Overview for Fluid Simulation

Zhouyuan Chen

March 2024

1 Introduction

After 2 years, I reread the book Fluid Simulation for Computer Graphics again. The first time I read this book, I couldn't fully understand the concepts and I only finished the first basics part. But this time, after learning Numerical Methods, I found that this book is really easy to read.

While I am implementing the algorithms that appeared in this book, I would like to write the idea down and do coding work with my notes.

I hope my note can help someone else to learn this computer graphics topic. But for the time limitation, I would like to skip some description about Finite Difference Methods and Finite Volume Methods

1.1 Fluid Formula

The core of the fluid simulation is the incompressible *Navier-Stokes* equation:

$$\begin{cases} \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{g} + v \nabla \cdot \nabla \vec{u} \\ \nabla \cdot \vec{u} = 0 \end{cases}$$

The second equation is easy to understand. It tells us that the accumulation of the flux in a grid should be 0. The first part actually can be written as:

$$\frac{Du}{Dt} = \vec{g} - \frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla \cdot \nabla \vec{u}$$

Where $\frac{Du}{Dt} = \frac{du(t, \vec{x}(t))}{dt}$. This term represents how a property(here is u) changes alone with position and time. Typically, this term equals 0 since the property typically holds the same value except it disappears or somehow is removed. The equation above comes from this equation:

$$\vec{F} = m\vec{g} - V\nabla\rho + V\mu\nabla \cdot \nabla \vec{u}$$

Where $m\vec{g}$ is gravity, $-V\nabla\rho$ is pressure force, and $V\mu\nabla \cdot \nabla\vec{u}$ is viscosity force.

1.2 Basics Idea

For fluid simulation, the basic idea has two branches, one is the particle-based method, called the Lagrangian Method, and another is the grid-based method, so-called the Euler Method.

Even taking the different atomic elements, their methods' basic workflow is the same: find a way to measure the pressure in the future and then update the velocity field.

2 Euler Fluid Simulation

If we ignore the viscosity term and use the grid method, then we are doing the Euler Fluid Simulation, which means we are trying to solve this equation:

$$\frac{Du}{Dt} = \vec{g} - \frac{1}{\rho} \nabla p$$

2.1 Skeleton

In this book[**?**], the author separates the whole process into three small steps.

- Preprocess: Compute the safe Δt
- Advection: get the target velocity u^{n+1}
- Projection: compute the pressure and update the new velocity.

2.2 Proprocess

For the safe Δt, we can take advantage of the grids, and choose a flow that can't move more than 1 grid at each time step. Then we have:

$$\begin{cases} u_{max} = max|u^n| + \Delta t|g| \\ \Delta t \leq \frac{\Delta x}{u_{max}} \end{cases}$$

If we plug the second inequality into the first equation, then we get:

$$u_{max} \approx max|u^n| + \sqrt{\Delta x g}$$

Therefore:

$$\Delta t \leq \frac{\Delta x}{max|u^n| + \sqrt{\Delta x g}}$$

2.3 Advection and Projection

We choose to use the flow map method to have a try in solving this equation, but we need to notice that forward make sure the algorithm is unconditionally stable. We can use the Rugge-Kutta method or something else. I think the main idea is that If we imagine a vector field. And the forward Euler method will definitely make the energy become more and more, which breaks the law of physics. For the Euler method, we do something like this:

$$u^{n+1} = u^n + \Delta t f(u^n)$$

The picture is as follows: So, in conclusion, we need to find the next status of the previous position.

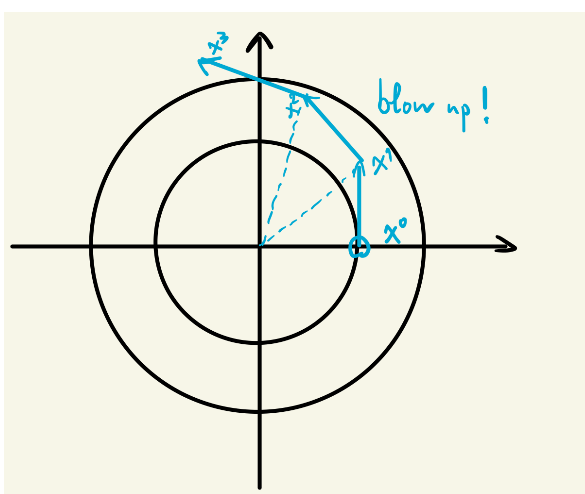

Figure 1: Forward Euler

We only need to substitute the equation for the pressure term and then update it. Then use the finite difference method to solve the partial equation to get the pressure term.

2.4 Boundary: Ghost Grid

One can also put some solid material in the scene. However, the boundary conditions need to be considered carefully. One of the most common tricks is to use the ghost grid. The main point is to set the boundary grid as a valid grid with $v = 0$ m/s and solve the whole system.

2.5 Implementation

Here is my small code base: Code Repository

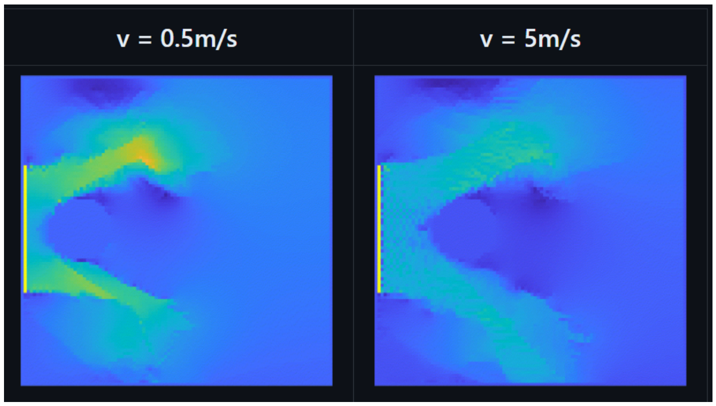

Figure 2: Euler method

3 Particle Method and Hybrid Method

The Euler method is a relatively hard one. The particle method and the hybrid method are more natural when managed using the Euler method. The idea for the particle method(one famous method called SPH), treat the water as many tiny balls and let them push each other. The good thing about this is that you can easily simulate the surface tension. However, it also comes with the drawback, of more computation consumption and inaccurate. To cure this, people later came up with the hybrid method(one is called flip), the idea is to simulate the tiny balls inside the grid. The hybrid method is still the state of the art.
A tutorial for implementation could be this Flip Tutorial.

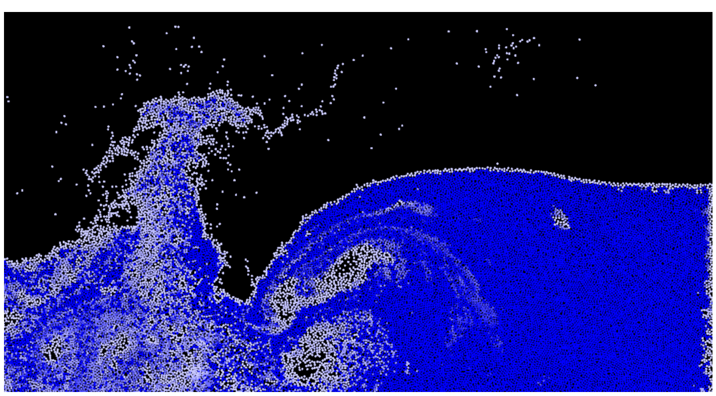

Figure 3: Flip