CG_9_SPH Fluid Report

PB21010479 Caoliwen Wang

May 12, 2024

1 Problem Statement

In this assignment, we will delve into the world of fluids, where you will learn to utilize Smoothed Particle Hydrodynamics (SPH), a classic particle-based simulation method, to compute fluid motion.

Smoothed Particle Hydrodynamics (SPH) is a method for fluid simulation that relies on representing the fluid with a large number of particles and simulating their interactions to mimic fluid motion. The core idea of SPH is to approximate continuous fluid properties (such as density, pressure, velocity, etc.) by summing over discrete particles. This allows SPH to effectively handle large deformations, surface tension, and interactions between fluids and solids.

To use SPH for simulating fluid motion, it is necessary to establish the basic equations of motion for the fluid, typically including equations for mass conservation, momentum conservation, and energy conservation. These equations describe the internal motion of the fluid, as well as interactions between the fluid and solid boundaries or other fluids. In SPH, these equations are discretized based on each particle in the fluid, and then approximated by summing over neighboring particles to approximate the conservation equations for continuous fluid.

In terms of numerical solving, SPH typically involves discretizing the equations in both time and space. Time discretization often employs explicit or implicit time integration methods such as Euler's method, Runge-Kutta method, etc. Spatial discretization involves effectively representing physical quantities like density, pressure, velocity, etc., and updating these quantities based on interactions between neighboring particles.

In practical applications, SPH must account for various complex physical phenomena such as surface tension, viscosity, shock waves in fluid flow, etc. Therefore, selecting appropriate numerical methods and parameter tuning is crucial for obtaining accurate simulation results. Additionally, to enhance simulation efficiency, considerations such as spatial partitioning, particle sorting, etc., are often taken into account.

2 Propose Algorithms

2.1 Navier-Stokes Equations

Fluid simulation and the solution of general partial differential equations (PDEs) are fundamentally similar, both based on appropriate temporal and spatial discretization formats, utilizing numerical methods to solve the governing PDEs describing their motion. Within fluid dynamics, we solve the Navier-Stokes equations:

$$\begin{aligned} \rho\frac{D\mathbf{v}}{Dt} &= \rho\mathbf{g} - \nabla p + \mu\nabla^2\mathbf{v} \\ \nabla\cdot\mathbf{v} &= 0 \end{aligned} \tag{2.1}$$

where ρ is the fluid density, $\mathbf{v}$ is the velocity field, $\mathbf{g}$ is the gravitational acceleration, p is the pressure within the fluid, and μ is the viscosity coefficient. The right-hand side terms of the first equation correspond to gravity $\rho\mathbf{g}$, pressure $-\nabla p$, and viscous forces $\mu\nabla^2\mathbf{v}$.

It's worth mentioning that the term $\frac{D\cdot}{Dt}$ in equation (2.1) is referred to as the "material derivative" or "derivative following the motion," which is a special case of the total derivative in fluid mechanics:

Consider a fluid element moving with the flow, and at each point within this element, we sample a physical quantity $f = f(\mathbf{x}, t)$, where $\mathbf{x}$ is the spatial position and t is time. As the fluid element moves, $\mathbf{x} = \mathbf{x}(t)$ becomes a function of time. Then, the total derivative of f with respect to time at the sample point is given by: $\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial f}{\partial t} + \mathbf{v}\cdot\nabla f$. We denote this as the material derivative of the physical quantity f on the moving fluid element, expressed as $\frac{Df}{Dt}$, which accounts for both the change of f due to time and the spatial motion of the sampling point.

NS equation's first line cannot be independently solved. To determine the pressure p, we need to consider the incompressibility condition of the fluid:

$$\nabla\cdot\mathbf{v} = 0 \tag{2.2}$$

Combining the physical meaning of divergence (fluid convergence and divergence), we can see that incompressibility is equivalent to mass conservation. That is, the material derivative of the fluid density at any position in space is zero (if the divergence is not equal to zero, i.e., the fluid converges to or diverges from a point, then the mass of the fluid at that point will increase or decrease, violating mass conservation). This is illustrated in Figure 1.

2.2 Spatiotemporal Discretization

In terms of temporal discretization, a common approach in fluid simulation is a technique called "operator splitting," which divides the Navier-Stokes equations into two parts:

1. Without considering pressure, updating $\mathbf{v}$:

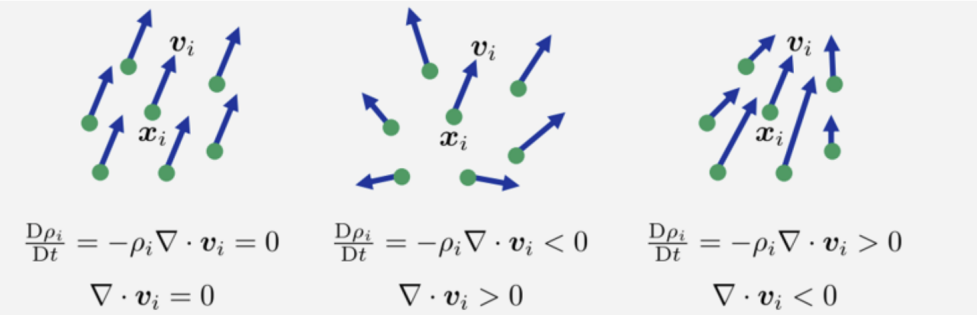

Figure 1: Illustration of incompressibility condition

$$\rho \frac{D\mathbf{v}}{Dt} = \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}$$

2. Considering pressure, calculating pressure, and updating $\mathbf{v}$:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p$$

3. Based on the final velocity, updating particle positions.

In terms of spatial discretization, Smoothed Particle Hydrodynamics (SPH) samples physical fields using particles and approximates the physical quantities at unsampled positions through kernel functions. Commonly used kernel functions in SPH are as follows (d is the dimension of the simulation):

$$W(r,h) = \sigma_d \begin{cases} 6(q^3 - q^2) + 1 & \text{for } 0 \le q \le \frac{1}{2} \\ 2(1-q)^3 & \text{for } \frac{1}{2} \le q \le 1 \\ 0 & \text{otherwise} \end{cases}$$
$$\text{with } q = \frac{1}{h}\|r\|, \sigma_1 = \frac{4}{3h}, \sigma_2 = \frac{40}{7\pi h^2}, \sigma_3 = \frac{8}{\pi h^3}$$

2.3 Computational Formula

The computational formulas in SPH for density and velocity divergence are as follows:

1. Density calculation:

$$\rho_i = \sum_{j+i} \left(\frac{m_j}{\rho_j} \right) m_j W(\mathbf{x}_i - \mathbf{x}_j, h) = \sum_{j+i} m_j W_{ij} \tag{2.3}$$

Here, we denote $W_{ij} = W(\mathbf{x}_i, \mathbf{x}_j, h)$, where j represents all neighboring particles of particle i. In SPH, h represents the radius of the kernel function, and we use Δt to represent the time step.

Note that when calculating the density of a particle, we need to consider its own contribution to density, $m_i W(0,h)$ (using the function `W_zero`).

In the program, we assume that each fluid particle has the same mass (equal to the volume multiplied by the density of the fluid at rest, where the resting density is 1000, the particle radius is set to 0.025, and the particle volume is estimated using a cube; please refer to the implementation of the `ParticleSystem` class), which can be accessed using `ps_.mass()`.

2. Velocity divergence calculation:

$$\nabla \cdot \mathbf{v}_i = \sum_j \frac{m_j}{\rho_j}(\mathbf{v}_j - \mathbf{v}_i) \cdot \nabla W_{ij} \tag{2.4}$$

For the calculation of viscosity, we recommend using the formula:

$$\nabla^2 \mathbf{v}_i = 2(d+2) \sum_j \frac{m_j}{\rho_j} \frac{\mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|^2 + 0.01h^2} \nabla W_{ij} \tag{2.5}$$

Where d is the dimension of the simulation, which is 3 in this case. We divide the viscosity coefficient μ by the particle density to obtain the value $\nu = \frac{\mu}{\rho}$ (referred to as the kinematic viscosity coefficient), which serves as a tunable simulation parameter named `viscosity`.

3. pressure calculation:

The task for this assignment is to implement a classical method called "Weakly Compressible Smoothed Particle Hydrodynamics" (WCSPH).

The pressure calculation follows the equation of state (EOS) formula:

$$p_i = k_1 \left(\left(\frac{\rho_i}{\rho_0} \right)^{k_2} - 1 \right) \tag{2.6}$$

In the program, k_1 is represented by the parameter `stiffness`, and k_2 is represented by the parameter `exponent`. Note that the pressure should generally be greater than 0. In case of insufficient particles, you can set $p_i = \max(0.0, p_i)$.

The acceleration due to pressure is given by $-\frac{1}{\rho}\nabla p$, where:

$$\nabla p_i = \rho_i \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W_{ij} \tag{2.7}$$

Here, ∇p_i represents the gradient of pressure for particle i, ρ denotes the density, m is the mass, and W_{ij} is the kernel function.

3 Programming

3.1 Node programming

In this assignment,The node used in this instance is very simple. We adopted the connection method as shown in Figure 2.

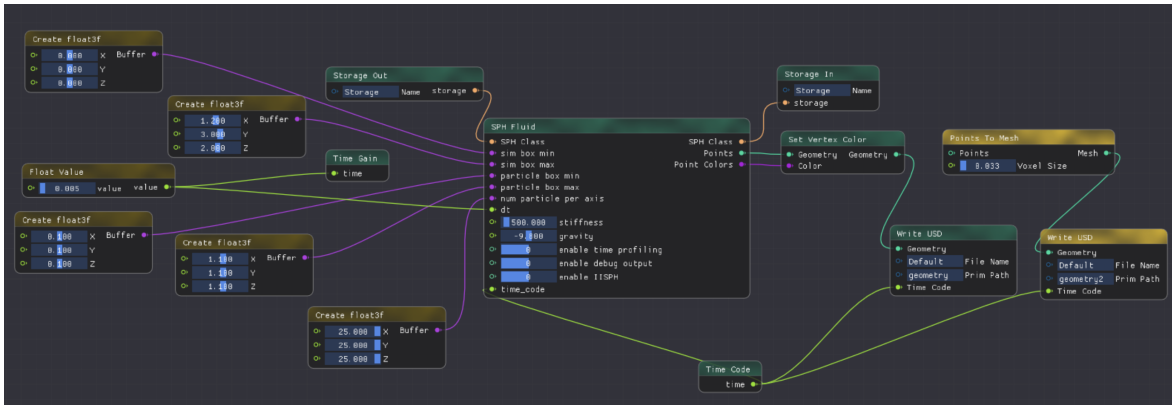

Figure 2: Geometry Nodes

4 Experimental Results

I think the results of this algorithm are fantastic.

We first conducted tests on different physical parameters, modifying the values of ”exponent” and ”stiffness,” resulting in different video outcomes. You can find the video results in the ”outcomes” folder.

We can observe that with an increase in the ”stiffness” and ”exponent” parameter, the rebound height increases.

Afterward, we changed the size of the viscosity parameter, representing the interaction between particles.

Finally, we present an animation regarding the grid.

We can observe that the results are still relatively realistic. All the videos are in the ”outcomes” folder.

5 Summary

In this assignment, we applied the SPH method to simulate fluid behavior, and the results were relatively good. However, when we adjusted certain parameters to very large values, the video exhibited an explosion phenomenon, indicating that the numerical computation may not be entirely stable. This presents another area for optimization.