ACG-Simulation: Mid-term Progress Report

Xingyan Chen
chenxy24@mails.tsinghua.edu.cn
IIIS, Tsinghua University
Beijing, China

Zhengyang Zhang
zhangzy24@mail.tsinghua.edu.cn
IIIS, Tsinghua University
Beijing, China

Abstract

This report summarizes the mid-term progress of our ACG simulation project, focusing on SPH fluid simulation, rigid body dynamics, and rendering pipeline.

Keywords

SPH fluid simulation, rigid body dynamics, Blender rendering

ACM Reference Format:
Xingyan Chen and Zhengyang Zhang. 2025. ACG-Simulation: Mid-term Progress Report. In *Proceedings of ACG Mid-term Project Progress Report*. ACM, New York, NY, USA, 2 pages.

1 Introduction

1.1 Project Topic

Our project aims to simulate the collision of different types of objects and render the simulation outcome with an industrial renderer.

1.2 Project Goals and Technical Points

Our primary goal includes implementing different types of object simulations: rigid body, cloth, fluid and their collision. We also aim to build a rendering pipeline to visualize the simulation results with high quality. Apart from basic implementation, we plan to optimize the simulation performance, provide user-friendly configuration options and support real-time interaction.

The technical points of our project include:

- Implementing rigid body dynamics with collision detection and response.
- Implementing fluid simulation using the Lagrangian point-based WCSPH method [2].
- Implementing fluid and rigid body collision handling with a momentum-conserving two-way coupling method [1].
- Building a rendering pipeline using Blender to visualize simulation results.
- Optimizing simulation performance through algorithmic improvements and multi-threading.
- Providing flexible configuration options for different simulation scenarios.
- Supporting real-time interaction for customized scene setup.

ACG Mid-term Project Progress Report, Tsinghua University, Beijing, China

2 Schedule

- week 1-9: Learn basic knowledge about simulation and rendering.
- week 10: Read related papers and documents about simulation and collision implementation methods.
- Weeks 11-12: Implement basic SPH fluid simulation and rigid body dynamics with collision handling. Build up basic rendering pipeline.
- week 13: Implement collision between fluid and rigid bodies. Handleing complex geometry. Try to improve efficiecy of simulation by improve algorithms or using multi-threading.
- week 14: Implement control for customized scene configuration, fixed operation in procudure and interactive in real-time.
- week 15: Test the whole pipeline, fix bugs and improve performance.
- Weeks 16: Finalize rendering pipeline and generate high-quality output videos. Prepare for final demonstration.
- Weeks 17-18: Organize the codebase, complete flexible configuration. Write final report.

3 Methods Completed

3.1 Rigid Body

3.2 Collision Handling

3.3 Fluid

We have implemented a basic WCSPH fluid simulator using the Taichi programming language. The simulator supports essential SPH operations such as density and pressure computation, viscosity and surface tension forces, and time integration. We have also integrated the SplashSurf library to reconstruct fluid surfaces from particle data for rendering.

The WCSPH method approximates fluid dynamics using a set of particles, where each particle represents a small volume of fluid. The core equations include:

Density computation for particle i:

$$\rho_i = \sum_j m_j W(r_{ij}, h)$$

where m_j is the mass of particle j, W is the smoothing kernel with support radius h, and $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$.

Pressure calculation using the Tait equation:

$$p_i = B\left(\left(\frac{\rho_i}{\rho_0}\right)^\gamma - 1\right)$$

where ρ_0 is the rest density, B is the bulk modulus, and $\gamma = 7$ for water.

The pressure force on particle i:

$$\mathbf{f}_i^{pressure} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(r_{ij}, h)$$

Viscosity force:

$$\mathbf{f}_i^{viscosity} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(r_{ij}, h)$$

where μ is the dynamic viscosity.

Surface tension force (approximated):

$$\mathbf{f}_i^{surface} = -\sigma \sum_j m_j \frac{\mathbf{r}_{ij}}{r_{ij}} \nabla W(r_{ij}, h)$$

where σ is the surface tension coefficient.

Time integration uses a semi-implicit Euler method to update positions and velocities.

To efficiently find neighboring particles, we employ a uniform grid data structure. Each particle is assigned to a grid cell based on its position, and only particles in the same or adjacent cells are considered as potential neighbors, reducing the computational complexity from $O(N^2)$ to $O(N)$.

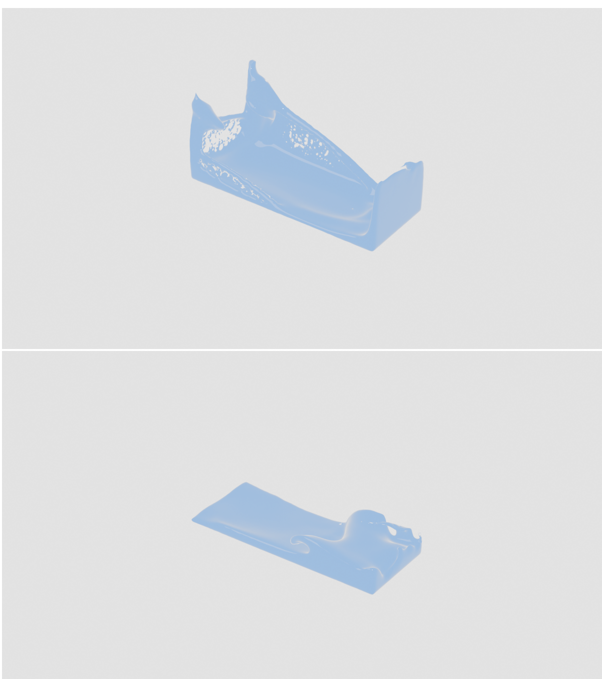

Figure 1: Fluid simulation results

3.4 Rendering Pipeline

We have successfully implemented the core pipeline components. The rendering pipeline features adjustable rendering with Principled BSDF materials for water effects, EEVEE engine with screen space reflections, and PNG output. Video creation uses ffmpeg to encode PNG sequences into MP4 videos with configurable settings.

4 Plan for Remaining Technical Tasks

The remaining tasks focus on refinement and optimization.

First, we plan tto read more advanced papers on fluid simulation and collision handling to explore potential improvements of the basic WCSPH method. We will also investigate parallelization techniques to enhance simulation performance.

Second, we will implement the fluid-rigid body collision handling using a momentum-conserving two-way coupling method. This involves calculating interaction forces between fluid particles and rigid body surfaces, ensuring realistic momentum exchange during collisions.

What is more, we will support more complex geometries for rigid bodies, and implement configuration options for different simulation scenarios.

Finally, we will fine-tune the rendering parameters to achieve more realistic water appearance, including material properties and lighting setup.

5 External Tools

We are using the following external tools and libraries:

- `taichi`: For high-performance physics simulation.
- `Blender`: For 3D rendering and animation.
- `splashsurf`: For surface reconstruction from particle data.
- `ffmpeg`: For video encoding from image sequences.

References

[1] Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile rigid-fluid coupling for incompressible SPH. *ACM Trans. Graph.* 31, 4, Article 62 (July 2012), 8 pages. doi:10.1145/2185520.2185558

[2] Markus Becker and Matthias Teschner. 2007. Weakly compressible SPH for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '07)*. Eurographics Association, Goslar, DEU, 209–217.