CSE291E Assignment 4: Fluid Simulation

Xinyuan Liang

June 13 2023

1 Introduction

This project implements the fluid simulation based on the Material Point Method (MPM). MPM is a hybrid Lagrangian-Eulerian method that can simulate a wide range of physical phenomena. It was originally extended from FLIP and takes the simulation framework of (Particle In Cell) PIC / (Fluid-Implicit-Particle) FLIP. A recent development, Moving Least Squares MPM (MLS-MPM) makes MPM more efficient and easier to implement. This project is basically implemented with MLS-MPM.

2 Overall

According to the paper [1], the essential steps in the MLS-MPM implementation in this project are summarized as the following:

- Particles to grid (P2G). Use Affine Particle-In-Cell Method (APIC) to transfer mass and momentum from the particles to the grid.
- Update grid momentum. Use the symplectic Euler to update grid momentum.
- Grid to particles (G2P). Use APIC to transfer velocities and affine coefficients from the grid to the particles.
- Particle deformation gradient. Update the particle deformation gradient using the MLS approximation to the velocity gradient.
- Update particle plasticity (if there is)

- Particle advection. Update particle position with the new velocities.

The only differences between MLS-MPM and traditional MPM are the force expression in step 2 and the deformation gradient update in step 4. The following sections will explain each step with the MLS-MPM as well as the original MPM method.

2.1 Particles to grid

As mentioned above, MLS-MPM is a hybrid method that picks and chooses between Lagrangian and Euler representations to discretize the domain. It first begins with the Lagrangian approach (particles) and then transfers to the Euler approach (grid) later. In the Lagrangian approach, the physical quantities will be carried by the particle's movement, whereas the physical quantities' changes are stored in the static grid in the Euler approach.

This step is basically updating the grid mass and grid momentum with particle mass, velocities, and spatial position. As mentioned in [3], the APIC method is recommended for particle grid transfer due to its nice numerical properties.

$$m_i = \sum_p w_{ip} m_p \tag{1}$$

$$m_i \mathbf{v_i} = \sum_p w_{ip} m_p (\mathbf{v_p} + \mathbf{B_p}(\mathbf{D_p})^{-1}(\mathbf{x_i} - \mathbf{x_p})) \tag{2}$$

where

$$\mathbf{C_p^n} = \mathbf{B_p^n}(\mathbf{D_p^n})^{-1} \tag{3}$$

according to [2]. The $\mathbf{C_p^n}$, or the affine momentum matrix, is extended from the idea of enriching velocities representation, and idealizing the velocity as locally affine on each particle. And $\mathbf{D_p^n}$ is analogous to an inertia tensor which is given by

$$\mathbf{D_p^n} = \sum_i w_{ip}^n (\mathbf{x_i} - \mathbf{x_p^n})(\mathbf{x_i} - \mathbf{x_p^n})^T$$

. Conveniently, when using quadratic interpolation, the $\mathbf{D_p^n} = \frac{1}{4}\Delta x^2 \mathbf{I}$. Before updating the momentum, a bit of difference occurs in MLS-MPM, which fuses the scattering of the affine momentum with the particle's force contribution as: $N_i(\mathbf{x_p^n})\mathbf{Q_p}(\mathbf{x_i} - \mathbf{x_p^n}))$ where

$$\mathbf{Q_p} = \Delta t V_p^0 M_p^{-1} \frac{\partial \Phi}{\partial \mathbf{F}}(\mathbf{F_p^n})\mathbf{F_p^n}^T + m_p \mathbf{C_p} \tag{4}$$

The V_p^0 is the initial particle volume and $N_i(\mathbf{x_p^n})$ is a kernel function (quadratic kernel is used in this project). $M_p = \frac{1}{4}\Delta x^2$ form quadratic kernel. Also, $V_p^0 \frac{\partial \Phi}{\partial \mathbf{F}}(\mathbf{F_p^n})\mathbf{F_p^n}^T = V_p^n \sigma$ with the weak form. The

2.2 Update Grid

Updating the grid momentum is basically derived from dividing the momentum with grid mass, which is then applied to explicit Euler to get the new grid velocities.

2.3 Grid to particles

G2P is the reverse transformation of P2G, using the affine matrices $\mathbf{B_p^{n+1}}$ to update the new particle velocities.

$$\mathbf{v_p} = \sum_i w_{ip} \mathbf{v_i} \tag{5}$$

$$\mathbf{B_p} = \sum_i w_{ip} \mathbf{v_i} (\mathbf{x_i} - \mathbf{x_p})^T \tag{6}$$

Then update the new $\mathbf{C}_p$ with the equation (3). Also, the particle deformation gradient is updated with:

$$\mathbf{F_p^{n+1}} = \mathbf{F_p^n} + (\Delta t \mathbf{C_p^{n+1}})\mathbf{F_p^n} \tag{7}$$

2.4 Particles advection

Finally comes to advecting the particle with their new velocities

$$\mathbf{x_p^{n+1}} = \mathbf{x_p^n} + \Delta t \mathbf{v_p^{n+1}} \tag{8}$$

when APIC is used.

3 Implementation

The project is implemented heavily based on the code from Yuanming Hu and Nialltl's blog. The script is written in Taichi, which is a productive and efficient simulation tool. Press 'R' can reset the default fluid scene. Please see the command hint for other scenes.

References

[1] Yuanming Hu, Xinxin Zhang, Ming Gao, and Chenfanfu Jiang. On hybrid lagrangian-eulerian simulation methods: practical notes and high-performance aspects. In *ACM SIGGRAPH 2019 Courses*, page 16. ACM, 2019.

[2] Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. The affine particle-in-cell method. *ACM Trans. Graph.*, 34(4), jul 2015.

[3] Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. The material point method for simulating continuum materials. In *ACM SIGGRAPH 2016 Courses*, SIGGRAPH '16, New York, NY, USA, 2016. Association for Computing Machinery.