INF585 Project Report

SPH Rigid Body Simulation

IAN BLAIR
MIRETTE MOAWAD

MARCH 2024

INF585 COMPUTER ANIMATION

IAN.BLAIR@POLYTECHNIQUE.EDU | MIRETTE.MOAWAD@POLYTECHNIQUE.EDU

1 Introduction

$M \in \mathcal{R}^d$

SPH - Smoothed Particle Hydrodynamics (SPH) is a computational method used in fluid dynamics and solid mechanics to simulate the behavior of fluids and deformable bodies. It represents fluid as a collection of particles, each with mass, position, velocity, and other properties. It is ideal for representing the somewhat chaotic fluidity of fluids, but is difficult to incorporate extraneous rigid bodies into. Rigid bodies, regardless of their implementation, do not have fixed guidelines for their interactive simulation with SPH particles/fields. This is because the SPH particles are harmoniously interacting with each other, and adding new particles which cannot adhere to their field properties throws the system into chaos unless handled properly.

For our project, we wanted to model a basic SPH fluid environment in two dimensions, representing a body of water. We then wanted to start the SPH/rigid body interaction with a basic falling object that would land in the water and force particles away from it. Once that falling object interacted in a way we were satisfied with, we wanted to add a boat to the environment that would float in the water and interact with the 'waves' caused by the falling object. Ideally we would have a scene that we could interact with, respawning the ship and falling objects at our will.

Our project began with the implementation of an SPH, which was (minorly) extended from the lab. We created the property of waves along the surface of the water, modified the color of the water as it moved to represent the whiteness of fast moving water, and of course added interactions with other objects' particles.

We then went on to create a rigid body, which was in fact a collection of particles with a mesh on top, and model its interaction with the water particles. Our work stalled there for a while, as we realized that the body's interaction with the water particles was not so trivial. Working on the rigid body until we were content with its interaction with the water, we added the capacity to interactively modify the object's mass, which we hoped would create more dramatic interactions with the water field.

Our work on the boat was postponed until after we were satisfied with the falling object's interaction… and ultimately proved to be a shortcoming of the project.

2 SPH

Unlike traditional grid-based methods, SPH uses a set of particles to represent the field, where each particle carries properties such as mass, position, velocity, and thermodynamic quantities. These particles move according to the governing equations of fluid dynamics, typically the Navier-Stokes equations.

In an SPH field, the state of the fluid at any point is determined by the properties of the neighbors particles, smoothed over a certain distance using a kernel function. With a discrete set of particles, this function approximates the continuous nature of the field by guaranteeing that a particle's influence decreases with distance. Due to its inherent Lagrangian nature, the method provides advantages when handling complex free-surface flows, large deformation, and interface dynamics. It does this by tracking the motion of fluid parcels through space and time. Because the SPH field concept is flexible and adaptable to different physical scenarios, it is widely used in a wide range of applications, from engineering to astrophysics.

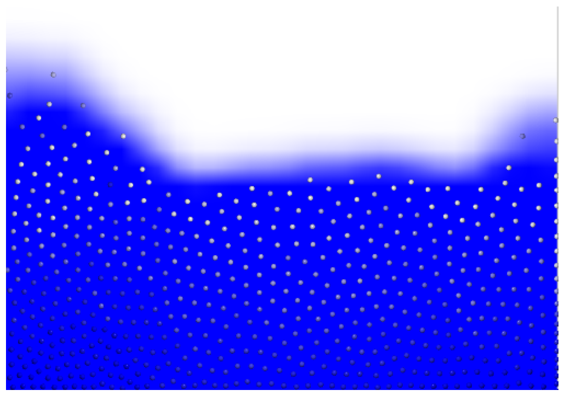

Figure 1: *SPH Fluid Field*

3 Waves

In our project, we simulated the effect of waves on the particles. The algorithm begins by calculating the wave number and angular frequency based on the given wave length and wave speed. In the loop that iterates over all particles, it selectively applies the wave effect to particles that meet a specific condition – in this case, particles with a y position greater than -0.5 which represent the surface particles, we also tried to apply another condition on density as the surface particles have less densities but it wasn't promising.

The wave effect is achieved by adjusting the y position of each particle based on a sinusoidal function, which represents the wave's motion as follows:

$$waveNumber = \frac{2\pi}{waveLength}$$

$$omega = waveNumber \times$$

$$phase = waveNumber \times particles[k].p.x - \omega \times t$$

$$particles[k].p.y += waveAmplitude \times \sin(phase)$$

4 Rigid Body

4.1 Falling Rock

Our idea for the falling object (which we will call a rock) was first to create a single large particle that could interact with all of the SPH particles. However, within the scope of the CGP library, we realized we could actually reuse the same particles that were being used in SPH, modify their behavior, and then draw a single sphere on top of them to represent a rigid body of a rock. To do this, we spawned in 20 new particle elements in the itialize_sph() function in Scene.cpp, put them along the border of a circle, and added them to a new rock_particles array similar to the SPH particles. We gave them constant pressure and rho parameters, although as described later this is slightly unnecessary.

Figure 2: *Rock Particles / Rock Mesh*

In terms of displaying the mesh on top of the particles, we had to ensure that the particles always were aligned in a circle, and then we could just take the center of that circle and draw a cgp::mesh_drawable, specifically a cgp::mesh_primitive_sphere at the center of the circle, which equates to the average position of the rock_particles. Then after every call to simulate, after having updated the positions of the rock particles, we could redraw the rock mesh, updating its position as well.

We also created an interactive checkbox that would decide if the rock should be active at all, checking it for the first time would spawn the rock in the air, and unchecking it would clear the rock from the scene. Moreover we added a button to respawn the rock at the top of the scene.

In terms of the simulated behavior between the rock and the particles, we had some decisions to make. At first we tried to replicate the behavior of the SPH particles for the rock particles, where the rock particles could exert a similar pressure on the water particles as the water particles do on each other. However without letting the rock particles move and fluidly reduce pressure, the behavior was very sporadic and uncontrolled.

$$F = norm(V) * V * 10 * m_r^2 * 1/d$$

Instead what we decided to do was inflict a more classical Newtonian force on the particles that the rock interacted with. We used the rocks mass(m_r), velocity(V), and distance (d) to each water particle to model a simple force on particles within a certain distance, scaled down the further away each water particle was. This worked quite well, and caused some recognizable interactions between a falling rock and water. Notably, there was a void following the falling rock into the water as the water pushed down on the water, which was then replaced by high pressure water that pushed up and in, causing a shot of water to backfire after the rock falls enough. What's more, when the rock was resting on the floor of the scene, the water no longer interacted with the rock, as its velocity was 0. It should be noted that this formula for force isn't perfect, and could be improved upon by factoring in the rock's acceleration.

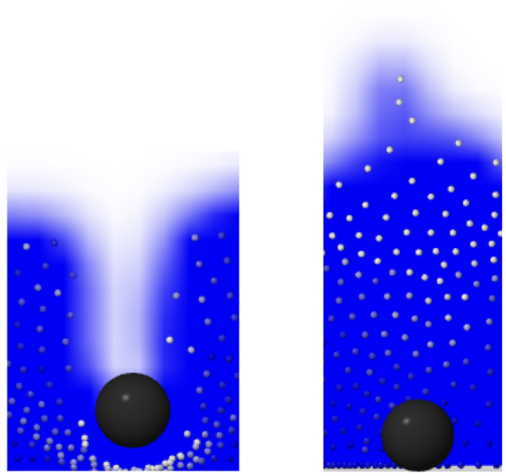

Figure 3: *Falling Rock Interaction*

One interesting feature of this interaction, which is not entirely intended but which we decided to keep, is that the water is actually pushed away by a rock with no velocity, meaning some water particles overlap with the rock as it is resting on the floor. This somewhat breaks the concept of 2-dimensionality, but it can be thought of in a 3D scene as water behind or in front of the rock. When we added in pressure to make the particles force away from the object when resting, the added force on the particles created less compelling interactions with the falling rock, so we decided to remove the pressure forces.

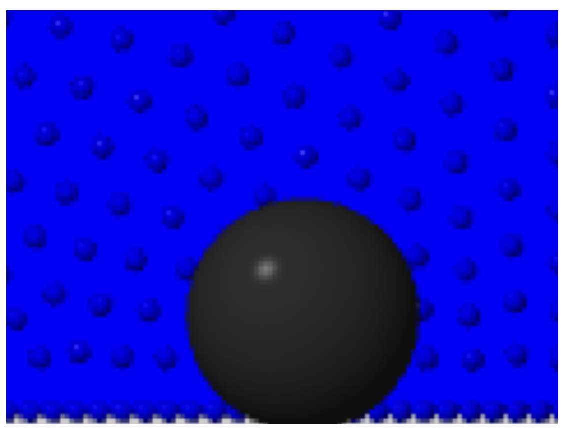

Figure 4: *Rock Particle Overlap*

Additionally, the particles had to remain on their border position, which meant that we would average the forces on each rock particle and set their velocity to be equal, maintaining the shape of the rock. With this, we were very satisfied with the results of the simulated interaction between the rock and water particles.

4.2 Boat

Similarly to the rock, we modeled the boat as a collection of particles which we drew a primitive mesh on top of. In terms of the interaction between water and boat, we did not get too far as we were working primarily on the falling rock. We reasoned that the boat particles should be pushed by the water particles similarly to how the water particles push themselves, but we importantly did not add torque to the rigid body, so it essentially stayed flat through its existence. We could also not model a decent floating of the boat, it was resting below water level which is not ideal. This in part was made harder by the fact that we did not have a good model for the boat, which should be some semi-circle like shape, but was instead implemented as a plank.

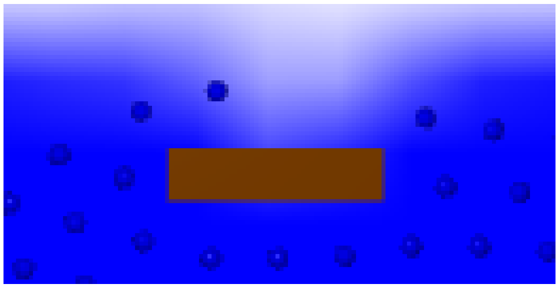

Figure 5: *Sad Little Boat*

5 Results

The interaction between the rock and the particles was indeed satisfying, but to extend our elation we tried to further the simulation by adding forces acting upon the rock by the water. Ideally this would allow the rock to be push around in the water depending on the shift of the water. However this interaction was hard to compute for us, and it ended up being more dysfunctional than we would have liked, so we decided to scrap the idea. As is, the rock can be modeled as too heavy to be influenced by the relatively shallow water.

The boat interaction was certainly disappointing to us, we had originally hoped for it to be the star of the show. However once we realized that the interaction between the rock and water was interesting enough, we decided to put our efforts into modelling that interaction. For the future, if we can add torque to the boat, create a better model for it, and nail the interaction with the SPH particles, I think we can make this project even more impressive.