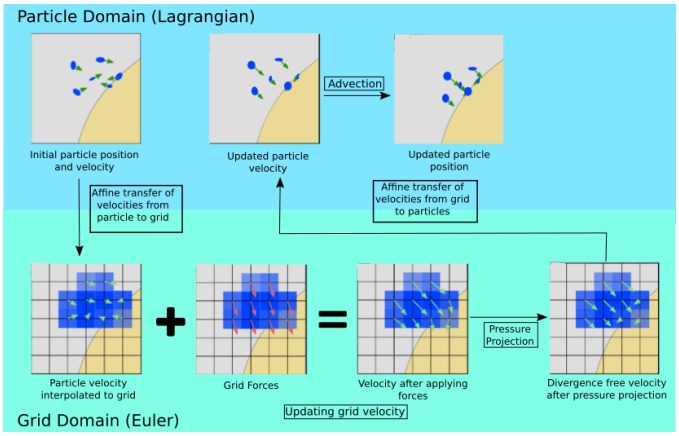

Figure 3: Overview of the Affine particle in cell (APIC) method for fluid simulation.

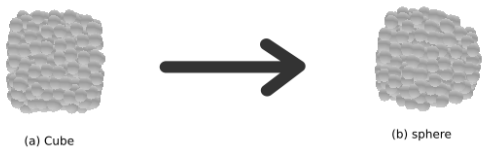

Figure 4: A cube of length 0.6 mm converts to a shape of sphere to minimize the surface area as a result of the surface tension force applied to it.

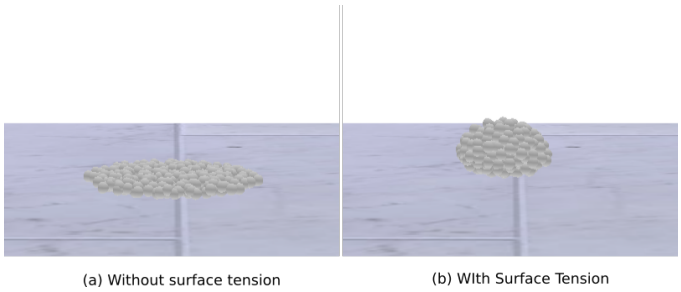

Figure 5: A sphere droplet of radius 0.3 mm is allowed to settle on the ground. Surface tension is absent in figure (a) and is taken into account in figure (b).

the $\phi(s)$ value for all the boundary nodes. After obtaining the $\phi(s)$ for all the boundary nodes, we use the fast sweeping method as described by Zhao in [Zhao 2005] to estimate the $\phi(s)$ for all the grid nodes.

The sign distance field thus computed is then used to enhance our fluid simulator. It is used in the implementation of enhanced pressure projection solve, making it more robust and makes the rendering less voxelized. The procedure was used in the Ghost fluid method by Gibou et al. [Nguyen et al. 2002] and is explained in Bridson's book [Bridson 2015].

We also utilize the sign distance field to create the surface mesh and thus improving our rendering. For this, we apply the Marching cubes method [Lorensen and Cline 1987] to generate 3D surface reconstruction from the sign distance field. Figure 1 showcase the rendering of liquid droplet splashing on the ground when dropped from a height incorporating both enhanced pressure projection and surface mesh rendering.

6 SURFACE TENSION

The sign distance field also plays an important role in implementing the surface tension model. The force due to surface tension F_{st} at the boundary of the surface is given by -

$$\vec{F_{st}} = -\frac{\gamma\kappa.\vec{N}}{\rho} \tag{7}$$

where ρ is the fluid density , γ is the surface tension constant (0.072 N/m for water), κ is the radius of curvature at the surface given by $\nabla\left(\frac{\nabla(\phi(s))}{|\nabla(\phi(s))|}\right)$ and N is the unit normal vector at the surface given by $\frac{\nabla(\phi(s))}{|\nabla(\phi(s))|}$.

We apply this surface tension force as an external force at the boundary cells before the pressure projection solve. We conduct two experiments showcasing our surface tension model. In the first experiment, figure 4 we showcase the evolution of a cubical surface to a spherical surface in a gravity-free space. The surface tension force tries to minimize the total surface area and thus changes the shape from a cube to a sphere.

In the second experiment - figure 5 we compare the effect of surface tension on the water droplet settled onto the ground. Without the surface tension force, we see the water droplet flattens on the ground (figure 5 (a)). On the other hand, in the presence of surface tension force (figure 5 (b)), we notice a finite thickness of water particles above the surface. The surface tension force allows the water droplet to maintain its curvature instead of flattening out.

7 CONCLUSION AND FUTURE WORK

In this project, we worked on a fluid simulator to produce results close to real-world behavior. We implemented an APIC based fluid solver, which is both stable and non-dissipative. We also implemented the sign distance function to improve our pressure projection solve and allowing us to render surface meshes. Further, we implemented a simplistic surface tension model and showcase the results.

Implementing 3D interfacial surface tensions for curved solid boundaries is a direct extension of the current work and can be incorporated into our fluid solver as future work. This method handles fracture and splitting of particles naturally and thus avoids the singularity problem related to breakage which drastically affects the mesh-based approach. Thus interesting physical phenomena that require breaking of the water surface can be modeled using our fluid solver.

REFERENCES

Jeremiah U Brackbill, Douglas B Kothe, and Hans M Ruppel. 1988. FLIP: a low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications* 48, 1 (1988), 25–38.

Robert Bridson. 2015. *Fluid simulation for computer graphics*. AK Peters/CRC Press.

Doug Enright, Duc Nguyen, Frederic Gibou, and Ron Fedkiw. 2003. Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *ASME/JSME 2003 4th Joint Fluids Summer Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 337–342.

Francis H Harlow and J Eddie Welch. 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids* 8, 12 (1965), 2182–2189.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.

Myungjoo Kang, Ronald P Fedkiw, and Xu-Dong Liu. 2000. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing* 15, 3 (2000), 323–360.

William E Lorensen and Harvey E Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics* 21, 4 (1987), 163–169.

Frank Losasso, Frédéric Gibou, and Ron Fedkiw. 2004. Simulating water and smoke with an octree data structure. In *ACM SIGGRAPH 2004 Papers*. 457–462.

Duc Nguyen, Frédéric Gibou, and Ronald Fedkiw. 2002. A fully conservative ghost fluid method and stiff detonation waves. In *12th Int. Detonation Symposium, San Diego, CA*.

Jos Stam. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 121–128.

Huamin Wang, Peter J Mucha, and Greg Turk. 2005. Water drops on surfaces. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 921–929.

Hongkai Zhao. 2005. A fast sweeping method for eikonal equations. *Mathematics of computation* 74, 250 (2005), 603–627.