

# Interaction of Fluid Simulation Based on PhysX Physics Engine

Huibai Wang, Jianfei Wan, Fengquan Zhang

College of Computer, North China University of Technology, Beijing, China

wanghb@ncut.edu.cn, wjf0924@sina.com, fqzhang@ncut.edu.cn

**Keywords:** fluid simulation, PhysX physics engine, SPH, interaction of fluid and solid

**Abstract.** To enhance the real-time and interactivity of fluid simulation, a method was proposed based on smooth particle hydrodynamics (SPH) technique in PhysX physics engine. According to SPH method, the fluid was treated as a particle system, analyzing composite force of each particle, and simulating real motion and interaction with static solid and dynamic solid of the fluid on physical environment. Experimental results show that this method satisfy the requirement of real-time simulation and the problem of interaction between fluid and solid was also solved.

## Introduction

With the study of fluid simulation in recent years, fluid-solid interaction has become a hot research topic, which is also a difficult problem in computer graphics and virtual reality. It is a complex technology to simulate the interaction between fluid and solid, because the fluid has no fixed shape, and the partial differential equation of time and space where the force of motion and interaction is converted into is extremely difficult to realize in program. Therefore, the sense of reality and computational efficiency are important criterion in the whole simulation process, however, it is highly difficult to have these two properties in simulation.

## Related Work

The fluid simulation method mainly consists of Euler and Lagrange method. The Euler method is a fluid simulation method based on grid, but it would consume excessive resources, and easily lead to large error because of the complex calculation process dealing with large deformation object. Lagrange method is a method based on particle system, the fluid is treated as lots of particles, and each particle has its own unique properties, such as position, speed, etc. The Smooth Particle Hydrodynamics (SPH) is a typical Lagrange method, which is simple and efficient in the simulation process, and is widely used in various fluid simulation and interaction.

Sun [1] presented a new simulation method for interactions of fluids and deformable solids was. In this way, both fluids and solids are described by SPH so that avoids complex interface for coupling fluids and solids. Based on SPH method, a search method by list was proposed, which accelerates searching the nearest neighbor particles and improves computational efficiency [2]. In order to simulate the water and water droplets, Kim [3] proposed a regional level sets with a mix of grid and particle, but this method is difficult to naturally complete the transition of grid and particle. Lenaerts [4] added number of absorbent particles to each solid particles, simulating the interaction between permeable materials and water using Darcy's law. Akinci [5] presented a general coupling method based on solid particle size, which has the characteristics of good reality and high efficiency.

In this paper, we mainly study the simulation method based on smooth particle hydrodynamics technique in PhysX physics engine, which can maintain the high fluency in the case of large particles.

## Introduction of SPH Method

**Principle of SPH.** Smooth Particle Hydrodynamics is a interpolation method based on fluid dynamics, which can calculate the physical properties of fluid by interpolation. Navier-Stokes equation, an important equation describing the motion of incompressible fluid in fluid dynamics, has become the

basic equation in SPH method. This equation describes the conservation of momentum of incompressible fluid and is equivalent to the Newton's second law. The equation is given as

$$\frac{Du}{Dt} = -\frac{1}{\rho} \Delta p + \mu \Delta^2 u + f \quad (1)$$

where  $\rho$  denotes the density,  $u$  the velocity,  $\mu$  the viscosity coefficient,  $f$  the external force (such as gravity or force of collision).

In SPH method, every particle carries individual properties. The scalar  $A(r)$  at any position  $r$  is a field variable calculated in the Eq. 2:

$$A(r) = \sum_i m_j \frac{A_j}{r_j} W(r - r_j, h) \quad (2)$$

where scalar  $A(r)$ , which is estimated by a weighted sum of neighboring particles  $j$ , can be an arbitrary property of continuous fluid, such as density, pressure, etc.,  $m_j$  is the mass. The function  $W(r - r_j, h)$ , called kernel function, is a smoothed kernel with the smoothed radius  $h$  which indicates the effective range of the function. This function is normalized that has the following properties:

$$\int W(r - r_j, h) dr = 1 \quad (3)$$

The left value  $Du/Dt$  in Navier-Stokes equation is the acceleration for a given particle. Therefore, for a given particle, calculating its density and the right values in Navier-Stokes equation firstly, then calculating resultant force putting on the particle so that the position of the given particle can be updated according to the kinematic mechanics, this is the principle of SPH.

**Basic Idea of SPH.** SPH method can be used to solve the partial differential equations of various energy fields in the fluid dynamics problem. The core idea of SPH algorithm is as follows:

- 1) The problem domain is represented by a series of arbitrarily distributed particles that do not need any connection between them.
- 2) The field function is approximatively indicated by integral method, which is called kernel approximation.
- 3) The kernel equation need to be further approximated using kernel approximation method.
- 4) The given particles depend on particles in current local distribution within each time step.
- 5) The kernel approximation method is applied to the filed function of all the partial differential equations in order to obtain a series of discretization of ordinary differential equations which only relate to time.
- 6) Explicit integration method is applied to solve the ordinary differential equations in order to obtain the fastest time integration, and get values of the field variables of all the particles with time.

These six points are combined, which makes SPH algorithm to be a dynamic problem solver with no grid, adaptive, stable and Lagrange properties.

## Fluid Simulation in PhysX Engine

**Introduction of PhysX Engine.** PhysX engine is one of three major physical computing engine in the world. It has excellent parallelism and can be carried out quickly on GPU in order to deal with the integer and floating-point calculation in physical computing. Therefore, PhysX engine can be used to handle large computational simulation of physical effects, such as fluid dynamics.

In PhysX, all physical motion simulation are completed in a scene, each of which is involved in the physical computation is actor, and physics is a controller computing and finishing the part of interactions.

**Fluid Interaction With Static Solid.** In the fluid space, particle of fluid enters into the static solid space, and the reactive force is exerted by the solid which will cause the particle to return to fluid space needs to be calculated after interaction. The calculation of the force is defined as

$$F = [k_s d + k_d (v \cdot n)]n \quad (4)$$

where  $d$  denotes the distance of interaction between particle and static solid,  $K_s$  and  $K_d$  are respectively the elastic constant and damping constant,  $v$  the relative velocity,  $n$  the normal vector in the point of collision,  $F$  is the reactive force in the direction of normal vector acting on the fluid particle. Added the result of dividing the force by mass of particle to the external force term  $f$  in Navier-Stokes equation and the acceleration under this reactive force can be calculated according to the Newton's law of motion.

**Fluid Interaction With Dynamic Solid.** The two-way coupling calculation of fluid to solid and solid to fluid is required when fluid particle reaches the collision distance. The fluid particle produces a force that causes the solid to move or rotate, simultaneously the solid exerts a reactive force on fluid particle forcing it to leave the solid.

The process of updating state of solid is described as follows:

- 1) Obtain the speed in initial state.
- 2) Solve the forces exerting by each particles.
- 3) Calculate the external force.
- 4) Calculate the acceleration of the solid after collision.
- 5) Update the velocity and position of solid.

**Motion Simulation of Fluid.** The motion of fluid is followed by Newton's second law which determines the relationship of mass, acceleration and resultant force of each particles. The force analysis of fluid particles, including the gravity, the force exerted by external object and the constraining force between adjacent particles, can be easily divided into the internal force and the external force. Therefore, the resultant force of the fluid particles is equal to the sum of all force at the moment  $t$ , and we can get

$$m \frac{\partial^2 S}{\partial t^2} = F_{in}(S, t) + F_{ex}(S, t) \quad (5)$$

where  $m$  denotes the mass of particle,  $S$  the displacement vector,  $F_{in}$  is the internal force and  $F_{ex}$  is the external force and both of these two forces are functions of displacement and time changing with  $t$  and  $S$ .

In the process of motion, fluid have to be detected collision in each frame so that it can interact with other objects. Updating motion trajectory of each particles after collision can form the motion effect of fluid. In PhysX, these collision detection is carried out in each time slice, and the specific detection process is as follows:

1) Collision filtering. In the scene, all objects involved in physical computing can collide, so it will cost a lot of time to detect all objects in the case of vast objects. Collision filtering is based on the space division, only to detect other objects near the current object can save plenty of time and improve the computing efficiency.

2) Rough collision detection. The fast detection to the collision pairs filtered in step of collision filtering based on axis-aligned bounding box method is carried out to filter the collision pairs again.

3) Precise collision detection. The precise detection based on triangle is applied to the rest of collision pairs and the calculation of interaction will be handled if the collision is occurred.

4) End of collision detection.

In the calculation of interaction after collision, the displacement value can be calculated according to Eq. 5.

The main steps of fluid motion simulation in PhysX are given by Fig. 1. Firstly, we should initialize properties of fluid and start the simulation loop. Then the collision detection will be carried out and calculating resultant force and updating velocity and position of fluid particles and dynamic solid are executed. Finally render fluid particles and solids and continue the simulation loop.

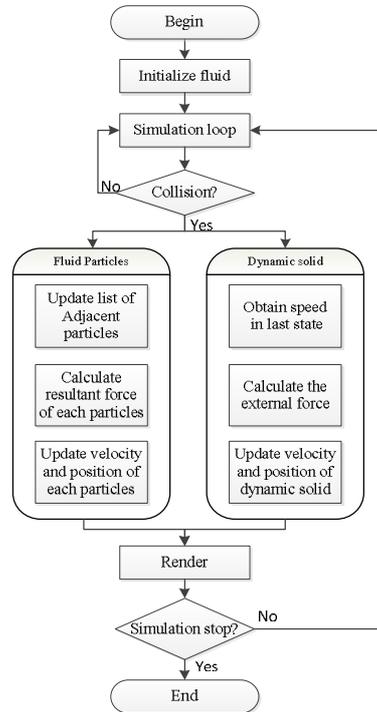


Fig. 1. The main steps of fluid motion simulation

## Results and Discussions

All experiments in this paper were conducted on a PC with an Intel Xenon E5560 2.13GHz processor, 12GB RAM, and an NVIDIA GeForce GTX 480 graphics card. The version of PhysX 3.3.2 is used.

Table 1: Parameters for experiments

Parameters	Values
Smoothed Radius	0.1
Viscosity	60.0
Dynamic Friction	0.002
Restitution	0.1
Mass of Particles	0.00017269

In the experiments, fluid simulation is carried out under the PhysX physics engine, which can make the fluid interact with the static solid and the dynamic solid. Some parameters used in the experiments are shown in Table 1.

The results of interaction between fluid and static solid are shown in Fig.2. When the fluid particles collide with static solid, the reactive force exerted by solid forces particles to leave, while the solid still remain in the original state.

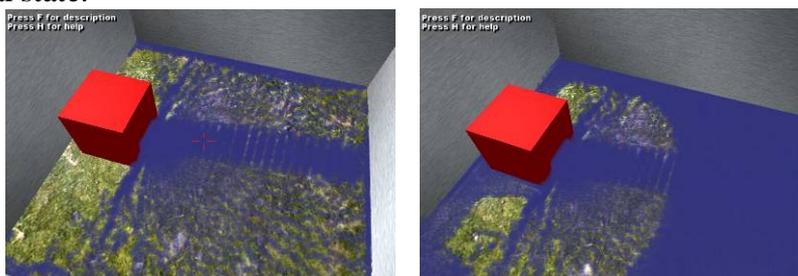


Fig. 2. Interaction between fluid and static solid

The results of interaction between fluid and dynamic solid are shown in Fig.3. When the fluid particles collide with the dynamic solids, solids are also affected by the fluid particles, which can be seen from the figures that the stacked blocks are pushed down and the original state is changed.

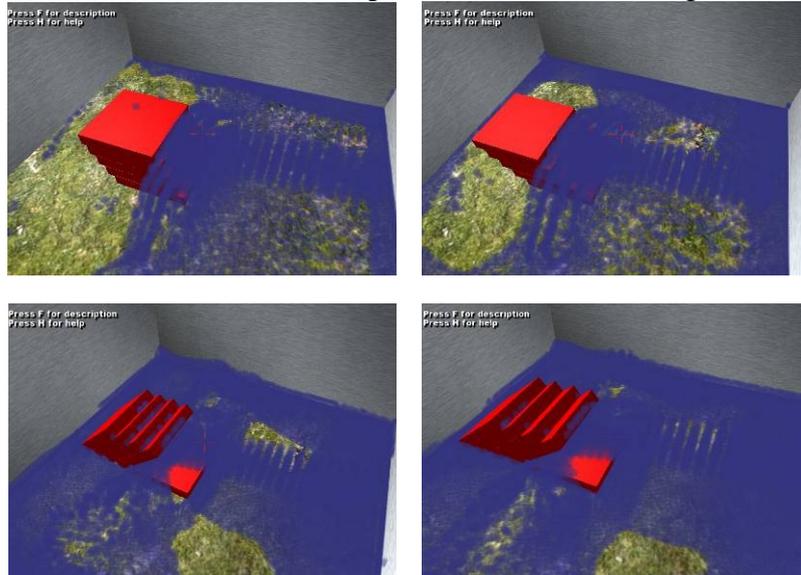


Fig. 3. Interaction between fluid and dynamic solid

Table 2 shows the relationship between the number of particles and the average frame rate, which represents the efficiency of the program. As can be seen from Table 2, when the number of particles is 15000, the calculating efficiency is high enough which can satisfy the requirements of real-time simulation.

Table 2 Numbers of particles and average frame rates

Numbers of Particles	Average Frame Rates
1000	84.35
5000	62.26
10000	38.19
15000	22.81
20000	12.01

## Conclusions

We presented a new method for fluid simulation and a experiment based on SPH method is carried out in PhysX physics engine which simulated a process of interaction between fluid and solids. The result of experiment showed that the method proposed in this paper can still maintain high frame rate in the case of a large number of particles which satisfy the requirement of real-time simulation and prove the feasibility of the method. However, we just performed the simulation using point-based rendering instead of fluid surface rendering, therefore, we will render the fluid surface using Marching Cubes method and adding the part of human-computer interaction based on Kinect in order to enhance the interactivity is taken into account.

## Acknowledgements

This research was supported by Project of National Science and Technology Support Plan (2012BAF84F02).

## References

[1] Hongquan Sun: submitted to Journal of Natural Science of Heilongjiang University (2012), in Chinese.

- [2] Yong Tang, Wenjing Zhao, Mengya Lv: submitted to Journal of Chinese Computer Systems (2012), in Chinese.
- [3] Kim B: ACM Transactions on Graphics Vol. 29 (2010), Article No. 175.
- [4] Lenaerts T, Adams B, Dutré P: ACM Transactions on Graphics Vol 49 (2008).
- [5] Akinci N, Ihmsen M, Akinci G, et al: ACM Transactions on Graphics (TOG) Vol 31 (2012).
- [6] Wojtan C, Thürey N, Gross M, et al: ACM Transactions on Graphics Vol 29 (2010).
- [7] Tongde Tan, Zhiguo Gao, Hongling Zhao, et al. submitted to Journal of Zhengzhou University (Engineering Science), in Chinese.
- [8] Akinci N, Akinci G, Teschner M: ACM Transactions on Graphics (TOG) Vol 32 (2013).
- [9] Bender J, Müller M, Otaduy M A, et al, in: *Position-based Methods for the Simulation of Solid Objects in Computer Graphics*, volume 1-22 of Eurographics 2013-State of the Art Reports[C] (2012).
- [10] Solenthaler B, Pajarola R: ACM Transactions on Graphics (TOG) Vol 40 (2009).
- [11] Becker M, Tessendorf H, Teschner M: Visualization and Computer Graphics Vol 15 (2009).
- [12] Kelager M: University of Copenhagen: Department of Computer Science (2006).