서울대학교
SEOUL NATIONAL UNIVERSITY

Programming Assignment Report #3

soheon yoo (2021-22823)
yumk0717@snu.ac.kr
Seoul National University - Advanced Graphics (Spring 2023)

Abstract

Water simulation was done using Smoothed Particle Hydrodynamics(SPH), which is a kind of lagrangian method for fluid simulation. Poly6 was used for kernel function and viscosity was disregarded. Using SPH simple engineering problem about dam was solved

1 Introduction

Fluid simulation and rendering is important part in computer graphics. There are eulerian method and lagrangian method for simulating fluids. Both method is for calculating navier-stokes equation. Eulerian method is based on grid and space and lagrangian method is based on particles. In this project, Smoothed Particle Hydrodynamics(SPH), which is a kind of lagrangian method, was used for water simulation. SPH was proposed by Gingold and Monaghan[1] and Lucy[2] for modeling in the field of astrophysics. Not only in astrophysics, but also in the field of fluid simulation, SPH was useful.

2 Methods

Particle in SPH was rendered as simple red circle in simulation. THREE.js and python was used for simulation and rendering.

3 Model explanation

SPH is grid-free method ,so calculation is done based on particles. Because we calculate based on particle, Navier-Stokes equation is simplified to below one

$$\frac{du}{dt} = g - \frac{1}{\rho}\nabla p + v\nabla^2 u$$

To calculate acceleration, we need to know gradient of pressure and laplacian of velocity. In SPH, these quantities are calculated by interpolating neighboring particles. Interpolation can be done by integral with kernel function W. Kernel function should have property below one.

$$\int W(|x - x^{'}|, h)dx^{'} = 1$$

If we want to calculate some quantity in x,$q(x)$, integral $q(x^{'})W$ is answer. And, for gradient of some quantity, integral $q(x^{'})\nabla W$ is answer.

$$q(x) = \int q(x^{'})W(|x - x^{'}|, h)dx^{'}$$
$$= \sum_j \frac{m_j}{\rho_j} q(x_j)W(|x - x^{'}|, h)$$

$$\nabla q(x) = \int q(x^{'})\nabla W(|x - x^{'}|, h)dx^{'}$$
$$= \sum_j \frac{m_j}{\rho_j} q(x_j)\nabla W(|x - x^{'}|, h)$$

In this project poly6 was used.

$$W_{ploy6}(r,h) = \alpha_1 \begin{cases} (h^2 - r^2)^3, 0 \le r \le h \\ 0, \text{otherwise} \end{cases}$$
$$\alpha_1 = \frac{4}{\pi h^8}, \frac{315}{64\pi h^9} \text{for 2D,3D}$$

Density and gradient of pressure at position x(position at particle i) can be calculated using this method.

$$\rho_i \approx \sum_j m_j W(|x - x_j|, h)$$
$$\frac{\nabla p_i}{\rho_i} \approx \frac{1}{\rho_i} \sum_j \frac{m_j}{\rho_j} p_j \nabla W(|x - x_j|, h)$$

p_i is proportional to ρ_i physically, because of ideal gas law(PV= NRT). But for stablity of fluid, we set rest density.
$p_i \approx k(\rho_i - \rho_0)$
And for calculating gradient of pressure, instead of using pressure at j, average pressure between i and j was used.

4 Algorithm

The algorithm is very similar to simple particle simulation. Calculate acceleration each step, and update velocity, position. Before calculating acceleration, we need to calculate density and pressure of each particle firstly.

for *all particles* i **do**
  Initialize x_i and m_i
end
while *simulating* **do**
  for *all particles* i **do**
    compute density ρ_i
    compute pressure p_i
  end
  for *all particles* i **do**
    compute pressure gradient $\frac{\nabla p_i}{\rho_i}$
    update $\frac{du}{dt}$
  end
  for *all particles* i **do**
    update velocity u_i
    update positions x_i
  end
end

Algorithm 1: SPH Algorithm

5 Problem

We will solve simple dam construction problem.

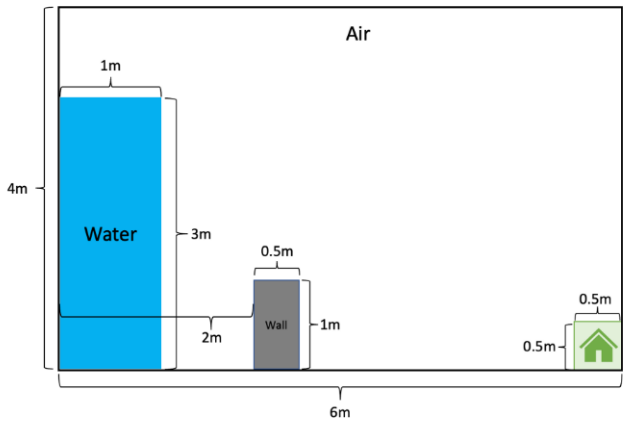

Figure 1: **Dam simulation**

Using SPH, water was implemented like real-life fluid. This system is in 2D and wall in the center block fluid particle. We want to know whether the particle can enter the 0.5m x 0.5m house. The density of water is 997kg/m3. So i set up density in 2D as 997kg/m2. The mass of one particle depends on how much particle is in the unit area.

To decide stiffness and rest density of this simulation, I $2m^2$ water with 200 particles, in start position. The water should remain left side of wall. The simulation should look like this one.

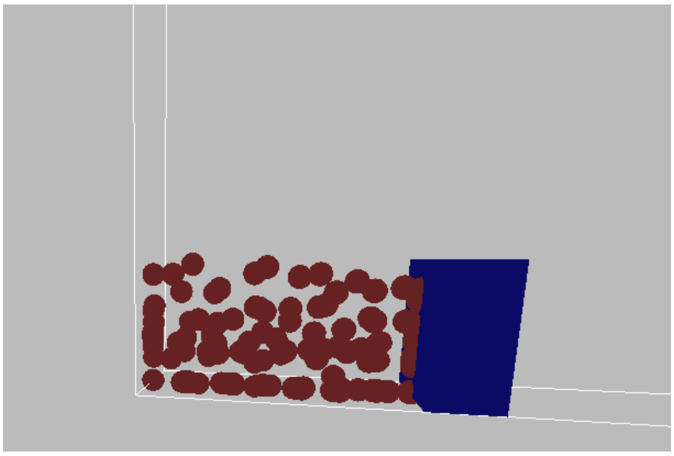

Figure 2: **Dam simulation example with 200 particles($2m^2$)**

5.1 Solution

The distance between particle is 0.1m. So 3mx1m water contains 300 particles. In the problem of figure 1, Wall can not hold $3m^2$ water. It is reasonable because area of left side is only $2m^2$. So we need to move wall right or make wall taller. Then how can we block the water from house? Firstly, because of zero-viscosity it is almost hard to contain fluid in dam. Zero- viscosity make fluid ideal gas.
If the height of dam is changed to 2.2m or offset of dam is changed to 7.3m house can be protected . But the house is 5.5m from zero point. So changing wall offset cannot protect house from water.

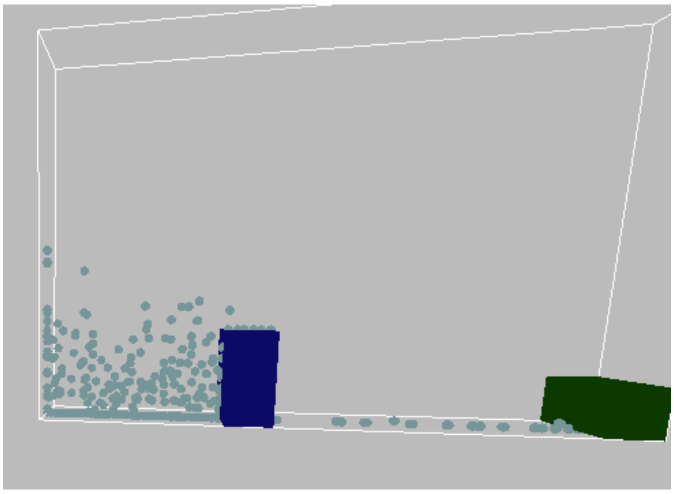

Figure 3: **Figure 1 simulation**

Because SPH algorithm is approximate implementation of fluid. It has some difference from real world. In SPH particle and time is discrete not like world. Smaller timestep and particle would make simulation more like continuous system with litte error.

6 Discussion

poly6 for kernel function is not only one. There is various kernel function like spiky or viscosity kernel. Especially, spiky kernel is useful in implementing pressure term[3]. Pressure term in navier-stokes equation is related to repulsion between particle. But the value of kernel function gradient near zero is almost zero. It means that, very close two particle will not repulse in fluid system. So I used 2D- spiky kernel for this simulation.
Because this system neglect viscosity term. It is really hard to implement water-like fluid. zero-viscosity made fluid to behave like air. And, because it was 2D simulation, kernel was changed to 2D version.

References

[1] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: theory and application to non-spherical stars," *Monthly Notices of the Royal Astronomical Society*, vol. 181, pp. 375–389, 12 1977.

[2] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis.," , vol. 82, pp. 1013–1024, Dec. 1977.

[3] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, (Goslar, DEU), p. 154–159, Eurographics Association, 2003.