RPI Advanced Computer Graphics 2019 Final Project
Coupling Fluid and Cloth Simulations
Water Seeping Through Fabric

Colin Higgins
(higgic4@rpi.edu)

Judy Zhang
(zhangz17@rpi.edu)

Abstract

We present our effort in coupling the simulations of cloth and fluid. With cloth simulated with the Mass-Spring Model and water simulation with Smoothed Particle Hydrodynamics, the initial results presented here show a promising approach for future implementation of similar projects.

1 Introduction

Both water and cloth simulations present significant technical challenges due to their nature of having multiple degree of freedom and are well-studied by many researchers in the field of Computer Graphics. To solidify our class learning, we chose to investigate the dynamics between water and cloth in this final project. With water represented as particles in SPH representation, and cloth represented as mass-spring system, major technical challenges reside in the dynamics: fluid dynamics with leaking effect and cloth collision detection. Real cloth holds some water before water seeps through. As cloth absorbs water, it also becomes heavier. On the other hand, when a water particle is shot through cloth, its trajectory path is changed after interacting with cloth.

In terms of cloth simulation, many methods have been studied. Mass-spring model was chosen for its simplicity and availability. Axis-aligned bounding boxes and a bounding volume hierarchy are used to provide efficient collision detection. To implement an interactive fluid simulation, SPH is implemented and preliminary results of handling water and cloth collision are included in this report.

We decided to do this project because both members feel the most comfortable with cloth simulation and fluid simulation. Although Colin was very interested in ray-tracing problem, Judy convinced him to take on the fluid challenge. Judy was also annoyed with self-intersection in the table cloth simulation and was motivated to implement cloth collision detection.

Works of researchers in these fields are summarized in section 2. Section 3 describes the overview of the algorithm while sections 4, 5, and 6 detail the data structures and implementations. Section 4 concerns the cloth simulation: cloth model and bounding volume hierarchy. Section 5 details the Smoothed Particle Hydrodynamics implementation. Then Section 6 describes preliminary fluid-cloth coupling. Section 7 discusses the results of the project and section 8 concludes the report with future works. Division of work is described in section 9 and acknowledgements in section 10.

2 Related Works

The dynamics behind cloth simulation, fluid simulation, and fluid-cloth coupling have been studied extensively. Bridson [3] presented a robust and efficient algorithm that handles cloth simulation that is used in fluid simulation with solid, thin shelled objects[1].

The Smoothed Particle Hydrodynamics(SPH) [6] model provides realistic and interactive water simulation, which is ideal to interact with cloth surfaces. Discussion from Basori's work[2] details simulation of wet cloth, effects of internal and external forces on a mass-spring cloth model. Realistic animation of cloth and water simulation have to incorporate key properties such as porosity and permeability as Andrysco discussed[5].

A successful modeling of water-cloth interaction is presented by Fei in 2018 [4]. The effects this paper presented are more extensive and beyond the scope of our project. We are inspired by their results and attempt to reconstruct similar test scenes.

3 Overview

The project concerns three major areas: cloth simulation, fluid simulation, and fluid-cloth coupling. The overall algorithm is summarized as follows:

```
For each animation step:
    Update cloth simulation
    update water simulation
    if water collides with cloth
        reflect the velocity of the water
```

In both cloth and water simulation, we use the most basic explicit method for numerical integration, Euler's method[8], to update cloth and water particle positions. There are many other integration methods that are more accurate and stable, however, at the current stage of development, Euler's method provides decent results and is easy to implement.

Further discussions on implementation are distributed in the following sections: Section 4 details the modeling and simulation of cloth. Section 5 concerns the fluid simulation, while section 6 presents the coupling between fluid and cloth. We include our preliminary results in section 7 and discuss future works in section 8.

4 Cloth Simulation

4.1 Cloth Model

To simulate the internal cloth dynamics, we use a mass-spring model given the simplicity and availability. In the basic model, cloth particles are arranged in a rectangular array connected horizontally, vertically, and diagonally. Particle to particle connections are categorized into three different kinds of springs:

- Particle [i,j] connect to [i+1,j] and [i,j+1] neighboring particles with **structural springs**.
- Diagonal pairs of particles: [i,j] to [i+1,j+1], and [i+1,j] to [i,j+1], are connected via **shear springs**.
- Particle [i,j] connect to [i+2,j] and [i,j+2] via **flexion springs**.

Cloth particles contain information such as mass, position, velocity, acceleration, etc. All particles have the same mass at the beginning of the simulation.

Illustration of the cloth model is included in figure above. During animation, Provot correction [7] constraints the springs of cloth particle to deform up to

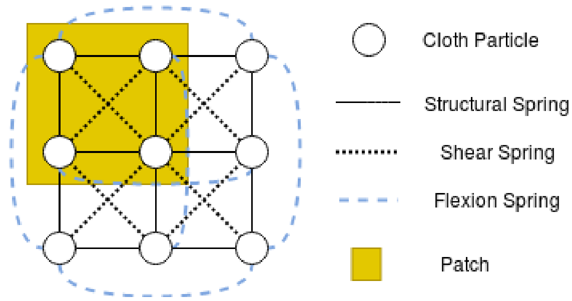

Figure 1: Cloth Model

10% of it's original length, preventing the springs from over-stretching.

This cloth model itself has been used by many authors such as Provot[7], Bridson[3], Guendelman[1], etc. Although there are variations to the usage, we follow the most basic and readily available model from the previous homework.

4.2 Bounding Volume Hierarchy

Since cloth particles are stored in an array, it is intuitive to have four neighboring particle constituting a cloth patch, as indicated by yellow highlight in figure 1. Each patch is bounded with an Axis-Aligned Bounding Box, with a minimum small constant thickness, illustrated in figure 2. Here we take similar approach to Bridson[3]. Once a patch changes its orientation, the bounding box

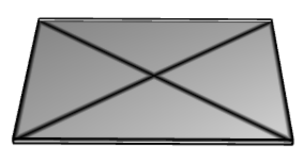

Figure 2: Single Patch Bounded

expands to bound the maximum and minimum positions of the cloth particles.

Inspired by Bridson[3], we implemented a bounding volume hierarchy(BVH) to accelerate collision detection with the cloth model. The BVH is implemented with a 3-d tree. At the root of the tree, the bounding volume bounds the entire cloth model while each leaf of the tree contains a single patch. At the beginning of the simulation and at the end of each animation step, the 3-d tree is constructed by taking in the overall bounding

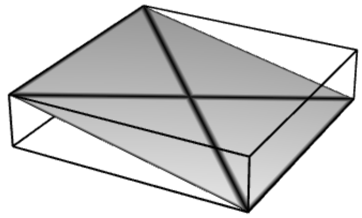

Figure 3: Axis-Aligned Bounded Patch

volume of the cloth model and all patches. The AABB of cloth model is stored as the root of the tree while the leaves are grouped into a vector and sorted based on x, y, or z axis position values. Nodes in the middle of the tree are constructed by dividing along the longest axis of a parent node and then subdivide until all patches belong to a parent.

To verify collision detection in cloth model, self-intersection detection was implemented. By taking the position of every particle into a AABB and collect all patches intersecting it, we can easily detect if this particle is colliding with a foreign patch. That is, if a particle is not within a patch, this patch is foreign to this particle.

5 Fluid Simulation

To simulate the properties of fluids we implemented Smoothed Particle Hydrodynamics (SPH). Our implementation closely followed that of Müller [6]. Before introducing our equations please note the following variables and their corresponding definitions:

$$\rho - density$$
$$d - restdensity$$
$$\mu - viscosity$$
$$p - pressure$$
$$k - gasconstant$$
$$g - accelerationduetogravity$$
$$v - velocity$$
$$t - time$$
$$m - mass$$
$$f - force$$
$$r - position$$
$$W - smoothing\ kernel$$
$$h - the\ size\ of\ the\ smoothing\ kernel$$
$$n - the\ total\ number\ of\ water\ particles$$
$$i - the\ index\ of\ the\ current\ particle$$
$$j - the\ index\ of\ the\ comparison\ particle$$

With our variables defined we begin our simulation by approximating the local density of the fluid. For each water particle we iterate through all other particles and apply the following equation on each of those particles, please reference Appendix A for the equations of the Smoothing Kernels:

$$\rho_i = \sum_{j=0}^{n} m_j * W_{poly6}(r_i - r_j, h)$$

These densities in turn allow us to compute the local pressure via the equation:

$$p_i = (\rho - d) * k$$

With the pressure and density calculated for each particle we are able to calculate the force resulting from pressure at each particle. This was done by again iterating through all particles and applying the following equation:

$$f_p = \sum_{j=0}^{n} -m_j * \frac{p_i + p_j}{2 * \rho_j} * W_{spiky}(r_i - r_j, h)$$

In the same iterations the force due to the fluid's viscosity was also calculated from the equation:

$$f_\mu = \sum_{j=0}^{n} \mu * mass_j * \frac{(v_i - v_j)}{\rho_j} * W_{viscosity}(r_i - r_j, h)$$

We deviate from Müller's [6] implementation by not including the force of surface tension at this point in our simulation to simplify our implementation. With these forces computed we are able to update the accelerations of each point and then perform Euler interpolation to update the velocities and locations of each particle.

6 Fluid-Cloth Coupling

To couple our fluid and cloth simulations together the bounding volume hierarchy is reused to speed up the detection of collisions between cloth patches and water particles. For each water particle we searched through the AABB tree to find nearby cloth patches and particles. With the nearby particles gathered we then calculate the equations of the planes for each of the triangles that make up the cloth patch by taking the cross product of their vector components. The current and previous positions of the water particle were then compared to these equations, and if it was determined that the two positions were on opposite sides of either plane the water particle was determined to have collided with the cloth patch. At this point the velocity of the water particle was reflected about the normal of the plane it collided with, and its position was stepped backwards one time-step.

7 Results, Performance, Known Bugs

7.1 Cloth Results

BVH construction and update both takes $O(nlog(n))$, where n is the number of patches in the cloth model. To search within the hierarchy and collect patches of interest, only $O(log(n))$ is needed.

With BVH, cloth self-collision could be detected efficiently by checking each cloth particle against all cloth patch. In other words, each animation step would take $O(nlogn)$ to prevent self-collision. By adding penalty force onto the cloth particle in the direction of Gouraud normal during collision, cloth self-intersection could be avoided. However, with small cloth models, self-collision correction does not settle down and could result in "hand-clapping" motion illustrated in figure 4.

Figure 4: Small Cloth Model with Hand Clap Motion

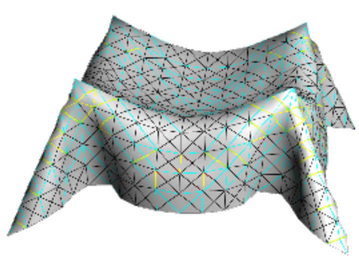

Figure 5: Resting Configuration of Table Cloth Simulation

7.2 Fluid Results

Updating the status of all water particles is on the order of $O(n^2)$ where n is the number of water particles in the simulation. Unfortunately, the constants for this simulation could not be accurately set to reflect visually

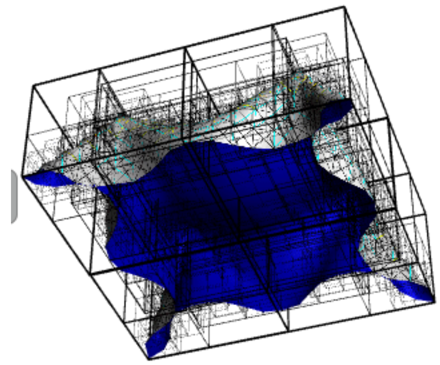

Figure 6: Bottom View of Table Cloth

plausible fluid behaviors due to time constraints. As a result the simulation is unstable and will either explode, figure 7, or implode, figure 8, with no signs of convergence. However, with few enough particles and the right constants, the fluid can behave well enough to test the other aspects of our program such as the coupling of the fluid and cloth simulation.

Figure 7: Explosion of water simulation to the four corners of the bounding box

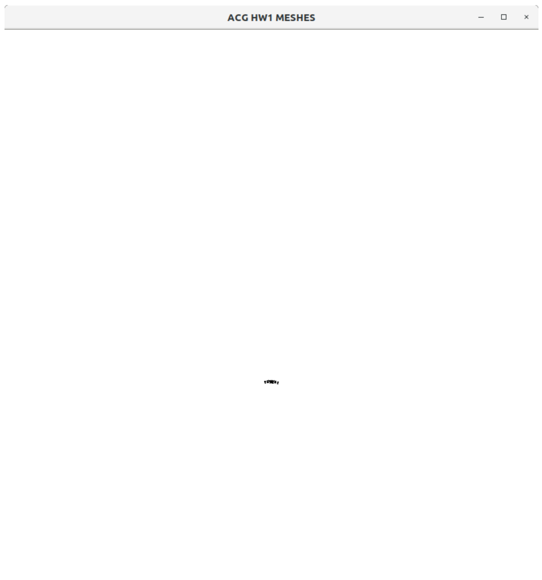

Figure 8: Implosion of water simulation to the center bottom of the bounding box

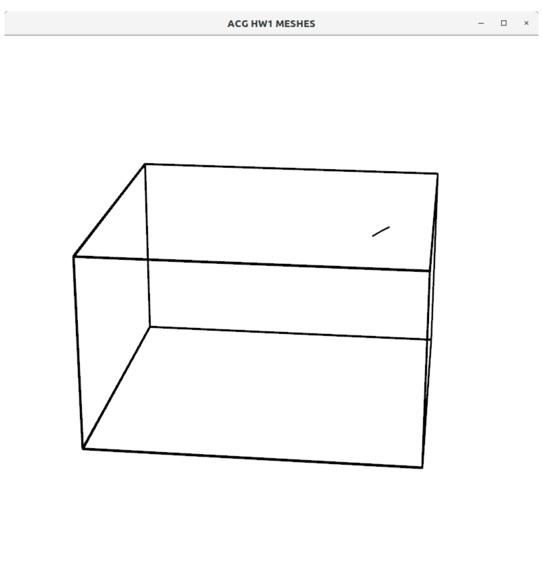

Figure 9: Stream of water

7.3 Fluid-Cloth Coupling Results

Detecting collisions between water particles and cloth patches is on the order of $O(nlog(m))$ where n is the number of water particles and m is the number of cloth patches. Furthermore, the collision detection is fairly robust and rarely allows particles to pass through the cloth when they should not. Figure 10 displays the water pooling in cloth. Despite it appearing that water particles have gotten through the cloth, hiding the surfaces shows that they have not and that it is merely the resolution of the marching cubes implementation we borrowed from Homework 2 which skews the results.

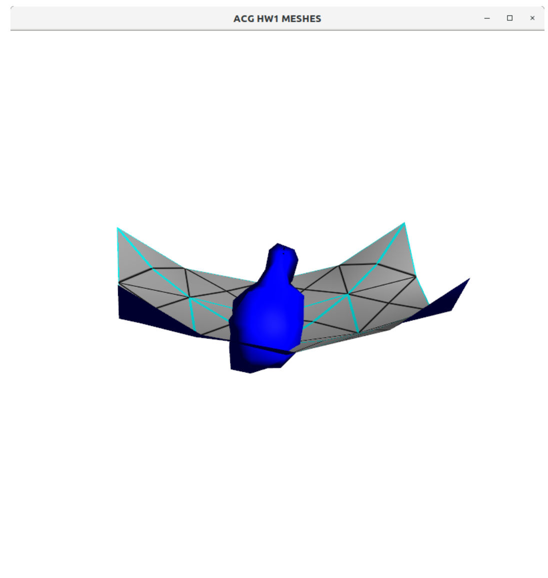

Figure 10: Collision between water and cloth surface view

Figure 11: Collision between water and cloth particle view

8 Conclusion and Future Work

Despite limited time and health conditions of both members, the project serves it's original purpose of learning. Both members learned valuable experience in the implementation and usage of BVH and explored the

SPH model of fluid simulation.

Due to the scope this project, many improvements could be done. In terms of cloth simulation, BVH update could be done in linear time without throwing out the entire tree. As for the fluid simulation, much more time could have been spent tweaking the parameters of our SPH implementation, as although we are confident in the implementation of the proper equations, we were unable to identify the combination of constants that provide visually plausible results. Alternatively, a return to Marker and Cell could offer a simpler implementation at the sake of precision. As for the coupling of the two simulations there are many features we wished we were able to implement. Chief among these are absorption and permeability. Our plans for these features was to incorporate storage arrays in each cloth particle that would gather water particles on collision to simulate absorbance. Once these filled the cloth particle would then transfer a small number of the colliding water particles to the other side of the cloth each time step to simulate permeability. In our research for this project we came across Andrysco's [5] paper, Permeable and Absorbent Materials in Fluid Simulations. This paper provided us with the following equations that we planned to implement to add absorbance and permeability to our simulation:

$$q = \frac{-\kappa}{\mu}(\nabla p - \rho g)$$

$$\phi = \frac{V_V}{V_T}$$

With these equations we could also model evaporation, capillary action, and other properties that we had hoped implement.

For future projects, team members recommend to read through previous students' projects and get inspired by their implementation.

9 Challenges

For this project, the main challenge was to choose which simulation technique to implement in a fixed amount of time. Given various options, both members spend a large amount of time contemplating the optimal choice. In the end, we settled with AABB with 3d-tree BVH and SPH.

10 Division of Work

Colin worked on the fluid simulation while Judy was responsible for the cloth collision detection. Both members estimated to spend 30+ hours each on implementation of this project, in addition to thorough reading of the papers. Despite various health conditions, we gained valuable experience evaluating various simulation techniques and implementing specific ones for this project.

11 Acknowledgments

This project is developed as a final project for Advanced Computer Graphics offered in Spring 2019. We want to take this opportunity to thank our peers for the comments they offered in Discussion Forum and inspiration they offered through their presentations. We also greatly appreciate the guidance, support, and homework code Professor Cuter has offered throughout this semester.

12 Appendix A

This appendix contains the equations used to compute the values of the kernels described above. Please note that all kernels evaluate to 0 if $(r_i - r_j)$ is greater than h. All kernels were taken from Müller [6].

$$Polynomial\ Kernel\ -\ W_{poly6}$$

$$W_{poly6} = \frac{315 * (h^2 - (r_i - r_j)^2)^3}{64 * \pi * h^9}$$

$$Spiky\ Kernel\ -\ W_{spiky}$$

$$W_{spiky} = \frac{-45 * (h - (r_i - r_j))^2}{\pi * h^6}$$

$$Viscosity\ Kernel\ -\ W_{viscosity}$$

$$W_{viscosity} = \frac{45 * (h - (r_i - r_j))}{\pi * h^6}$$

References

[1] *Coupling Water and Smoke to Thin Deformable and Rigid Shells* (New York, NY, USA, 2005), SIGGRAPH '05, ACM.

[2] Basori, A. H., AlJahdali, H. M. A., and Abdullah, O. S. Forced-driven wet cloth simulation based on external physical dynamism. *International Journal of Advanced Computer Science and Applications 8*, 11 (2017).

[3] Bridson, R., Fedkiw, R., and Anderson, J. Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics 21* (2002), 594–603.

[4] Fei, Y. R., Batty, C., Grinspun, E., and Zheng, C. A multiscale model for simulating liquid-fabric interactions. *ACM Trans. Graph. 37*, 4 (Aug. 2018), 51:1–51:16.

[5] McDonnell, R., Ed. *Permeable and Absorbent Materials in Fluid Simulations* (2008), The Eurographics Association. Posters and Demos.

[6] MÜLLER, M., CHARYPAR, D., AND GROSS, M. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), SCA '03, Eurographics Association, pp. 154–159.

[7] PROVOT, X. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *IN GRAPHICS INTERFACE* (1995), pp. 147–154.

[8] WEISSTEIN, E. W. "euler forward method." from mathworld–a wolfram web resource.