

# Incompressible Projective Smooth Particle Hydrodynamics

Jian Chen<sup>1, a</sup>

<sup>1</sup>School of Communication University of China, Beijing 510000, China.

<sup>a</sup>cj1824@cuc.edu.cn

**Abstract.** We have improved the micro-compressibility problem of Projective Fluid by seamlessly embedding the Divergence Correction and Density Correction in the DPSPH method into the Projective Fluid framework. Projective fluid is a fast simulation method introduced by Weiler into fluid simulation for various constraints. The Divergence Correction and Density Correction schemes are deduced to be seamlessly embedded in the PF method and to solve the micro-compressibility problem of the method due to hard constraints. The experiment proves that the algorithm can combine the advantages of the two methods, which has high parallelism and good incompressibility.

**Keywords:** Smoothed Particle Hydrodynamics, incompressibility, implicit Integration, Projective Fluid.

## 1. Introduction

Smoothed Particle Hydrodynamics (SPH) is a well-established fluid simulation method for simulating complex scenes based on particle models. For the first time, Weiler [1] introduced the Projective Dynamic method, which is used to simulate the distribution and flexible body at a high parallel, into the Fluid simulation, known as Projective Fluid (PF), to enrich and accelerate the Fluid simulation. The fluid constraint is derived from the equation of state (EOS) of the relationship between particle pressure and density. The equation of state used is

$$p(x) = \frac{\rho_i(x)}{\rho_0} - 1 \quad (1)$$

Here the stiffness factor is taken,  $\rho_0$  is the initial density of the fluid, and when it is 0, it ensures that the density of the fluid particles is constant. This constant density only guarantees the density of individual particles and lacks global constraints, so this paper introduces an average density correction scheme. The scheme uses the conditions in the DFSPH [2] method as the global judgment condition, and derives the average density formula according to the PF framework. The fourth part will be deduced in detail.

So the particle-based fluid constraints are as follows

$$C_i(x) = \frac{1}{\rho_0} \left( \sum_{j=1}^n m_j W_{ij} \right) - 1 \quad (2)$$

Now we need to find an auxiliary variable to make  $C_i(x)$ . The newly obtained particle position  $p$  is such that the particle density is constant. The position correction is determined by the following formula.

$$p = p - \frac{C}{\|\nabla C\|} \nabla C \quad (3)$$

In the actual calculation, iteratively updating until  $C$  is a small constant. Experiments have shown that an average of three to four iterations can be achieved. The choice of constraints here approaches a positive value, which avoids the problem of free surface particle defects, which is often handled in free surface SPH, for example [3].

However, as the particle speed is updated later, the speed accumulation error will increase over time. Therefore, this paper introduces a divergence correction scheme at the update speed, so that the particles satisfy both the divergence free and the constant mass condition, and the particle density remains unchanged with time. Speed divergence correction is only used in a small number

of SPH simulation methods, but they are mostly micro-compressible or cannot maintain low-density errors, such as Predictive-Corrective Incompressible SPH (PCISPH) [5], Implicit Incompressible SPH (IISPH) [6]. The DFSPH [2] method converts the velocity divergence condition to the density-time derivative, which satisfies both the density constant and the divergence free condition, so we use this idea to give our divergence correction scheme. Specifically elaborated in the third part.

## 2. Related Work

As GPU processing speeds up, fluid simulation based on physical motion equations is further improved in both real-time and authenticity. Among them, because of the ability to simulate more subtle and more complex fluid surfaces, the SPH-based Lagrangian method has received a lot of attention from scholars.

For the first time, Monaghan [22] introduced the concept of particle-based fluid simulation into the field of computer graphics. The SPH method was later extended by Adams [23] to be discretized in space. In low viscosity liquids, such as water, the incompressibility of the liquid is very important in the simulation, so various incompressible method of ensuring fluid volume develop. Weakly compressible SPH [24] is a method proposed by Becker and Teschner that relies on the scene and user-defined stiffness coefficient to calculate the force and ensure that the volume is incompressible. This is a time-specific integration approach. Subsequently, Solenthaler and Pajarola [5] proposed that the hermit time integration method iteratively calculates the particle pressure using a predictive correction scheme. The Hermit Time Integration approach is more flexible and more versatile, but only ensures constant density. Bender and Koschier [25] first considered the divergence-free velocity field to make the incompressibility method more diverse. Bender and Weiler [1] first introduced Projective Dynamic in software and fabrics to fluid simulation, and proposed the fluid constraints based on the pressure-based equation of state (EOS) to further enrich the fluid simulation method. This paper is based on the previous method to improve the micro-compressibility problem in the PF method, and proposes a divergence correction solver suitable for its framework.

## 3. Divergence Correction Solver

The divergence free condition is  $\nabla \cdot u = 0$ , this is the mass conservation equation of fluid motion equation Navier-Stokes Equations [4], describing the particle velocity field satisfies the divergence free. From the continuous equation  $\frac{D\rho}{Dt} = -\rho \nabla \cdot u$ , the divergence correction cycle judgment condition can be derived.

$$\frac{D\rho}{Dt} = 0 \quad (4)$$

For the particle model, the continuous equation of density versus time derivative [8] can be organized into

$$\frac{D\rho_i}{Dt} = \sum_j m_j (v_i - v_j) \nabla W_{ij} (\mathbf{p}_i - \mathbf{p}_j) \quad (5)$$

This formula takes into account the weighted effect of particles in the particle domain on their values, where the kernel function is calculated using the projected position of the particle. The projection position is the position of particles corrected by constraint conditions, where the projection position can ensure the constant density. The density convection term calculated by using this position can satisfy both the density and divergence errors.

In order to satisfy the cyclic judgment condition, we determine the corresponding pressure for each particle  $i$  to correct the divergence error in the particle field. The pressure is determined by the following formula

$$f_i^p = -\nabla p_i \quad (6)$$

The pressure divergence is derived from equation (1).

$$\nabla p_i = \frac{1}{\rho_0} \nabla \rho_i = \frac{1}{\rho_0} \sum_j m_j \nabla W_{ij} \quad (7)$$

Therefore, the speed update formula determined by this pressure term is

$$v_i^* = v_i^* - \Delta t * \alpha * \kappa * \sum_j m_j \nabla W_{ij} \quad (8)$$

The divergence correction scheme algorithm is shown in Algorithm 1, which ends the loop when the average density convection term is less than a given threshold. When calculating the particle pressure term, we introduce a constant that is user-defined to determine how much the pressure term affects the particle. In the simulated Dam Breaking experiment, when, the experimental results have a good divergence and run faster.

---

Algorithm1 myDivergence-Correct solver

---

```

1: function DivergenceCorrection( v * )
2:   while( ( (Dρ/Dt)avg > η ) && (iter < 1) ) do
3:     for all particle i do
4:       vi* = vi* - Δt * α * κ * ∑j mj ∇Wij
5:     for all particle i do
6:       Dρi/Dt = ∑j mj (vi - vj) ∇Wij (pi - pj)

```

---

#### 4. Incompressible Projective

---

Algorithm 2 the global average density correction algorithm

---

```

1: function AverageDensityCorrection(p)
2:   while ρavg - ρ0 > η do
3:     for all fluid models
4:       C ← calcConstraintPotential(p)
5:       while C > ε do
6:         ∇C ← calcConstraintPotential(p)
7:         if ||∇C|| is 0 then
8:           break
9:         p ← p - C / ||∇C|| ∇C
10:        C ← calcConstraintPotential(p)
11:        ρavg = 1 / numParticles ∑i ∑j mj Wij (pi - pj)
12:      return p

```

---

From the introduction of the first part, we can see that the particle projection position after the fluid constraint condition is performed can satisfy the particle density, but the overall fluid block cannot determine the density constant due to the value accumulation error, so we embed a layer of judgment conditions of average fluid density, in the whole fluid model. Is the static density of the fluid block, and the formula for calculating the average density of the fluid block is?

$$\rho_{avg} = \frac{1}{numParticles} \sum_i \sum_j m_j W_{ij} (p_i - p_j) \quad (9)$$

As with the divergence correction scheme, the kernel function is also calculated using the particle projection position. The algorithm stops when the difference between the average density of

the fluid and the resting density is less than a given threshold. The algorithm simultaneously computes the projected position of each particle in parallel and returns to the particle position variable.

## 5. Implementation Details

Our simulation framework is implemented on the SPLishSPlasH [7] open source library provided by Bender and its lab. The open source library is a fluid simulation engine based on OpenGL and C++. The library contains five popular SPH fluid simulation methods (DFSPH [2], IISPH [6], PBF [8], PCISPH [5], PF [1]) and three surface tension processing methods (Becker 2007 [9], Akinci 2013 [10], He2014 [11]) and six viscosity term processing methods (standard [12], XSPH [13], Bender 2017 [2], Peer 2015 [14], Peer 2016 [15]) and two methods of curl processing (MicropolarModel\_Bender2017 [16], VorticityConfinement [19]) and two types of DragForce (Macklin 2014 [17], Gissler 2017 [18]).

We use a parallel hash algorithm to speed up the domain search of particles [20], and the density calculation uses the cubic spline kernel function [21]. The viscosity of the XSPH is used to simulate the low-viscosity object of water [13].

## 6. Results

In this section we will list the simulation results, including the simulation scenarios that are common in fluids. Compare and analyze the advantages and disadvantages of our method (Incompressible Projective Smooth Particle Hydrody, IPSPH) and DFSPH and PF methods.

The dam scene is a commonly used test scenario in fluid simulation. Fig. 1 shows the overtaking wave in the dam scene. From left to right, a wave is rolled forward. In all the results plots we use particle velocity to calibrate the color of the particles, blue is the minimum velocity, and white is the maximum velocity. Fig. 2 Double dragons scene, simulating the interaction of fluids with complex boundaries, can be seen to form a good crushing effect at the interaction. There is a total of 28,899 fluid particles in the scene.

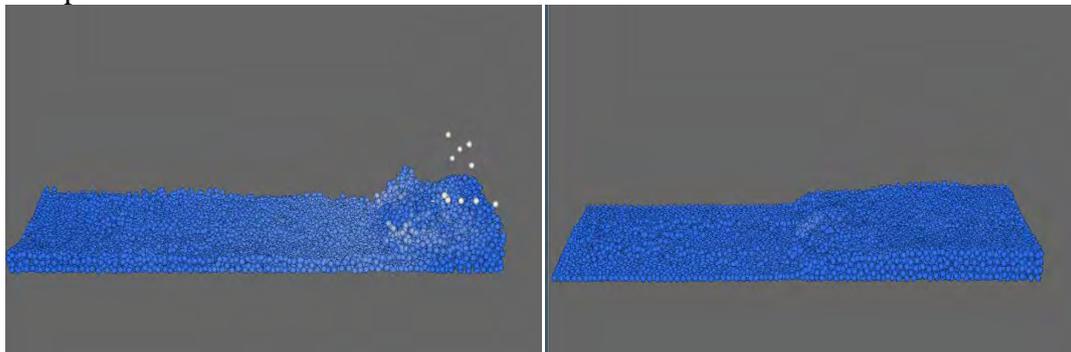


Fig. 1 A breaking dam to show overtaking wave

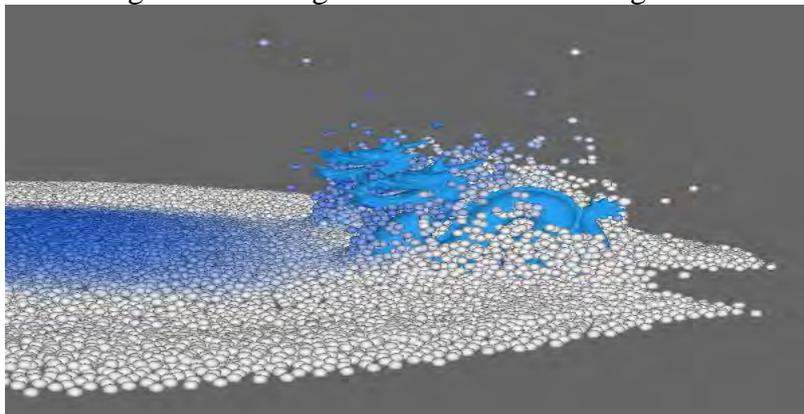


Fig. 2 Double dragons scene to show Complex boundary interactions

Fig. 3 shows the effect of the collision of two fluid blocks, simulating the classic water splash and flake. The left image forms a thin layer of water at the junction of the two water blocks. The right image shows the final state of the interaction between the two blocks, with splashes of water particles on the diagonal.

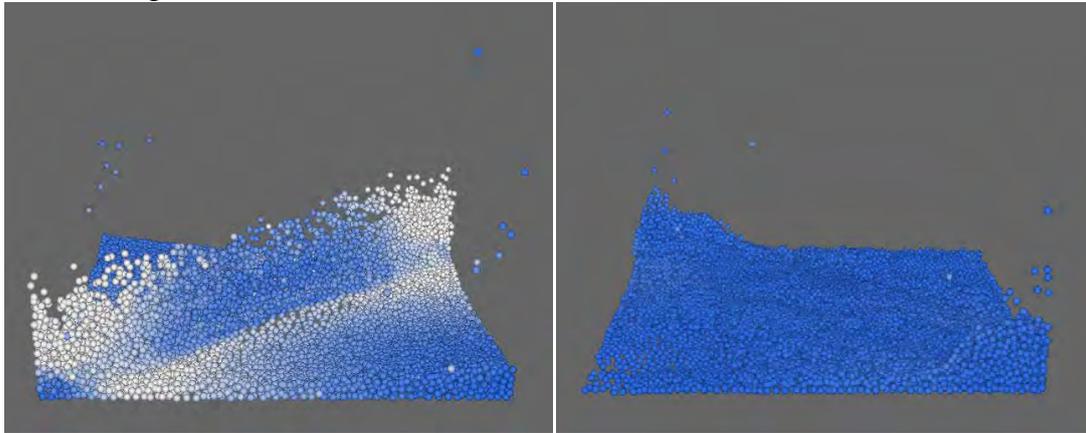


Fig. 3 Double dam break to slice and splash

In this paper, the density constant scheme and the divergence free scheme of DFSPH are improved and seamlessly embedded into the PF framework, so next we will list the average time used for the divergenceSolver, the average time used by the solvePDConstraint, and the average time used for each simulation step of each method. time. As shown in Table 1, the average simulation time of each method in our method is less than that of the DFSPH method in three scenarios, and the time used by the PF is similar. This proves that the method inherits the parallelism and high efficiency of the PF method. In the Double Dam sence, as shown in Fig. 3, the average calculation time of the divergenceSolve in this method is smaller than the original method DFSPH. In the TwoDragons scene, as shown in Fig. 2, and the Breaking dam scene, as shown in Fig. 4, the average calculation time of divergenceSolve in this method is slightly larger than the original method, which shows that the divergence correction method introduced in this paper does not increase the computational power consumption.

Table 1 The time measurement of IPSPH、DFSPH、PF

Scene (fluid Particles number)	IPSPH			DFSPH		PH	
	Avg.time divergen ceSolve	Avg.time solvePDCon straints	Avg.time per step	Avg.time divergenceS olve	Avg.time per step	Avg.time solvePDCon straints	Avg.time per step
TwoDragons (28899)	38.4863 ms	45.9366 ms	79.9104 ms	36.19 ms	97.2973 ms	39.7688 ms	73.615 ms
Double dam(4228)	22.9544 ms	70.9681 ms	102.491 ms	37.6388 ms	104.015 ms	67.7487 ms	98.262 ms
Breaking dam(24389)	22.7259 ms	54.158 ms	71.5683 ms	21.1082 ms	93.3378 ms	55.6955 ms	72.7622 ms

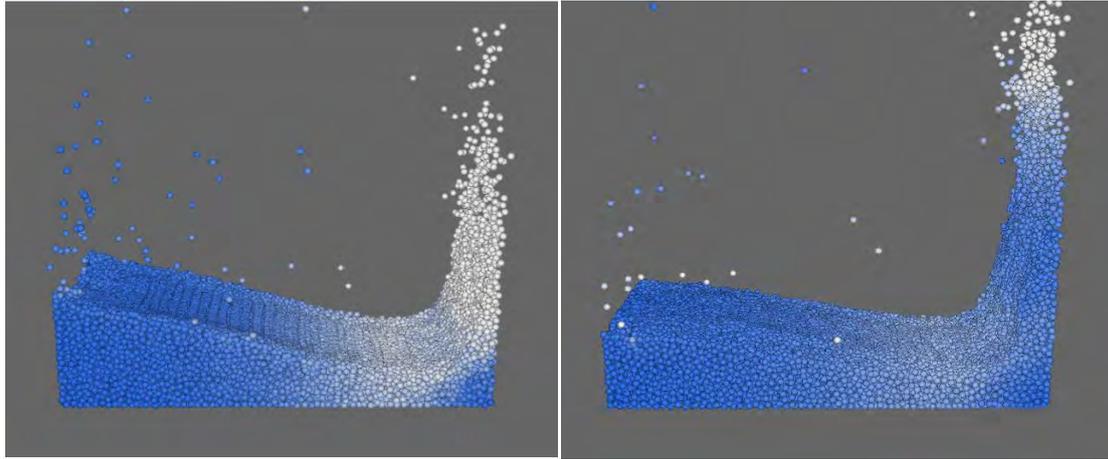


Fig. 4 Breaking dam,IPSPH(left) and PF(right)

The incompressibility of the fluid directly affects the authenticity of the fluid animation. If the particle velocity divergence at the free surface satisfies the mass conservation equation well, it will reveal the true fluid surface. From the two figures in Fig. 4, we can clearly compare the particles on the right side of the dam with a certain collapse on the upper surface of the dam, while the particle movement range in the left picture is larger. This shows that our method can solve the problem of micro-compressibility in the PF method and ensure the divergence free.

## 7. Conclusion and Future Work

In this paper, we first enhance the incompressibility of the fluid simulation method (PF) to simulate a wider range of incompressible fluids. The DFSPH scheme is used to derive the divergence correction scheme and density correction scheme applicable to the framework of this paper. Experiments prove that this paper inherits the advantages of both, which can meet the incompressibility and high parallel processing ability.

In the future, we will explore the application of this method in high viscosity fluid simulation and propose a viscosity term processing model suitable for this framework. On the other hand, it will further speed up the processing power of the algorithm and process performance on the GPU.

## References

- [1]. Weiler, Marcel; Koschier, Dan; Bender, Jan: Projective Fluids. [Online-Edition: <http://doi.acm.org/10.1145/2994258.2994282>] In: Proceedings of the 9th International Conference on Motion in Games. MIG '16. ACM, New York, NY, USA New York, NY, USA, pp. 79-84. ISBN 978-1-4503-4592-7 [Buchkapitel], (2016).
- [2]. Bender J, Koschier D: Divergence-Free SPH for Incompressible and Viscous Fluids,IEEE Trans Vis Comput Graph. 2017 Mar; 23(3):1193-1206. doi: 10.1109/TVCG.2016.2578335. Epub 2016 Jun 8.
- [3]. IHMSEN, M., CORNELIS, J., SOLENTHALER, B., HORVATH, C., AND TESCHNER, M. 2014. Implicit Incompressible SPH. IEEE Trans. on Visualization and Computer Graphics 20, 3, 426 - 435.
- [4]. A. J. Chorin, "Numerical Solution of the Navier-Stokes Equations," Mathematics of Computation, vol. 22, no. 104, p. 745, 1968.
- [5]. B. Solenthaler and R. Pajarola, "Predictive-corrective incompressible SPH," ACM Transactions on Graphics, vol. 28, no. 3, pp. 40:1 - 40:6, 2009.

- [6]. M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner, “Implicit incompressible SPH,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, pp. 426 – 435, 2014.
- [7]. [http://www.interactive-graphics.de/SPlisHSPlasH/doc/html/class\\_s\\_p\\_h\\_1\\_1\\_time\\_step.html](http://www.interactive-graphics.de/SPlisHSPlasH/doc/html/class_s_p_h_1_1_time_step.html)
- [8]. M. Macklin and M. Muller, “Position Based Fluids,” *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 1 – 5, 2013.
- [9]. M. Becker and M. Teschner, “Weakly compressible SPH for free surface flows,” in *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2007, pp. 1 – 8.
- [10]. N. Akinci, G. Akinci, and M. Teschner, “Versatile surface tension and adhesion for SPH fluids,” *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 1 – 8, 2013.
- [11]. HE, X., WANG, H., ZHANG, F., WANG, H., WANG, G., AND ZHOU, K. 2014. Robust Simulation of Sparsely Sampled Thin Features in SPH-Based Free Surface Flows. *ACM Trans. On Graphics* 34, 1, 7:1 – 7:9.
- [12]. Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. SPH Fluids in Computer Graphics. In Sylvain Lefebvre and Michela Spagnuolo, editors, *Eurographics 2014 - State of the Art Reports*. The Eurographics Association, 2014.
- [13]. Hagit Schechter and Robert Bridson. Ghost sph for animating water. *ACM Trans. Graph.*, 31(4):61:1 – 61:8, July 2012.
- [14]. A. Peer, M. Ihmsen, J. Cornelis, and M. Teschner. An Implicit Viscosity Formulation for SPH Fluids. *ACM Trans. Graph.*, 34(4):1 – 10, 2015.
- [15]. Andreas Peer and Matthias Teschner. Prescribed velocity gradients for highly viscous SPH fluids with vorticity diffusion. *IEEE Transactions on Visualization and Computer Graphics*, pages 1 – 9, 1.
- [16]. Jan Bender, Dan Koschier, Tassilo Kugelstadt, and Marcel Weiler. A micropolar material model for turbulent sph fluids. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA '17*. ACM, 2017.
- [17]. Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified Particle Physics for Real-Time Applications. *ACM Trans. Graph.*, 33(4):1 – 12, 2014.
- [18]. Christoph Gissler, Stefan Band, Andreas Peer, Markus Ihmsen, and Matthias Teschner. Approximate air-fluid interactions for sph. In *Virtual Reality Interactions and Physical Simulations*, pages 1 – 10, April 2017.
- [19]. Miles Macklin and Matthias Müller. Position based fluids. *ACM Trans. Graph.*, 32(4):104:1 – 104:12, July 2013.
- [20]. IHMSEN, M., AKINCI, N., BECKER, M., AND TESCHNER, M. 2011. A Parallel SPH Implementation on Multi-Core CPUs.
- [21]. MONAGHAN, J. 1992. Smoothed Particle Hydrodynamics. *Annual review of astronomy and astrophysics* 30, 1, 543 – 574.
- [22]. MONAGHAN, J. 1992. Smoothed Particle Hydrodynamics. *Annual review of astronomy and astrophysics* 30, 1, 543 – 574.
- [23]. ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively Sampled Particle Fluids. *ACM Trans. on Graphics* 26, 3, 48.

- [24]. BECKER, M., AND TESCHNER, M. 2007. Weakly Compressible SPH for Free Surface Flows. In ACM SIGGRAPH / Eurographics Symposium on Computer Animation, 1 – 8.
- [25]. BENDER, J., AND KOSCHIER, D. 2015. Divergence-Free Smoothed Particle Hydrodynamics. In ACM SIGGRAPH / Eurographics Symposium on Computer Animation, 1 – 9.