

A Dual-Particle Approach for Incompressible SPH Fluids

SHUSEN LIU, SKLCS, Institute of Software, Chinese Academy of Sciences and UCAS, Beijing, China

XIAOWEI HE, Institute of Software, Chinese Academy of Sciences, Beijing, China

YUZHONG GUO, Institute of Software, Chinese Academy of Sciences, Beijing, China

YUE CHANG, Peking University, Beijing, China

WENCHENG WANG, SKLCS, Institute of Software, Chinese Academy of Sciences and UCAS, Beijing, China

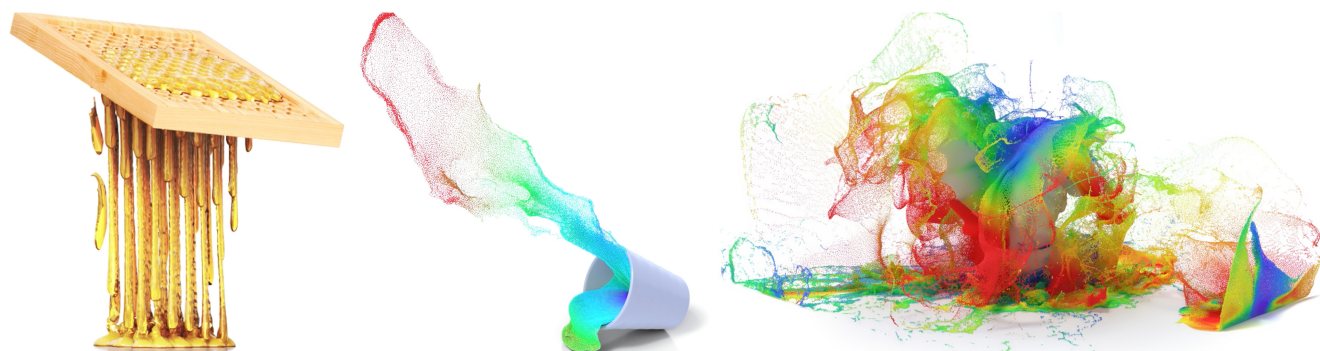


Fig. 1. The dual-particle approach is able to reduce the tensile instability of SPH fluids for well simulating free-surface flows with rich thin detailed features. Left: A high viscous fluid stretches under gravity; Middle: A cup falls onto the ground with water spurting out of the cup; Right: Fluid particles spilled onto the gargyle, producing a lot of thin features under a tensile stress state.

Tensile instability is one of the major obstacles to particle methods in fluid simulation, which would cause particles to clump in pairs under tension and prevent fluid simulation to generate small-scale thin features. To address this issue, previous particle methods either use a background pressure or a finite difference scheme to alleviate the particle clustering artifacts, yet still fail to produce small-scale thin features in free-surface flows. In this article, we propose a dual-particle approach for simulating incompressible fluids. Our approach involves incorporating supplementary virtual particles designed to capture and store particle pressures. These pressure samples undergo systematic redistribution at each time step, grounded in the initial positions of the fluid particles. By doing so, we effectively reduce

tensile instability in standard SPH by narrowing down the unstable regions for particles experiencing tensile stress. As a result, we can accurately simulate free-surface flows with rich small-scale thin features, such as droplets, streamlines, and sheets, as demonstrated by experimental results.

CCS Concepts: • **Computing methodologies** → **Physical simulation**;

Additional Key Words and Phrases: A dual-particle framework, incompressible fluids, tensile instability, smoothed particle hydrodynamics

ACM Reference Format:

Shusen Liu, Xiaowei He, Yuzhong Guo, Yue Chang, and Wencheng Wang. 2024. A Dual-Particle Approach for Incompressible SPH Fluids. *ACM Trans. Graph.* 43, 3, Article 28 (April 2024), 18 pages. <https://doi.org/10.1145/3649888>

1 INTRODUCTION

Since **smoothed particle hydrodynamics (SPH)** was first proposed by Gingold, Lucy, and Monaghan in 1970s [Gingold and Monaghan 1977; Lucy 1977], the tensile instability issue has become one of the major obstacles to prevent the use of SPH in practical applications. Without addressing the tensile instability, particles may tend to clump in pairs under attractive interparticle forces. As a result, traditional incompressible SPH solvers typically fail to produce small-scale thin features (e.g., water streamlets and sheets), especially for free-surface flows where particles near the free-surface boundary mainly endure attractive interparticle forces. To reduce tensile instability in fluid simulation, two strategies are commonly used in computer graphics. One is to use a background pressure to avoid the occurrence of attractive interparticle pressure forces, and the other is to use a scheme similar to the finite difference method to discretize the pressure gradient.

Within the first strategy, a positive background pressure is typically introduced to provide repulsive forces in pairs to

Y. Chang now at University of Toronto. Work was done while at Peking University. The project was partially supported by the National Key R&D Program of China (No.2021YFB1715800), and the National Natural Science Foundation of China (No. 62072446, 62302490, 61872345).

SKLCS: State Key Laboratory of Computer Sciences.

UCAS: University of Chinese Academy of Sciences.

Authors' addresses: S. Liu and W. Wang (Corresponding author), SKLCS, Institute of Software, Chinese Academy of Sciences and UCAS, No.4, South Fourth Street, Zhong Guan Cun, Beijing, Beijing 100190, China; e-mails: liushusen@iscas.ac.cn, whn@ios.ac.cn; X. He (Corresponding author), Institute of Software, Chinese Academy of Sciences, No.4, South Fourth Street, Zhong Guan Cun, Beijing, Beijing 100190, China; e-mail: xiaowei@iscas.ac.cn; Y. Guo, Institute of Software, Chinese Academy of Sciences, No.4, South Fourth Street, Zhong Guan Cun, Beijing, Beijing 100190, China; e-mail: guoyuzhong@iscas.ac.cn; Y. Chang, Peking University, No.5 Yiheyuan Road, Haidian District, Beijing, Beijing 100871, China; e-mail: yuechang@pku.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 0730-0301/2024/04-ART28

<https://doi.org/10.1145/3649888>

regularize the particle distribution. However, several drawbacks prevent its use. Firstly, excessive numerical dissipation can arise from the background pressure, and redundant repulsive pressure forces may destroy the small-scale thin features of SPH fluids [Chalk et al. 2020; He et al. 2014; Vacondio et al. 2021]. Secondly, it cannot be applied in the context of projection-based methods to simulate free-surface flows where a zero-pressure boundary condition is typically imposed at the liquid interface [Colagrossi et al. 2009].

With regard to the second strategy, a variety of different methods have been developed. For example, a Taylor-series consistent pressure gradient model was used in the MPS method [Khayyer and Gotoh 2011] to stabilize the simulation for regions in presence of attractive interparticle forces. Similar ideas have also been introduced in the context of the SPH method for computer graphics [He et al. 2012b, 2020]. Nevertheless, the stability of the simulation performed by pressure gradient models are generally not guaranteed if no additional **dynamic stabilization (DS)** [Macklin and Müller 2013; Tsuruta et al. 2013] schemes are used. Besides, the effectiveness of using pressure gradient models to remove the particle clamping artifact is influenced by a large number of other factors including kernel normalization [He et al. 2020], boundary conditions [Yang et al. 2016], and so on. However, none of those previous methods have touched the essence of the tensile instability problem.

Back to the nature of tensile instability, its occurrence is believed to be arising from the mismatch between the stress state and the kernel function [Swegle et al. 1995]. Belytschko and Xiao [2002] showed that the tensile instability can be completely removed when the particle kernel is defined as a function of the material (Lagrangian) coordinates. While fully Lagrangian kernels are impossible to define in the case of large deformation problems, the occurrence of tensile instability can be attributed to an error in particle approximation when using Eulerian kernels instead of Lagrangian kernels [Belytschko and Xiao 2002]. The question is: *how can we reduce the error in particle approximation when using Eulerian kernels since it is not possible to be completely removed?* We propose a dual-particle framework to address tensile instability in fluid simulation. Aside from original fluid particles, which carry the particle masses and velocities, we introduce additional “stress points” to carry the particle pressures. In the following discussion, we call the original fluid particles as **real particles** while the generated “stress points” as **virtual particles** for simplicity. With virtual particles generated, the Navier-Stokes equation is then discretized with Eulerian kernels defined on both real and virtual particles. Since generating virtual particles follows a custom rule, it allows us to obtain a regular enough distribution of virtual particles, and effectively reduces tensile instability in standard SPH by narrowing down the unstable regions of Eulerian kernels under a tensile stress state. Compared with other particle based approaches, our dual-particle approach is easy to implement, yet can capture rich thin features of fluids without requiring us to clamp negative pressures to zero. Besides, although our method shares similarities with the **hybrid particle-grid (HPG)** method [Fei et al. 2021], the methodology employed in this study relies exclusively on SPH discretization [Koschier et al. 2019], imparting distinct characteristics that differentiate it from the HPG method, as elucidated in Section 6.3.

In summary, we have made the following contributions:

- A dual-particle approach to address the tensile instability in fluid simulation.
- A reformulation of the SPH approximate projection method based on our dual-particle framework, in which the velocity and pressure fields can be defined at different locations.
- A spatially-adaptive strategy to generate virtual particles from real ones in parallel, where the regularity and compactness for the particle distribution are guaranteed.

The outline of this article is as follows. Section 3 first gives a brief introduction of the tensile instability issue followed by our motivation to address the problem. Section 4 demonstrates the numerical implementation of our dual-particle approach in solving incompressible free-surface flows. Section 5 illustrates three typical strategies to generate virtual particles. The remaining sections provide a variety of evaluations and examples to show the superiority of our dual-particle approach over previous methods in terms of suppressing tensile instability and enriching small-scale thin features of SPH fluids.

2 RELATED WORK

In this section, we first summarize previous works on SPH methods that deal with the tensile instability. Then, we present a brief introduction to the HPG methods due to the similarities to our dual-particle approach.

2.1 The Tensile Instability in the SPH Methods

SPH was first invented by Gingold, Monaghan and Lucy to simulate interstellar flows [Gingold and Monaghan 1977; Lucy 1977], and introduced into the computer graphics community by Desbrun and Cani in 1996 [Desbrun and Gascuel 1996]. After this, numerous studies have focused on using SPH to model free surface flows in computer graphics [Ihmsen et al. 2014b; Koschier et al. 2019, 2022]. To remove the severe time step restrictions of the original SPH method [Becker and Teschner 2007; Müller et al. 2003], researchers propose to either use iterative predictive-corrective schemes to enforce a constant density [He et al. 2012a; Solenthaler and Pajarola 2009] or projection-based methods to enforce a divergence-free velocity [Bender and Koschier 2015; Ihmsen et al. 2014a; Takahashi et al. 2018]. To accelerate the convergence rate of the pressure solver, a projection method based on an Eulerian grid is introduced into SPH [Raveendran et al. 2011].

Tensile Instability. Long-term viability poses a significant challenge for SPH due to the persistent issue of tensile instability [Vacondio et al. 2021]. As studied in Swegle et al. [1995], the tensile instability is largely attributed to the conflict between the stress state and the second order derivative of the kernel function. Without the tensile instability being addressed, particles will tend to clump due to attractive interparticle pressure forces. To reduce tensile instability, many approaches have been proposed in recent decades. In the computer graphics community, the most commonly used way is to avoid inter-particle attractive forces, e.g., clamping the negative pressure to zero or to use an artificial background pressure [Bender and Koschier 2015; Ihmsen et al. 2014a; Macklin and Müller 2013; Schechter and Bridson 2012; Si et al. 2018]. Unfortunately, small-scale fluid details can also be removed due

to the absence of negative pressures. In engineering, the “**tension instability control**” (TIC) approach is used to control the value of background pressure [Lyu et al. 2021; Sun et al. 2018, 2017]. However, this may introduce other instability issues, such as particle oscillation, that require us to apply a “stabilizer” to stabilize the simulation [Xu et al. 2009]. Another way to deal with tensile instability is by using the gradient correction scheme, which was first invented by Khayyer and Gotoh [2011] and introduced to the computer graphics community by He et al. [2012b; 2020]. Moreover, the incorporation of pressure gradient estimation, addressing non-momentum conservation as proposed by Sun et al. [2018], proves advantageous for enhancing the tensile stability of the particle method. However, without completely removing the collocation nature of standard SPH, its effectiveness in reducing tensile instability is less persuasive (see Figure 10). According to the discussion in Belytschko and Xiao [2002], the tensile instability is a kind of numerical error introduced by particle approximation equipped with Eulerian kernels. Therefore, a totally Lagrangian formalism (named as TLSPH) [Belytschko and Xiao 2002; Sun et al. 2021; Zhan et al. 2019] that uses a kernel function of material coordinates is considered to be another effective way to avoid tensile instability. However, these fully Lagrangian methods are difficult to model fluids due to their complex motions. **Zero-energy mode.** Researchers are also aware when all field variables as well as their derivatives are defined at the same locations in the SPH method, the zero-energy mode is another numerical issue that aggravates the tensile instability. Therefore, additional stress points are introduced to enhance the numerical stability by removing the collocation nature in SPH [Chalk et al. 2020; Randles and Libersky 2000]. However, most previous SPH methods based on stress points can only be used to model material with small deformations since stress points and velocity particles need to remain interleaved.

Small-scale features. Instead of addressing the tensile instability directly, some works aim at enhancing the visual quality of small-scale features of the fluid with other strategies. To correct the density estimates at the free surface, Schechter et al. [2012] suggested to create ghost particles around the fluid surface; To avoid numerical instability on thin features of fluids, He et al. [2014] proposed to estimate the pressure at two scales and introduced the anisotropic kernel to filter the pressure force. Furthermore, a number of surface tension algorithms have been developed [Akinci et al. 2013; He et al. 2014; Yang et al. 2017a, b], which can minimize fluid surface area and model the splashes of fluids. However, without the underlying tensile instability being addressed, these techniques have limited improvements on modeling the small-scaled thin features in fluids.

2.2 Hybrid Particle-Grid Methods

While pressure projection is preferred to be done on an Eulerian grid [Batty et al. 2007; Chen et al. 2020b; Larionov et al. 2017] and material tracking on Lagrangian particles [Chen et al. 2020a; Gissler et al. 2019], early studies on the HPG methods aim to attain both the benefits by using a hybrid strategy [Harlow 1962; Harlow and Welch 1965]. Since the standard PIC suffers from significant dissipation due to a direct interpolation between particles and grids, Brackbill et al. [1986] proposed the **Fluid-Implicit-Particle**

(FLIP), whose idea is to only transfer velocity changes from the grid to particles. In 1996, the HPG method was first introduced to computer graphics to simulate fluids [Foster and Metaxas 1996]. Recently, Jiang et al. [2015] proposed APIC, which replaces the local velocity field of particles with an affine velocity field to preserve rotational and shearing motions. Fu et al. [2017] further extended their work and presented the Poly-PIC method to better preserve the energy and vorticity. Hu et al. [2018] introduced a generalized form of the APIC and Poly-PIC methods by using a Galerkin-style **Moving Least Squares (MLS)** discretization. Nakashi et al. [2020] proposed a new PIC-like solver by integrating RBF-FD (Radial Basis Function-Finite-Difference), which gives a higher-order scheme for velocity transfer between the grid and particles. Fei et al. [2021] proposed a new integration scheme that can effectively reduce diffusion and dissipation of interpolations.

In terms of enhancing the splashing quality of fluids, Ando proposed a method to effectively improve the stability of small-scale details by increasing the particle resolution in the splashing area. Ando et al. [2012]. Nonetheless, the present challenge lies in the complexity of parallelizing the adaptive particle resolution strategy on GPUs to attain superior computational efficiency. Gerszewski and Bargteil [2013] introduced the mass-full FLIP method, which integrates unilateral incompressibility to simulate large fluid splashes. However, they observed that simulating fluid thin sheets presents a challenge for their method.

Generally speaking, the HPG methods should have the benefits of both Eulerian and Lagrangian methods. However, compared with a fully Lagrangian method, the HPG methods typically appear to be more dissipative in simulating fluids [Fei et al. 2021]. Besides, it is also not quite easy to preserve the whole volume for hybrid particle-grid methods. Therefore, Cornelis et al. [2014] proposed to combine the Lagrangian IISPH [2014a] method and FLIP method [1986] to better preserve the volume.

3 A DUAL-PARTICLE FRAMEWORK

In this section, we will present the basic theory for simulating incompressible SPH fluids, followed by our motivation to derive the dual-particle approach.

3.1 Basic Theory

In an incompressible model, the governing equations for a free-surface fluid can be written in a general form as follows

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f} \quad (1)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (2)$$

where \mathbf{v} is the velocity, p is the pressure, ρ is the density, μ is the kinematic viscosity, and \mathbf{f} is the external force. To enforce fluid incompressibility, the prediction-correction scheme first takes an integration to calculate the intermediate velocity \mathbf{v}^* by only taking account of the viscous and gravitational terms. Then, a second corrective step is taken to update the intermediate velocity with the pressure term [Koschier et al. 2022]

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \frac{\Delta t}{\rho} \nabla p, \quad (3)$$

where Δt is the time step size, and the new velocity \mathbf{v}^{n+1} should satisfy the divergence-free condition, i.e., $\nabla \cdot \mathbf{v}^{n+1} = 0$. By

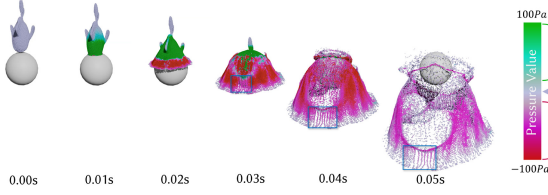


Fig. 2. Demonstration of the particle pressure distribution using the approximate projection method [Cummins and Rudman 1999] imposed with a zero-pressure Dirichlet boundary condition on the free surface boundary [Nair and Tomar 2014]. Particle pressures are visually represented through color coding, with purple and red particles situated in the negative fluid region, while blue and green particles occupy the positive pressure region. The particle count is 72k, the time step size is set to 0.001s, and the artificial viscosity model (XSPH) is implemented with a parameter value of 0.05.

inserting Equation (3), the divergence-free condition can be fulfilled by solving the following **pressure Poisson equation (PPE)**

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = \frac{\nabla \cdot \mathbf{v}^*}{\Delta t}. \quad (4)$$

Assuming a zero-pressure Dirichlet boundary condition is imposed on the free-surface boundary [Nair and Tomar 2014; Takahashi et al. 2018; Yang et al. 2016], solving Equation (4) yields the pressure field distribution across particles containing both positive and negative values, as illustrated in Figure 2.

Citing various engineering studies [Lyu et al. 2021; Monaghan 2000; Sun et al. 2019; Zhang et al. 2017], it is well-established that negative pressure induces the onset of tensile instability. More specifically, as the fluid is under a tensile stress state, attractive forces originating from negative-pressure regions can lead to anomalous particle clustering [Gotoh and Khayyer 2016; Zhang et al. 2017], as illustrated in a snapshot at $t = 0.05s$ in Figure 2. Although it is easy for us to weaken the tensile instability by simply clamping all negative particle pressures to zero [Bender and Koschier 2015; Ihmsen et al. 2014a], the error introduced near free surface boundaries may destroy thin fluid features. Note the DFSPH solver [Bender and Koschier 2015] using the negative pressure clamping scheme is unable to capture thin fluid sheets, as shown in Figures 23 and 24. Therefore, the main purpose of this work is to find a solution to alleviate the tensile instability in particle methods, yet can still capture all small-scale thin features.

3.2 Motivation

Before giving our solution, let us first investigate how tensile instability arises from a particle method. Intuitively, tensile instability occurs in presence of attractive inter-particle forces when the inter-particle interaction strength increases as the two particles approach [Khayyer and Gotoh 2011]. Assume a particle i is located in the middle of two neighboring particles j and k in the right figure. If particle i suffers from the tensile instability problem, its balance can be easily broken by slightly shifting its position to left or right because the particle that is closer can impose a larger force to drive particle i further apart. Figure 3 depicts the instability issue discussed previously in one-dimensional space. Please

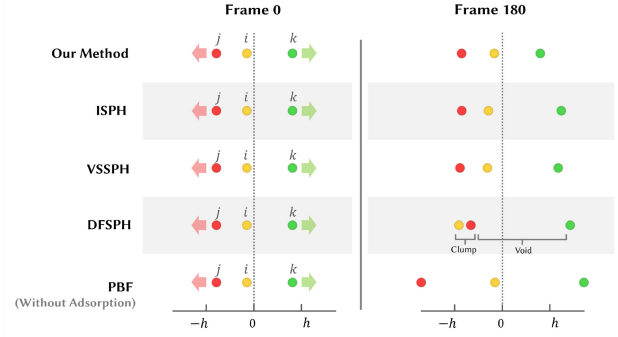
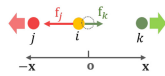


Fig. 3. The scenario involves the demonstration of three particles experiencing tensile stress. Particle j and k move apart at a small initial velocity ($0.05m/s$), while particle i , initially at rest, is positioned asymmetrically between particles j and k . Notably, only incompressibility solvers are utilized, while artificial viscosity or alternative solvers are not employed. Our approach employs the dual-particle method with a spatially adaptive strategy (S.C.). The PBF method [Macklin and Müller 2013] does not incorporate adsorption in our implementation. The ISPH method [Nair and Tomar 2014] maintains negative pressures. The DFSPH method [Bender and Koschier 2015] incorporates a negative-pressure clamping scheme. Additionally, the VSSPH [He et al. 2020] extends the “staggered-particle” SPH method introduced by He et al. [2012b].

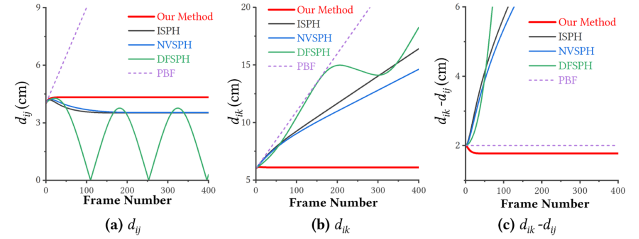


Fig. 4. Inter-particle distance of the three-particles scenario (Figure 3). (a) The distance between particle i and j (d_{ij}); (b) The distance between particle i and k (d_{ik}); (c) The difference between d_{ik} and d_{ij} .

observe the distributions of all three particles and the curves representing inter-particle distances plotted in Figure 4. In the PBF method [Macklin and Müller 2013], the attractive inter-particle force is disabled, therefore particle i remains stationary. The ISPH [Nair and Tomar 2014], VSSPH [He et al. 2020] and DFSPH [Bender and Koschier 2015] all preserve attractive inter-particle force. However, particle i fails to remain stable due to a larger attractive force from the left neighboring particle. This kind of instability can lead to the formation of voids and particle clustering on thin fluid sheets.

Within the pairwise-force model, Swegle et al. [1995] has theoretically demonstrated a sufficient condition to avoid tensile instability. For an incompressible fluid solver, the sufficient condition is formulated as

$$pW'' > 0, \quad (5)$$

where W'' is the second derivative of the SPH kernel function. Notice when a particle is under a tensile stress state, i.e., $p < 0$, its motion can remain stable only when the condition $W'' < 0$ is

satisfied. Please refer to Figure 5 for a demonstration of a cubic spline kernel function with the unstable regime $W'' \geq 0$ plotted. To simplify the following discussion, we only consider a one dimensional problem with an equally-spaced distribution. Without loss of generality, we assume all fluid particles have the same types of physical quantities and their movements are only affected by attractive pressure forces. In a standard SPH method, both the particle pressure and velocity are defined at the same locations as shown in Figure 5(a). As a result, if particle i is exactly located in the middle of its two neighboring particles, it will stay static due to a zero net force. However, if particle i deviates from its original position slightly to the left, the pressure force \mathbf{f}_j exerted by particle j becomes larger than the pressure force \mathbf{f}_k exerted by particle k . This indicates particle i 's movement can erroneously be accelerated due to the tensile instability, causing pairwise clumping between particle i and j . From Figure 5(a), it can also be noticed particle i is just located inside the unstable region, indicating particle i may suffer from tensile instability no matter how it moves between particle j and k .

To address the above problem, one simple way is to increase the smoothing length h to guarantee particle i is located inside the stable region. However, if the smoothing length is too large, the details and accuracies of the fluid may be smoothed out as well [Liu and Liu 2010]. More importantly, a larger smoothing length requires a much higher demand for both the computational and storage costs. Motivated by the stress points used in soil mechanics [Chalk et al. 2020; Randles and Libersky 2000], we introduce additional virtual particles (J and K) to store particle pressures, as demonstrated in Figure 5(b). The pressure forces exerted on particle i are now calculated with its two neighboring virtual particles (J and K). It can be noticed that particle i is now located inside the stable region of kernel functions defined on virtual particles. As a result, when particle i moves slightly from its original to the left, the pressure force \mathbf{f}_J exerted by the left virtual particle becomes smaller than the pressure force \mathbf{f}_K exerted by the left virtual particle, thus helps slow down the particle clumping trend. Besides, it can be noticed tensile instability can mostly be avoided when particle i moves between its two neighboring virtual particles by choosing an appropriate kernel function. Finally, the removal of the colocal nature of standard SPH helps completely remove the first-order sawtooth mode (typically known as the **zero-energy mode** [Dyka and Ingel 1995]) which could grow under the tensile instability. As shown in Figures 3 and 4, the method introducing additional virtual particles ensures the particles remain well-distributed during the stretching process, which is crucial for fluid simulation to capture streamlines and thin sheets. Based on the above motivation, we will give more details on how to setup our dual-particle approach in the following context.

3.3 Dual-Particle Discretization

Our dual-particle framework uses two different kinds of particles to carry field variables. The real particles carry particles' original mass, velocity, and so on. Virtual particles are regenerated at the beginning of each time step to carry particles' pressure and their locations will be updated in the next time step to stay close to the real ones. In the following discussion, we use lowercase letters i, j

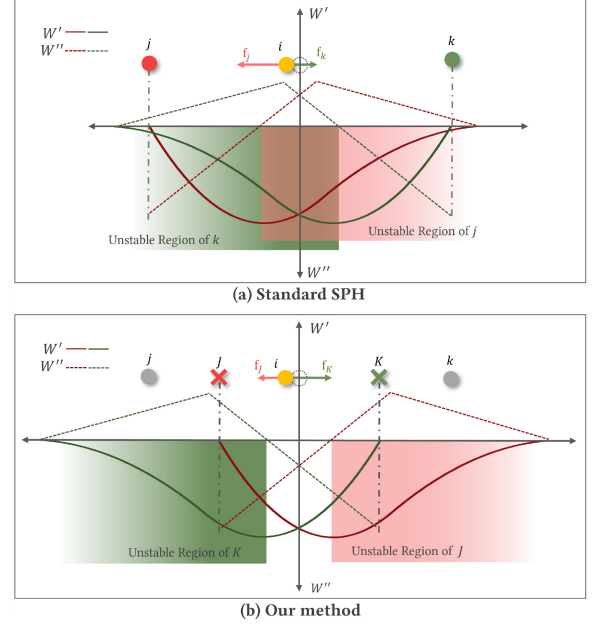


Fig. 5. Demonstration of tensile instability in a standard SPH and our dual-particle method. Assume the particle is under a tensile stress state, i.e., $p_i < 0$. The solid curves and the dashed curves respectively represent the first-order and second-order derivative of the cubic spline kernel W [Koschier et al. 2019]. The unstable regions are plotted for the cubic spline kernel. Notice (a) in the original SPH method, particle i is always in the unstable region because of its colocal nature. (b) Our motivation is to reduce the size of unstable regions for particles under a tensile stress state by introducing additional pressure calculation points, i.e., virtual particles J and K .

to denote field variables defined on real particles while uppercase letters I, J for field variables defined on virtual particles. By taking the summation approximation based on our dual-particle model, and introducing the discretization of the SPH method [Monaghan 1992], a physical quantity q can be computed with four different formulae as follows

$$\begin{aligned} q_i &= \sum_j V_j [q_j] W(\mathbf{x}_i - \mathbf{x}_j, h) \\ \hat{q}_i &= \sum_J V_J [q_J] W(\mathbf{x}_i - \mathbf{x}_J, H) \\ q_I &= \sum_j V_j [q_j] W(\mathbf{x}_I - \mathbf{x}_j, h) \\ \hat{q}_I &= \sum_J V_J [q_J] W(\mathbf{x}_I - \mathbf{x}_J, H) \end{aligned} \quad (6)$$

where the hat is used to denote quantities that are calculated by taking contributions from the neighboring virtual particles J , W is the SPH kernel function, h and H are the smoothing lengths of real and virtual particles, and V is the particle volume. As suggested by Koschier et al. [2019], the volume of real particle is directly calculated as

$$V_i = \frac{m_i}{\rho_i}, \quad \text{where } \rho_i = \sum_j V_j [\rho_j] W_{ij} = \sum_j m_j W_{ij}. \quad (7)$$

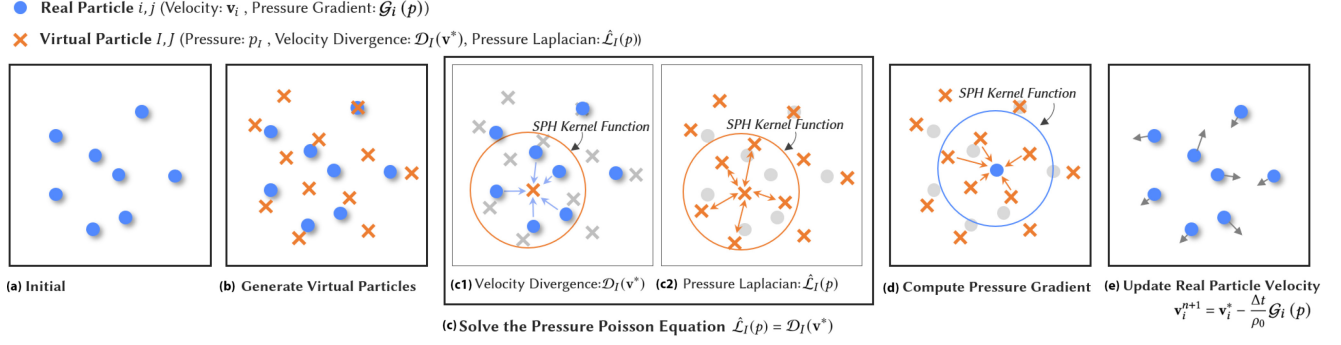


Fig. 6. An overview of our dual-particle method. In our dual-particle approach, the velocity and pressure gradients are defined on real particles, while the pressure, velocity divergence and pressure Laplacian are calculated on virtual particles. (a)-(b) Virtual particles are generated from real particles; (c1) Discretize the divergence operator on virtual particles with respect to real particle; (c2) Discretize the Laplacian operator on virtual particles with respect to virtual particles, and solve the PPE; (d) Calculate pressure gradients on real particles with respect to virtual particles; and (e) Update real particles' velocities.

Since virtual particles lack the mass property, we cannot use the above equation to compute its volume. To guarantee a smooth transition and ensure the stability of the simulation, Equation (6) is applied to calculate the volume for virtual particles as follows

$$V_I = \sum_j V_j [V_j] W(\mathbf{x}_{IJ}, h) = \sum_j V_j^2 W(\mathbf{x}_{IJ}, h), \quad (8)$$

where \mathbf{x}_{IJ} denotes $\mathbf{x}_I - \mathbf{x}_J$. It can be found that when a virtual particle moves far away from the boundary, its volume will be decreasing to zero.

3.4 Our Algorithm

By combining the dual-particle discretization into an approximate SPH projection method [Cummins and Rudman 1999], the full procedure to simulate incompressible fluids can be outlined as follows:

- (1) Generate virtual particles according to the description in Section 5.
- (2) Compute particles' volumes using Equations (7) and (8).
- (3) Discretize the PPE using Equation (11) and (14), then impose the free-surface boundary condition by updating the diagonal elements of the coefficient matrix (see details in Section 4.2).
- (4) Solve the linear system of equations using a conjugate gradient method.
- (5) Calculate the pressure force for real particles according to Equation (16).
- (6) Update velocities and positions for real particles and delete virtual particles.

Please also refer to Figure 6 for a more intuitive illustration and the pseudocode in Algorithm 1 for a detailed description.

4 DUAL-PARTICLE APPROXIMATE PROJECTION

In this section, we will discuss how to enforce fluid incompressibility using an approximate projection within our dual-particle model.

4.1 The Pressure Poisson Equation

To discretize the PPE, a variety of different strategies are actually available for both the Laplacian and divergence operators in SPH

community [Fürstenau et al. 2017]. Since our main purpose is to verify the effectiveness of using our dual-particle model to remove tensile instability, we choose to implement the most commonly used approximate projection method introduced by Cummins and Rudman [1999]. With an incompressibility assumption, i.e., $\rho = \rho_0$, the pressure Laplacian (i.e., the left hand of the Equation (4)) can be defined on virtual particles as follows

$$\hat{L}_I(p) = \frac{2}{\rho_0} \sum_j V_j \frac{p_I - p_J}{r_{IJ} + \eta} W'_{IJ}, \quad (9)$$

where V_j is the virtual particle volume with J denoting all neighboring virtual particles, $r_{IJ} = \|\mathbf{x}_I - \mathbf{x}_J\|$, η is a small constant to prevent being divided by zero, and $W'_{IJ} = \frac{\partial W(\mathbf{x}_{IJ}, H)}{\partial r_{IJ}}$. In a similar way, the source term in Equation (4) can be defined on virtual particles as follows

$$\mathcal{D}_I(\mathbf{v}^*) = -\frac{1}{\Delta t} \sum_j V_j \mathbf{v}_j^* \cdot \nabla_I W_{IJ}, \quad (10)$$

where V_j is the real particle volume with j denoting all neighboring real particles and $\nabla_I W_{IJ} = \frac{\mathbf{x}_{IJ}}{r_{IJ}} W'_{IJ}$. With the above two discretizations, a linear system of equations can then be formulated to solve for the unknown pressure field.

However, the stable solve of the linear system of equations for arbitrary particle configurations requires us to first address the two following numerical issues.

Particle deficiency. The form of Equation (10) may suffer large errors near the free surface boundary due to the boundary deficiency. To address this problem, Equation (10) is modified as follows [Koschier et al. 2019]:

$$\mathcal{D}_I(\mathbf{v}^*) = -\frac{1}{\Delta t} \sum_j V_j (\mathbf{v}_j^* - \bar{\mathbf{v}}_I^*) \cdot \nabla_I W_{IJ}, \quad (11)$$

where $\bar{\mathbf{v}}_I^*$ represents the intermediate virtual particle velocity mainly used to reduce boundary errors. Since the virtual particle I does not store the velocity quantity in our dual-particle approach, we introduce the corrective smoothed particle method [Chen and Beraun 2000] to estimate the fluid velocity at the virtual particle

position:

$$\bar{\mathbf{v}}_I^* = \frac{\sum_j V_j \mathbf{v}_j^* W_{IJ}}{\sum_j V_j W_{IJ}}. \quad (12)$$

Note the intermediate virtual particle velocity $\bar{\mathbf{v}}_I^*$ can be viewed as an weighted interpolation calculated from neighboring real particles.

Density drifting. The particle density drifting problem should be addressed to avoid permanent volume loss. A typical solution is to add a compensation term to the source term of the pressure Poisson equation [Khayyer and Gotoh 2011]. Since the pressure Poisson equation is solved on virtual particles within our dual-particle method, we should first calculate the virtual particle density $\rho_I = \sum_j m_j W_{IJ}$, and then add the following term to Equation (10)

$$\Lambda_I = \kappa \frac{\max(\rho_I - \rho_0, 0)}{\rho_0 \Delta t}, \quad (13)$$

where κ represents a constant to control the value. In practice, κ is typically chosen to be of the same order of magnitude as m/ρ_0 to ensure stability. Combining both Equations (11) and (13), the final source term is formulated as $\mathcal{D}_I(\mathbf{v}^*)^{new} = \mathcal{D}_I(\mathbf{v}^*) + \Lambda_I$.

4.2 Boundary Handling

For a projection-based incompressible fluid solver, both free-surface and solid boundary conditions should be appropriately incorporated. In the following discussion, we will show how to handle both boundaries within our dual-particle framework.

Free-surface boundary condition. For particles near the free-surface boundary, only particles inside the boundary contribute to the summation integration in Equation (6) since there are no particles outside the boundary. Within a projection method, a zero pressure boundary condition should be imposed to guarantee the discretized pressure Poisson (Equation (4)) obtain a positive-definite coefficient matrix [Bridson 2015]. A common practice is to explicitly identify boundary particles near the free-surface boundary, and assign zero pressures on those particles [Takahashi et al. 2018]. Unfortunately, identifying boundary particles explicitly will make the pressure projection sensitive to the particle distribution because the zero-pressure boundary condition is imposed on particles rather than on a smooth transition region. Therefore, a semi-analytical method was proposed to impose free-surface boundary conditions implicitly [Nair and Tomar 2014; Yang et al. 2016], which shows significant improvement on the convergence speed and stability in solving the PPE.

Motivated by their work, we will extend the semi-analytical method and make it better suited to our dual-particle framework. Unlike [Nair and Tomar 2014], virtual particles can be located outside of the free surface boundary. However, as the virtual particle travels far away from the free surface boundary, its volume will be small as well according to Equation (8). Therefore, the Laplacian operator of the PPE (Equation (4)) can be uniformly discretized as

$$\begin{aligned} \hat{\mathcal{L}}_I(p) &= \frac{2}{\rho_0} \sum_j V_{JP_I} \frac{W'_{IJ}}{r_{IJ} + \eta} - \frac{2}{\rho_0} \sum_j V_{JP_J} \frac{W'_{IJ}}{r_{IJ} + \eta} \\ &= \hat{\alpha}_I p_I - \sum_j \hat{\alpha}_{IJ} p_J, \end{aligned} \quad (14)$$

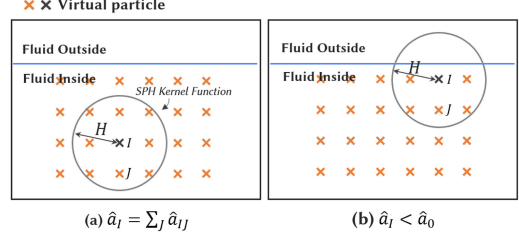


Fig. 7. Illustration of the free-surface boundary condition. A semi-analytical scheme is applied to impose the free-surface boundary condition by setting $\hat{\alpha}_I = \max(\hat{\alpha}_I, \alpha_0)$, where α_0 is precalculated at the beginning of simulation for a prototype virtual particle filled with full neighboring real and virtual particles. Note (a) when a virtual particle is located inside a fluid and far away from the boundary, the summation of α_{IJ} could be larger than $\hat{\alpha}_0$ due to compression and we will not clamp its value; (b) Otherwise, if a virtual particle I is near the free-surface, the value of $\sum_j \alpha_{IJ}$ is typically smaller than α_0 , and we will clamp its value to α_0 .

for all virtual particles, where $\hat{\alpha}_I$ and $\hat{\alpha}_{IJ}$ are defined as

$$\hat{\alpha}_I = \sum_j \hat{\alpha}_{IJ}, \quad \hat{\alpha}_{IJ} = \frac{2}{\rho_0} \frac{V_J W'_{IJ}}{r_{IJ} + \eta}. \quad (15)$$

Note α_I represents the diagonal elements of the coefficient matrix of the discretized PPE, and α_{IJ} represents the off-diagonal elements. According to Equation (8), the virtual particle volume V_J is smoothly decreasing across the free surface boundary. Therefore, we identify a virtual particle as a boundary particle only if its coefficient α_I is smaller than a predefined threshold α_0 , as shown in Figure 7. In our implementation, α_0 is precalculated at the beginning of simulation for a prototype virtual particle filled with full neighboring real and virtual particles. The coefficient α_I is clumped to α_0 for all boundary particles when we solve the linear system of equations, i.e., $\hat{\alpha}_I = \max(\hat{\alpha}_I, \alpha_0)$. This operation can ensure the coefficient matrix is weakly diagonally dominant, the discretized PPE, therefore, can be efficiently solved with a standard iterative solver. Compared with a zero pressure boundary condition imposed on particles, the implicit strategy to only update the diagonal elements of the coefficient matrix for boundary particles makes it possible to impose a smooth zero pressure condition on the free-surface boundary. As a result, the motion of real particles is guaranteed to be smooth, and will not suffer from the stair-step artifacts introduced by virtual particles.

Solid boundary condition. We impose solid boundary conditions with standard ghost particles [Akinci et al. 2012; Bridson 2015; Takahashi et al. 2018]. As shown in Figure 8, ghost solid particles are evenly sampled in the solid region and assigned with the same mass as real fluid particles. The ghost solid particles carry velocities of solid materials, and they are also considered as real particles to calculate the velocity divergence of the virtual particle (Equation (11)).

4.3 Velocity Update

After the PPE is solved, the pressure field defined on virtual particles is used to update the velocity of real particles. Following a standard particle approximation, the pressure gradient defined on

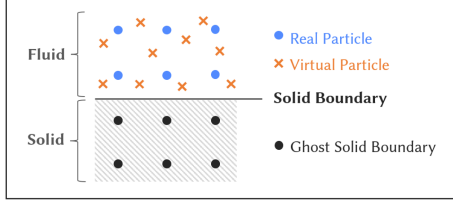


Fig. 8. Illustration of the solid boundary condition.

real particles can be calculated as

$$\mathcal{G}_i(p) = \sum_j V_J p_J \nabla_i W_{iJ}. \quad (16)$$

To simulate the interaction with solid boundaries, Equation (16) needs to be corrected by the solid particles as well, i.e.,

$$\mathcal{G}_i(p)^{new} = \mathcal{G}_i(p)^{old} + \sum_{j^s} V_j^s \text{Proj}_{\mathbf{n}_{j^s}}(\mathbf{v}_i - \mathbf{v}_{j^s}) W_{ij^s}, \quad (17)$$

where the \mathbf{n}_{j^s} is the normal vector of the solid boundary at the position of the solid particle j^s , $\text{Proj}_{\mathbf{n}_{j^s}}(\mathbf{v}_i - \mathbf{v}_{j^s})$ represents the projection of $\mathbf{v}_i - \mathbf{v}_{j^s}$ on vector \mathbf{n}_{j^s} [He et al. 2020]. As a result, the real particle velocities can be updated as follows

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^* - \frac{\Delta t}{\rho_0} \mathcal{G}_i^{new}(p), \quad (18)$$

In general, particle methods prefer a regular distribution of particles to obtain simulations with high stability and accuracy [Fries and Belytschko 2008]. The same situation comes up with the distribution of virtual particles. To evaluate how the distribution of virtual particles affect the stability of fluid simulation, we introduce three different strategies to generate virtual particles, as demonstrated in Figure 9. The following contains our principle for each strategy as well as the implementation details.

5 VIRTUAL PARTICLE GENERATION

S.A. Colocational strategy. Virtual and real particles share the same locations.

– **Principle and Implementation:** To be comparable to a standard SPH projection method where both velocity and pressure are defined at the same locations [Cummins and Rudman 1999; Takahashi et al. 2018; Yang et al. 2016], this strategy is to simply assign real particles' position to the virtual particles and set $H = h$. By referring to Algorithm 1, it can be noticed that the dual-particle approximate projection with this strategy degenerates to the original approximate projection method (i.e., ISPH method) proposed by Cummins and Rudman [1999].

S.B. Particle shifting strategy. Virtual particles are generated by shifting the positions of real particles, and the distribution of virtual particles is slightly more uniform than that of real particles.

– **Principle and Implementation:** Before giving our final strategy to generate an ideal uniform distribution of virtual particles, we try with a strategy, where virtual particles are less well-distributed, to demonstrate how the distribution of virtual particles affect the simulation results. We first generate a replica of real particles and assign their positions to

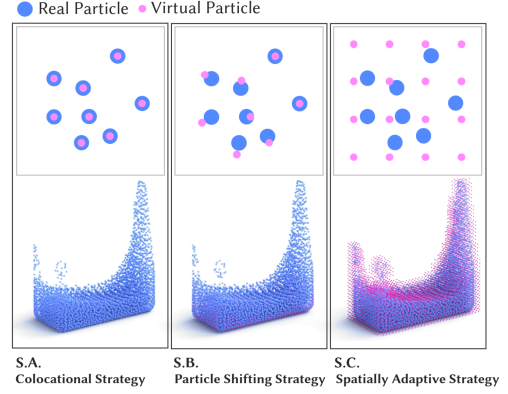


Fig. 9. Illustration of the three different virtual particle generation strategies.

virtual particles as S.A. does, and then use the **position-based fluid (PBF)** method [Macklin and Müller 2013] to slightly shift virtual particles' positions without updating the particle velocities. During this process, all other terms including particle inertia, viscosity and external forces are neglected, ensuring that the distribution of virtual particles is more evenly distributed and matches real particles well.

S.C. Spatially adaptive strategy. Virtual particles are generated to have a spatially adaptive uniform distribution. Unlike the first two strategies, the virtual number in this strategy is dynamic.

– **Principle and Implementation:** In this strategy, our purpose is to generate an ideal uniform distribution of virtual particles. Besides, to reduce memory consumption, the coverage of virtual particles should be compact. To obtain a well-distributed yet compact distribution of virtual particles, our solution is to generate virtual particles at equal-spaced and orthometric fixed points around real particles.

In fact, there exist other strategies to generate virtual particles in the dual particle framework. Nevertheless, we believe the above mentioned three strategies are enough to demonstrate the effectiveness of using virtual particles in suppressing the tensile instability. Since it is straightforward to implement the first two strategies, we only give more details in the next section to show how to implement the spatially adaptive strategy (S.C) on GPU to get a high performance simulation.

5.1 Spatially Adaptive Virtual Particle Generation

While other sparse data structures, such as OpenVDB [Gao et al. 2018; Museth 2021; Museth et al. 2013; Wang et al. 2020], should work with our dual-particle method, we prefer a GPU-friendly way to generate spatially adaptive virtual particles to make the whole simulation compatible with modern GPUs.

As shown in Figure 11, the full procedure is outlined as follows

- (1) Update the positions of real particles and delete old virtual particles generated at the previous timestep.
- (2) For each real particle, generate candidate points within the support domain of the real particle. The candidate point is always initialized at cell vertices of a uniform Eulerian grid.

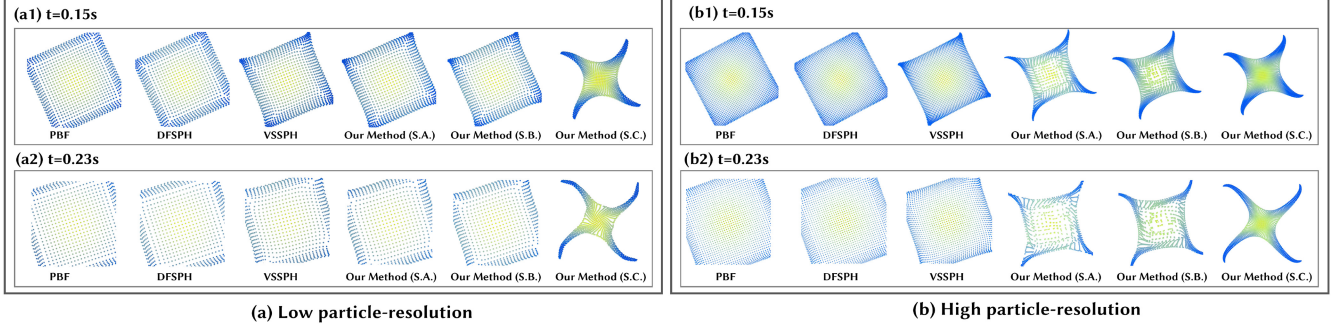


Fig. 10. Rotating square fluid patch. The smoothing length of all particles are $0.0125m$, initial angular velocities are $\frac{2\pi}{3} \text{ rad} \cdot \text{s}^{-1}$, and an artificial viscosity of 0.03 [Schechter and Bridson 2012] is applied. The particle spacing for (a) and (b) is $0.0075m$ and $0.005m$ respectively, and the (real) particle number is 1.02k and 2.3k respectively. In the scenarios, three other methods are implemented for comparison, including the PBF [Macklin and Müller 2013], the divergence free SPH (DFSPH) [Bender and Koschier 2015] and the variational staggered SPH [He et al. 2020] (which can be viewed as an extension to the staggered particle method [He et al. 2012b]).

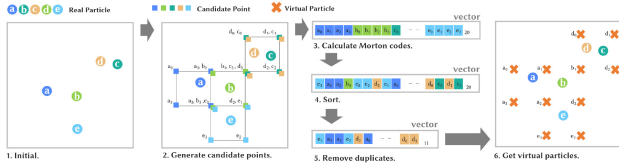


Fig. 11. An overview of our parallel implementation of the spatially adaptive strategy (S.C.) to generate virtual particles.

- (3) Calculate the Morton codes for all candidate points.
- (4) Sort in parallel the candidate points according to their Morton codes.
- (5) Remove duplicative candidate points by comparing their Morton codes.
- (6) Allocate a new buffer to store all unique virtual particles.

Notice implementation of all steps in the above procedure can be fully parallelized on the GPU.

6 EVALUATIONS AND DISCUSSIONS

In this section, we first provide a stability analysis using our dual-particle approach in simulating incompressible fluids. Then, we compare our method to both the hybrid particle-grid method and the stress-particle SPH method to demonstrate the similarities and differences between these methods.

6.1 Stability Analysis

To assess the efficacy of our dual-particle approach in mitigating the tensile instability issue, we initially simulated the rotational deformation of a square fluid patch, depicted in Figure 10, which has been widely recognized in engineering as a classical measure for evaluating the tensile stability [Colagrossi 2005; Khayyer and Gotoh 2011; Liu et al. 2018; Sun et al. 2017; Tsuruta et al. 2013]. In this zero-gravity scenario, the square fluid patch is initially subjected to a rigid-rotation velocity field defined as

$$\begin{cases} u_0 = \omega(y - y_c) \\ v_0 = -\omega(x - x_c) \end{cases}, \quad (19)$$

where (x_c, y_c) is the center of the fluid patch, ω is the angular velocity, and the divergence free condition for the velocity field is fulfilled, i.e., $\nabla \cdot \mathbf{v}_0 = 0$. In the evaluation process, the square fluid patch experiences a centrifugal force resulting from a significant negative pressure field. Consequently, it undergoes a gradual transformation into a compact shape with four arms [Colagrossi 2005; Sun et al. 2017]. We modeled this experiment using the same smoothing length and two different particle resolutions. Both comparisons show that most previous SPH methods in computer graphics fail to suppress the tensile instability due to insufficient attractive inter-particle forces under the same conditions. Moreover, doubling the particle resolution does not effectively improve the stability. This demonstrates that particle resolution is not the key factor in improving tensile stability. Within our dual-particle framework, the same kernel functions and free-surface boundary conditions are used for all three different virtual particle generation strategies. Our methods using S.A. and S.B. show improved performance in removing tensile instability compared to other methods, yet particle clustering still can be observed. By applying the S.C., our method is able to regenerate the correct behavior of the square fluid patch under rotation. Besides, the particle distribution also remains regular during the evolution of the rotating fluid patch.

From the above comparison, progressive improvement of the real particle distribution can be found as the virtual particle generation strategy changed from S.A., S.B. to S.C. Similar enhancements are noticeable in the experiments depicted in Figures 16 and 24. It appears that the sampling strategy of virtual particles is the key of our dual particle approach to address tensile stability, indicating removing the colocal nature and placing sufficient stress points at uniform positions can substantially improve the tensile stability. We will provide more evaluations in the following context.

It is important to note that the VSSPH method in Figure 10 also introduces the “stress points” to discretize the PPE. However, unlike our method, the VSSPH method does not explicitly generate “stress points”, which means it cannot guarantee the uniform distribution of virtual particles and fails to narrow down the unstable regions for particles under a tensile state. Consequently, it fails to weaken the tensile instability.

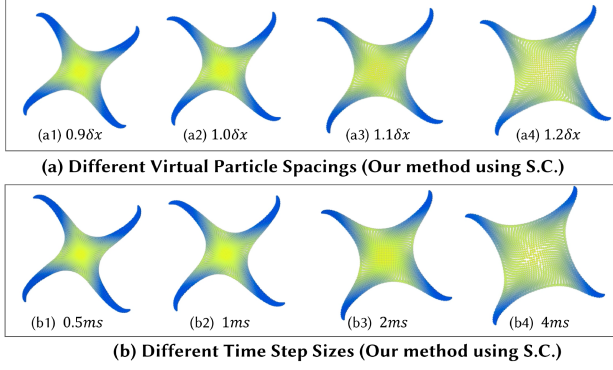


Fig. 12. The simulation involves rotating square fluid patches with varying virtual particle spacings and time steps. (a) Simulations using our S.C. method with different virtual particle spacings, namely, $0.9\delta x$, $1.0\delta x$, $1.1\delta x$, and $1.2\delta x$, where the real particle spacing δx remains fixed at $0.005m$, and the time step is set to $1ms$. (b) Simulations using our S.C. method with different time steps, specifically $0.5ms$, $1.0ms$, $2.0ms$, and $4.0ms$, while both virtual and real particle spacings are maintained at $0.005m$.

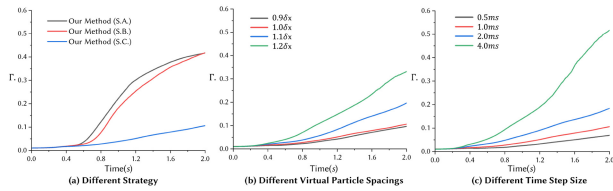


Fig. 13. Illustration of averaged particle distribution measure Γ for various simulations in Figures 10 and 12. (a) simulations conducted with our method using different virtual particle generation strategies; (b) simulations with varying virtual particle spacings; and (c) simulations with different time step sizes.

To elucidate the influence of virtual particle spacing and time-step size on tensile instability, we conducted additional experiments, as depicted in Figure 12. All experiments show no significant artifacts of particle aggregation. However, employing larger time step sizes or virtual particle spacings could result in an increased separation speed between real particles, causing challenges in maintaining accurate fluid shapes [Colagrossi 2005; Oger et al. 2007]. To quantify the particle distribution in Figure 10, we propose the following particle distribution measure for evaluating the performance of various configurations

$$\Gamma_i = \beta_0 \left(\frac{\rho_0 - \rho_i}{\rho_0} \right)^2 + \beta_1 \left| \frac{\sum_j V_j \nabla_i W_{ij}}{\sum_j V_j W_{ij}} \right|^2, \quad (20)$$

where the constant coefficients β_0 and β_1 are set to 1.0 in our current implementation. The first term assesses particle density, while the second term evaluates the symmetry of the particle distribution [He et al. 2014]. This combination helps quantify the quality of the particle distribution. Figure 13 depicts the curve of measure Γ_i applied to Figures 10 and 12, illustrating that distinctions between time step sizes of $0.5ms$ and $1.0ms$, as well as virtual particle spacings between $0.9\delta x$ and $1.0\delta x$, are negligible. Larger time step

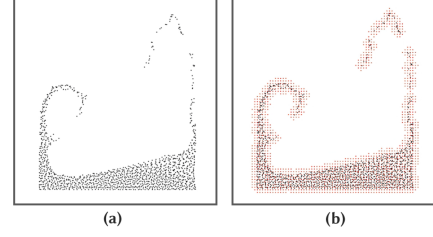


Fig. 14. 2D Dambreak. Our method with S.C. simulates the dam break example in the 2D space. Real and virtual particles are rendered as black and red points, respectively.

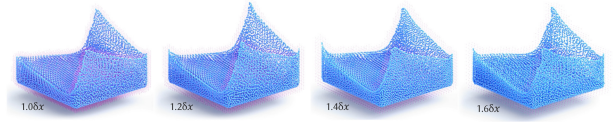


Fig. 15. Dam break. We simulate this example with four different virtual particle spacing distances. From Left to right, the spacing distances of virtual particles are set $1.0\delta x$, $1.2\delta x$, $1.4\delta x$ and $1.6\delta x$, respectively, where the spacing distances δx is always set to $0.005m$.

sizes or virtual particle spacings may degrade the quality of the simulation.

2D Dambreak. To evaluate the stability of the free-surface boundary condition, Figure 14 shows a 2D dam-break simulated with our dual-particle method with S.C. Real and virtual particles are rendered as black and red points, respectively. Notice when the real particles move to cross the boundary of virtual particles, no obvious artifacts or instabilities are observed, indicating the semi-analytical free-surface boundary is stable within our method.

Different resolutions of virtual particles (Figure 15). Figure 15 further demonstrates the stability of using different resolutions to generate spatially adaptive virtual particles. Notice similar results can be produced for all four cases. In a practical implementation, we suggest using a spacing distance that is 1 to 2 times of the spacing distance of real particles because too big a spacing distance for virtual particles can smooth out fluid details. In the following discussion, if not specified, the spacing distances of real particles and virtual particles are both set to the same value, i.e., $\delta x = 0.005m$.

Inviscid vs. viscous fluids. Simulating fluids with zero viscosity has been regarded as a long-standing challenge for particle methods. An artificial viscosity model (e.g., XSPH) is typically used to smooth the velocity field in order to avoid unphysical oscillations in the numerical results, thus making particles move more coherently. To demonstrate whether our method is able to model inviscid fluids, another dam break case is simulated, as shown in Figure 16. Comparison in the top row shows all previous methods as well as our dual-particle method using the first two virtual particle generation strategies cannot preserve coherent particle motions well in simulating an inviscid fluid. In contrast, our method with S.C. shows best in preserving coherent particle motions. After introducing artificial viscosity, all simulations demonstrate significant improvements. However, a comparison in the bottom row shows our method with S.C. still produces the best simulation

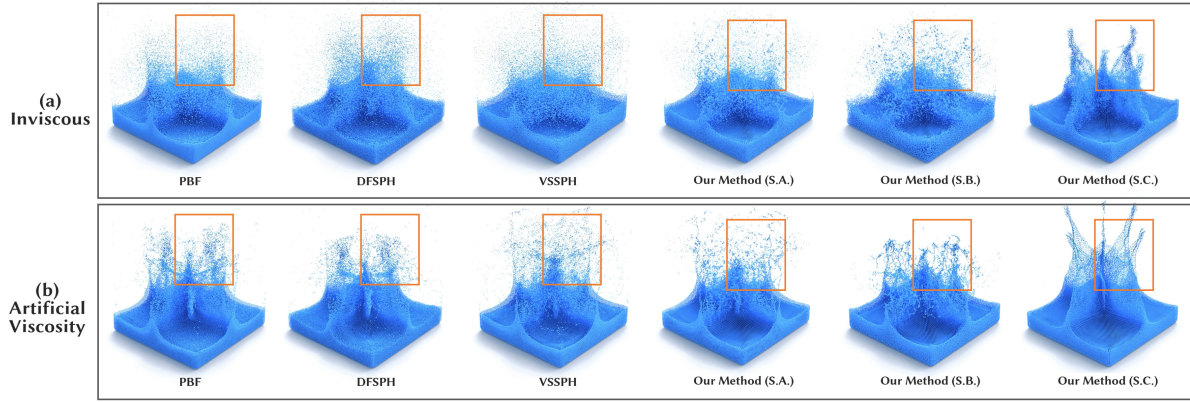


Fig. 16. Inviscid and viscous fluids. Each fluid block contains 237.3k particles. Splashes of the fluids usually occur in negative-pressure regions, which can cause tensile instability. If computational efficiency is not considered, the top row shows our dual-particle method using S.C is the best approach for simulating inviscid fluids since it preserves the coherence of particle motions. After introducing artificial viscosity (XSPH model with a parameter of 0.02), all simulations demonstrate significant improvements. However, the comparison in the bottom row shows our method with S.C still produces the more stable thin-fluid sheets, indicating the tensile instability introduced by the fluid incompressibility solver cannot be addressed by simply introducing artificial viscosity into particle fluids.

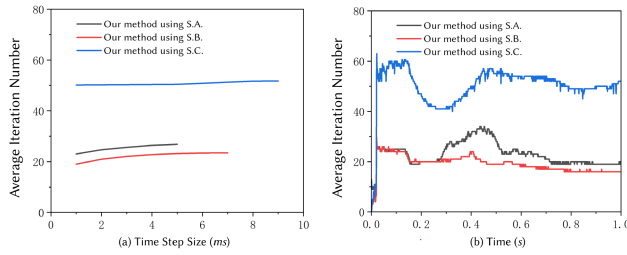


Fig. 17. Performance comparison. (a) Average iteration numbers of our dual-particle approach using different time step sizes for the breaking dam scenario in Figure 15; (b) Curve of the iteration number for a simulation with a time step size of 0.001. The convergence criterion is set to 10^{-4} of the maximum error for all simulations.

results, indicating the tensile instability introduced by the fluid incompressibility solver cannot be addressed by simply introducing artificial viscosity into particle fluids.

It is important to note that our method using S.C. can capture more stable small-scale thin features, but this improvement comes at a cost as the strategy may generate more virtual particles in modeling fluids. Our method using S.C. requires more computing time and consumes more memory in neighborhood lookup and pressure calculation processes. Table 1 shows the virtual particle counts and time costs for the three strategies in figure 16.

6.2 Performance Comparisons

We first present comparisons of our method with the three different strategies to generate virtual particles. Figure 17(a) shows the average computational cost per time step in solving the linear system of equations remains nearly constant for all three sampling strategies for a breaking dam with different time step sizes. In addition, the comparison between the three strategies shows that, despite slower convergence, S.C. can use a larger time step size

for simulation than the other two strategies. Figure 17(b) shows the iteration number curve for a simulation with a timestep size of 1ms, the comparison demonstrates that the S.B. has a faster convergence speed.

To illustrate the efficacy of our method, we conducted simulations of a dam-break scenario with varying time step sizes. The results, including the average iteration count for each computational method utilized, are depicted in Figure 18. In this experiment, the allowable average density and divergence errors for each method are set to be less than 10^{-3} . In contrast to alternative approaches, our method, particularly with S.C., yields better results with a large time step of $\Delta t = 5ms$.

6.3 Comparison to the Hybrid Particle-Grid Method

Comparing our dual-particle approach to the HPG method [Fei et al. 2021; Um et al. 2014], we can find similarities, e.g., the virtual particles in our method resemble the Eulerian grid in the HPG method. However, they are essentially two different approaches. We will demonstrate their differences in this subsection.

In a HPG method, the pressure projection is identical to the standard Eulerian grid method, and particles are only used to track materials and carry quantities such as velocities. Depending on the interpolation method between the particle and background grid, all HPG methods can be mainly classified into the following categories:

- **Particle-In-Cell (PIC):** The velocities or momentums directly transfer between particles and the background grid, and the velocities or momentums on particles are not retained after the **particle-to-grid (P2G)** interpolation [Harlow 1962]. Due to the excessive numerical dissipation of the original PIC method, other variants with higher-order accuracies, such as **Affine-PIC (APIC)** and **Polynomial-PIC (PolyPIC)**, are developed. We take the standard PIC method as an example, and the equations for the P2G and

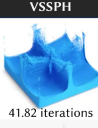
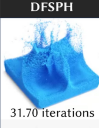
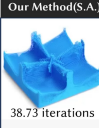
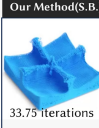



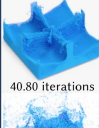


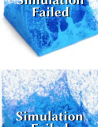



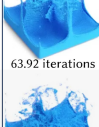
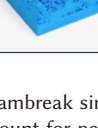
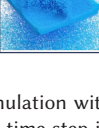
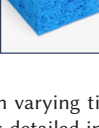
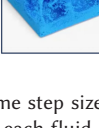
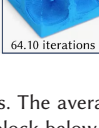
	VSPH	DFSPH	Our Method(S.A.)	Our Method(S.B.)	Our Method(S.C.)
$\Delta t=2\text{ms}$	 41.82 iterations	 31.70 iterations	 38.73 iterations	 33.75 iterations	 62.92 iterations
$\Delta t=3\text{ms}$	 44.99 iterations	 56.36 iterations	 40.80 iterations	 36.47 iterations	 63.55 iterations
$\Delta t=4\text{ms}$	 Simulation Failed	 Simulation Failed	 Simulation Failed	 37.67 iterations	 63.92 iterations
$\Delta t=5\text{ms}$	 Simulation Failed	 Simulation Failed	 Simulation Failed	 Simulation Failed	 64.10 iterations

Fig. 18. Dam-break simulation with varying time step sizes. The average iteration count for per time step is detailed in each fluid block below. In this test, each fluid block containing 237.3k particles, the artificial viscosity is set to 0.1, the particle spacing δx is set to $0.005m$, and the smoothing length is set to $2.5\delta x$.

grid-to-particle (G2P) interpolations are formulated as

$$\begin{aligned} \text{PIC P2G: } m_g \mathbf{v}_g^n &= \sum_p w_{gp} m_p \mathbf{v}_p^n \\ \text{PIC G2P: } \mathbf{v}_p^{n+1} &= \sum_g w_{gp} \mathbf{v}_g^{n+1} \\ \mathbf{x}_p^{n+1} &= \mathbf{x}_p^n + \Delta t \sum_g w_{gp} \mathbf{v}_g^{n+1} \end{aligned}$$

where m is the mass, w is the interpolation function, \mathbf{x} is the particle position, \mathbf{v} is the velocity, n denotes the timestep, subscripts g and p represents physical quantities defined on the grid and particle, respectively.

- **Fluid-Implicit-Particles (FLIP):** The basic idea of the FLIP method is to try to blend the velocity defined on the background grid and particles. The high-frequency velocity fields defined on particles are partially retained during the G2P step to mitigate excessive numerical dissipation in the PIC method [Brackbill and Ruppel 1986]. The P2G and G2P processes in FLIP are written as

$$\begin{aligned} \text{FLIP P2G: } m_g \mathbf{v}_g^n &= \sum_p w_{gp} m_p \mathbf{v}_p^n \\ \text{FLIP G2P: } \mathbf{v}_p^{n+1} &= \sum_g w_{gp} \mathbf{v}_g^{n+1} + \alpha (\mathbf{v}_p^n - \sum_g w_{gp} \mathbf{v}_g^n) \\ \mathbf{x}_p^{n+1} &= \mathbf{x}_p^n + \Delta t \sum_g w_{gp} \mathbf{v}_g^{n+1} \end{aligned}$$

where α is the blending ratio between FLIP and PIC. Note a fraction of the high-frequency velocity modes defined on particles is retained. Therefore, a value of α that is close to 1 can effectively reduce the numerical dissipation in fluid simulation.

- **Naturally-modified FLIP (NFLIP):** Following Fei et al. [2021], the G2P and P2G processes of NFLIP

[Stomakhin et al. 2013] are written as

$$\begin{aligned} \text{NFLIP P2G: } m_g \mathbf{v}_g^n &= \sum_p w_{gp} m_p \mathbf{v}_p^n \\ \text{NFLIP G2P: } \mathbf{v}_p^{n+1} &= \sum_g w_{gp} \mathbf{v}_g^{n+1} + \alpha (\mathbf{v}_p^n - \sum_g w_{gp} \mathbf{v}_g^n) \quad (21) \\ \mathbf{x}_p^{n+1} &= \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1} \end{aligned}$$

Compared with a standard FLIP method, the major difference here is NFLIP uses \mathbf{v}_p^{n+1} rather than $\sum_g w_{gp} \mathbf{v}_g^{n+1}$ to update particle positions. In other words, the high-frequency velocity modes are added to particle positions as well. Therefore, particles in the same sub-grid can move away from each other at a faster speed.

Based on the above discussions, we can find several features that distinguish our dual-particle approach from the HPG methods. **First**, our method does not take the P2G transfer, and the divergence of velocity defined on virtual particles is taken with respect to the real particles directly (see Equation (11)). **Second**, our method updates particle velocities by directly taking pressure gradients with respect to virtual particles. There is no G2P process to transfer velocities from virtual particles to real ones as well. **Third**, virtual particles carry all physical quantities (such as density, volume, etc) similar to a standard particle in SPH, facilitating the solving of the PPE (e.g., we can easily incorporate the semi-analytical free-surface condition).

To provide more validation among those methods, we simulate a dam-break scenario as shown in Figure 19. Apart from the discretization, all conditions are identical in this comparison. In addition, for all implementations, we added the density error compensation term in Equation (13) to the source term of the PPE to make the comparison fair. From the comparison, it can be noticed that previous HPG methods fail to generate a regular distribution of particles at the moment of the record. Besides, the comparison shows that the Full-NFLIP is much more noisy than other methods, which also matches the conclusion given in previous studies [Ando et al. 2012; Fei et al. 2021; Jiang et al. 2015]. Within our method, we adopt the SPH form of the pressure projection, distinguishing it from HPG methods that rely on Eulerian grid discretization schemes. The SPH form discretization avoids the dissipation problem introduced by P2G/G2P operations [Fei et al. 2021; Jiang et al. 2015] and allows our virtual particles to carry a wider range of quantities than the grids in HPG methods. Accordingly, our method achieves a more stable result compared to the standard particle-grid methods, as depicted in Figure 19.

To assess numerical dissipation, we perform multiple simulations of a rotating square fluid patch and depict the total kinematic energy curves in Figure 20. It is evident that HPG methods generally exhibit higher energy losses compared to our method using S.C. Additionally, both FLIP0.99 and the Full-NFLIP method exhibit markedly unstable behaviors in this test.

6.4 Comparison to the Stress-Point SPH Method

Our dual-particle approach shares a motivation similar to the stress point SPH method [Belytschko and Xiao 2002].

The stress-particle method was initially introduced Dyka [Dyka et al. 1997], which incorporates a set of points to store stress and addresses the tensile instability inherent in the one-dimensional

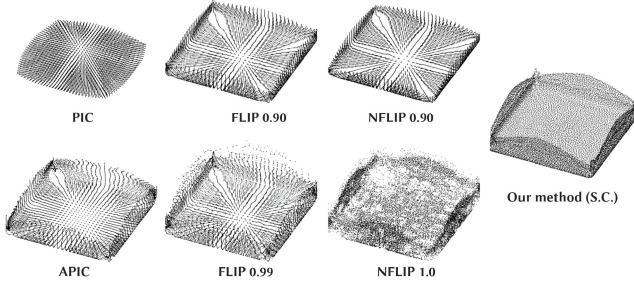


Fig. 19. Comparison to hybrid particle-grid methods. In this scenario, we perform simulations with the PIC, APIC, FLIP0.90, FLIP0.99, NFLIP0.90, Full-NFLIP (α is set to 1.0) and our method with S.C. In order to keep fairness for comparison, the grid spacing in all hybrid particle-grid methods is set to be equal to the spacing distances of virtual particles in our method, and the threshold in solving the pressure Poisson equation is set to 10^{-4} , a density error compensation scheme similar to Equation (13) is also added to all hybrid particle-grid methods. Besides, all fluids are modeled as inviscid.

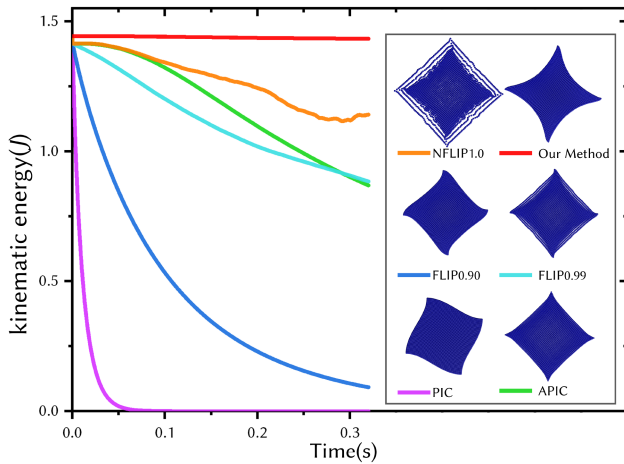


Fig. 20. The kinematic energy is plotted over time for the rotating square fluid patch test. All simulations maintain a real particle spacing of $0.005m$, with grid spacing and virtual particle spacing set to $0.006m$, and a time step size of $0.05ms$. It is observed that the rotating square fluid patches modeled by hybrid particle methods incur higher energy losses compared to our S.C. method.

SPH method. Subsequently, Randles and Libersky [2000] extended the application of the stress-particle method to tackle the zero-energy mode problem, boundary conditions, and tensile instability issues associated with solid materials in two-dimensional space. However, since the stress point SPH method needs to maintain a staggered arrangement of stress points and velocity particles, it can only be used to model materials with a relatively small deformation. To extend the method to handle large deformation geomechanics problems, such as soil samples and slopes, Chalk et al. [2020] recently documented various node-stress-point configurations, highlighting instances of success and failure among them. In fact, their work fails to answer how to work out the best fit

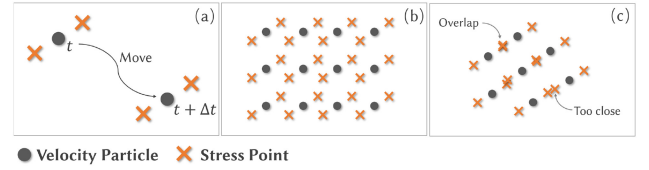


Fig. 21. Illustration of the Stress-Point SPH method [Chalk et al. 2020]. (a) Stress points move with their associated velocity particles; (b) The distribution of velocity particles and stress points in an ideal case; and (c) When the material is deformed, some of the stress points may be too close to each other or even overlap.

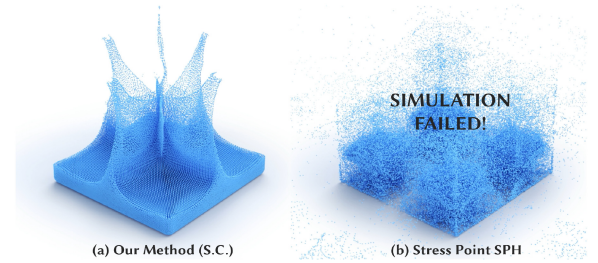


Fig. 22. Comparison to Stress-Point SPH Method [Chalk et al. 2020]. (a) Our method using S.C.; (b) a combination of the stress point generation method [Chalk et al. 2020] and our approximate projection method. In this test, an artificial viscosity of 0.05 and a total of 560k particles are used. This test shows that the Stress-Point SPH Method is not robust enough to handle large deformations.

node-stress-point configuration to simulate problems with large displacements and high velocities.

Our dual-particle approach can also be viewed as an alternative stress point SPH method, as virtual particles in our method are introduced to store particle pressures only (note pressure and stress have the same physics dimension). The major difference is that “stress points” in our method are regenerated every timestep while those in other stress point SPH methods follow their associated particles through out the whole simulation. To demonstrate the difference, Figure 22 shows a comparison between our method and the stress-point SPH method given by Chalk et al. [2020] (see Figure 21(a)). Notice the stress-point SPH method is quite unstable because “stress points” can get too close to each other as demonstrated in Figure 21(c). As a result, the simulation finally fails as the fluid undergoes a large deformation while our method remains stable throughout the whole simulation.

7 RESULTS

In this section, we simulate fluids in a variety of different scenarios. All examples are run on a single machine with an Intel i7-8700k CPU and an NVIDIA GeForce RTX2080 GPU, and time-consuming parts (such as neighbor-list searching, incompressibility solver, etc) are parallelized on the GPU. For all scenarios, the time step size is set to $1ms$, the cubic spline kernel [Monaghan 1992] is used, the real and virtual particle spacing δx are $0.005m$ if not specified, the smoothing lengths of real particles and virtual particles are set to $2.5\delta x$. In all cases, the pressure solver terminates when the residual

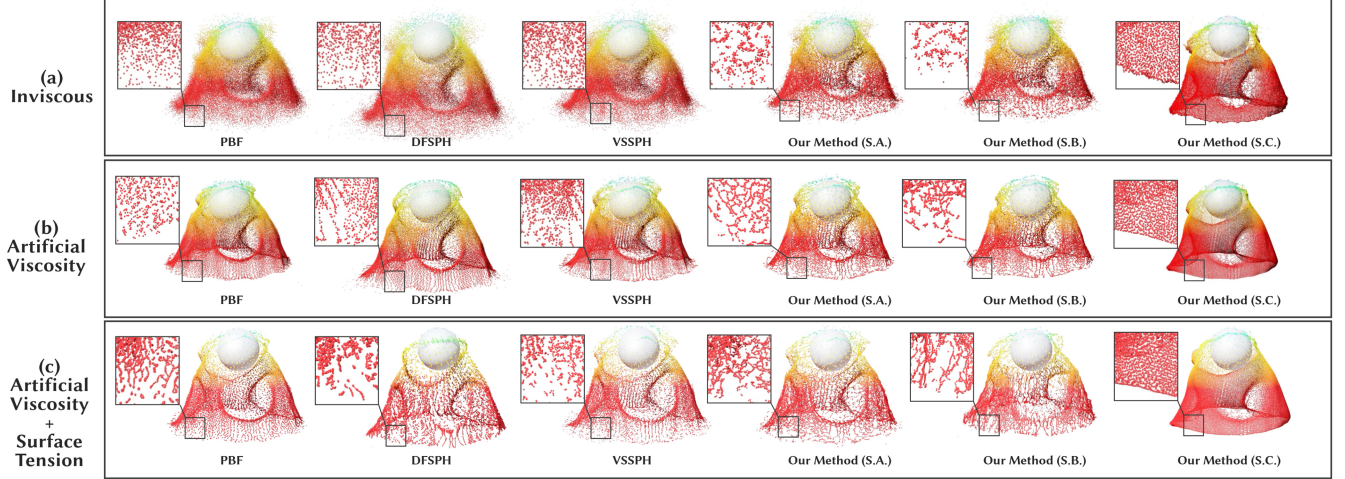


Fig. 23. Comparison of fluid stretching among different methods. A water fish drops on a ball under gravity. (a) all fluids are inviscid; (b) an artificial viscosity of 0.05 is applied; and (c) both the artificial viscosity and the surface tension model [He et al. 2014] are applied.

of conjugate gradient iterations falls below 10^{-4} [Shewchuk et al. 1994].

7.1 Comparison to Other Particle Methods

To verify the effectiveness of our dual-particle approach in improving the tensile stability of particle-based fluids, we also implemented several existing particle methods for comparison, including the **divergence-free SPH method (DFSPH)** [Bender and Koschier 2015], the variational staggered SPH method (VSSPH) [He et al. 2020] which can be viewed as an extension to the staggered particle method [He et al. 2012b], and the PBF [Macklin and Müller 2013] with the adsorption force turned off.

Small-scale thin features of fluids (Figures 23 and 24). In scenarios where the fluid undergoes splashing or forms thin sheets, particles tend to move away from each other, creating negative-pressure regions that trigger tensile instability. Therefore, these scenarios pose significant challenges for most particle-based methods. We setup a liquid collision example to compare our method to several existing approaches, as shown in Figure 23. In this test, the PBF method fails to produce liquid sheets due to a lack of negative pressures. In DFSPH and VSSPH, both suffer from tensile instability due to the irregular distribution of fluid particles. Since the dual-particle method with S.A is equivalent to the original approximate projection method, the simulation results still suffer from tensile instability. By instead using S.B, it can be noticed the tensile instability issue is alleviated, but still not much. Our dual-particle method using S.C shows the best performance in producing liquid films. By adding a little viscosity, Figure 23 shows the simulation results can be significantly improved for our dual-particle method with S.C. Unfortunately, all other methods still fail to generate a liquid film with uniformly distributed particles. We also use the surface tension solver [He et al. 2014] to enhance the small-scale features of fluids, but there are no significant improvements for other methods. The comparison in Figure 23 shows that if the tensile instability is not weakened, the viscosity and surface tension have little effect on improving the stability of small-scale details.

The same results can also be noticed with the simulation in Figure 24. In this scenario, all methods except our method with S.C fail to capture stable thin features of fluids.

Fountain (Figure 25). As shown in Figure 25, the fountain is modeled by the PBF method [Macklin and Müller 2013], VSSPH method [He et al. 2020], and our method with S.C. This scenario contains 1M fluid particles. The XSPH artificial viscosity is used and the parameter is set to 0.05. According to the comparison, the PBF fails to preserve a good shape liquid sheet due to missing attractive interparticle pressure forces. Please note both the VSSPH and our methods have taken into account of the corrective gradient scheme, the semi-analytical boundary [Nair and Tomar 2014], the particle-shifting technique [Xu et al. 2009] and the kernel normalization. The only difference is our method defines pressure samplings on newly created virtual particles. From the comparison, it shows our dual particle framework can effectively reduce the tensile instability and generate a much better well-shaped liquid sheet than that generated by the VSSPH method [He et al. 2020].

7.2 More Demonstrations

Due to the reduction of tensile instability, our dual-particle approach can significantly improve the visual quality of SPH fluids. Figure 27 demonstrates highly viscous fluids. These viscous fluids with a viscosity of $400 Pa \cdot s$ are simulated with the PBF method [Macklin and Müller 2013], VSSPH method [He et al. 2020] and our method, respectively. The surface tension [He et al. 2014] is also applied to the fluids. Our method achieves a more stable result in the comparison. Figure 1(Middle) demonstrates a cup dropped onto the ground and a bulk of water consisting of 504.6k particles inside the cup splashes out. In Figure 1(Right) several water fishes crash onto a solid gargoye, and countless small-scale thin features of fluid are generated. This scenario contains 1.46M particles and the XSPH artificial viscosity is set to 0.1. In Figure 26, several water fishes are dropped onto the ground, it contains a total of 1.50M particles and the parameter of XSPH artificial viscosity is set to 0.05.

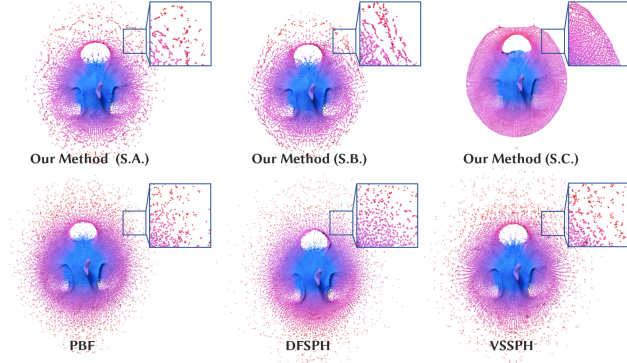


Fig. 24. Two fishes collide in zero-gravity space. The XSPH artificial viscosity model is applied and the parameter is set to 0.05.

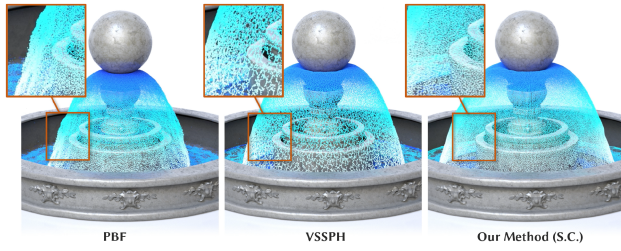


Fig. 25. Fountain. A fountain consisting of 1M particles is simulated by the PBF [Macklin and Müller 2013](Left), VSSPH method [He et al. 2020](Middle) and our method with S.C. An artificial viscosity of 0.05 is applied to all simulations.

8 CONCLUSION

We have presented a dual-particle approach to address tensile instability in a projection-based fluid incompressibility solver. Under the dual-particle framework, we demonstrate how to discretize the approximate projection operator, the divergence operator and the gradient operator used in a PPE on dual particles. Experiments show our method can effectively reduce the particle clustering artifacts introduced by tensile instability. Small-scale thin features, such as liquid streamlets and sheets, can be well preserved in incompressible free-surface flows. Our dual-particle approach makes it possible to build a functional and robust SPH incompressible fluid solver without relying on clamping negative pressure. Besides, tests also demonstrate that the regular virtual particle distribution is beneficial to alleviate the restriction on the time step size for pressure projection in particle fluids.

9 LIMITATIONS AND FUTURE WORK

Our method has several limitations. **First**, it does not preserve the momentum conservation property of traditional SPH methods. However, the loss of momentum is small provided that the virtual particle distribution is regular. **Second**, referring to previous works [Chen et al. 1999; Liu et al. 1995], particle methods with a higher order of consistency, such as the **Corrective Smoothed-Particle Method (CSPM)** or **Reproducing Kernel Particle Methods (RKPM)**, demonstrate enhanced tensile stability in comparison to the standard SPH method. Consequently, we aim to

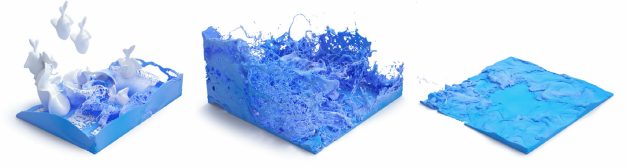


Fig. 26. Fishes. This scenario is simulated by our method using S.C.

integrate higher-order accuracy discretization schemes [Chen et al. 2020a; Reinhardt et al. 2019] into our dual-particle framework to further enhance the performance of our method. **Third**, the virtual particles increased the computational cost of our method, particularly for the S.C. Compared to the standard SPH method such as DFSPH and VSSPH, our method requires more memory and computational costs. However, the uniform distribution of virtual particles facilitates the application of various acceleration techniques, and it would be interesting to study more neighbor querying algorithms, spatial acceleration structures, multigrid methods and other projection acceleration algorithms to improve the efficiency of the dual-particle approach. **Fourth**, S.C. may not be the optimal approach to generate virtual particles due to the meshless nature of the SPH discretization. We shall experiment with other approaches, e.g., by using a GPU-based sparse octree, to

ALGORITHM 1: Dual Particle ISPH

```

while  $t < t_{stop}$  do
  for All real particle  $i$  do
     $\mathbf{v}_i^* \leftarrow \mathbf{v}_i^n + \Delta t \cdot \mathbf{f}$ ;
     $\mathbf{x}_i^* \leftarrow \mathbf{x}_i^n + \Delta t \cdot \mathbf{v}_i^*$ ;
  end
  Generate virtual particle  $I$ ;
  for All real particle  $i$  and virtual particle  $I$  do
    Find real and virtual neighboring particles of  $i$  and  $I$ ;
  end
  Compute the volumes of virtual and real particles (Equations (7) and (8));
  for All virtual particle  $I$  do
    Compute velocity divergence  $\mathcal{D}_I(\mathbf{v}^*)$  (Equation (11));
    Add  $\Lambda_I$  (Equation (13)) to  $\mathcal{D}_I(\mathbf{v}^*)$ ;
  end
  Discretize the pressure Laplacian  $\hat{\mathcal{L}}_I(p)$  (Equation (14));
  Impose the free-surface boundary condition;
  for All virtual particle  $I$  do
    Solve the discretized Poisson pressure equation (Equation (4))
    using the conjugate gradient method, until the residual is less than a specific threshold;
  end
  for All real particle  $i$  do
    Compute pressure gradient  $\mathcal{G}_i(p)$  (Equation (17));
  end
  for All real particle  $i$  do
     $\mathbf{v}_i^{n+1} \leftarrow \mathbf{v}_i^n - \frac{\Delta t}{\rho_0} \cdot \mathcal{G}_i(p)$ ;
  end
end

```

Table 1. **Statistics** In These Scenarios, the Maximum Allowable Relative Errors for Our Methods and VSSPH are Set to 10^{-4} , and The Iteration Number of PBF Method is Fixed to 10

Name ¹	N_{real} ²	$N_{virt.}$ ³	$t_{press.}(s)$ ⁴	$t_{virt.}(s)$ ⁵	$t_{neigh.}(s)$ ⁶	$t_{total}(s)$ ⁷	Frame Number ⁸	Total Time ⁹
Figure 1(Middle)	505k	0.57-1.1M	1.8	0.030	0.107	2.3	2400	1h32min
Figure 1(Right)	1.5M	1.8-9.3M	5.2	0.088	0.375	5.9	2600	4h15min40s
Figure 25(Our Method)	1.0M	<2.6M	2.7	0.016	0.125	2.9	1900	1h31min50s
Figure 25(VSSPH)	1.0M	<2.6M	0.94	—	0.059	1.2	1900	38min
Figure 25(PBF)	1.0M	<2.6M	0.32	—	0.056	0.41	1900	12min59s
Figure 16(a)(S.C)	237k	312-401k	0.62	0.015	0.059	0.74	1600	19min44s
Figure 16(a)(S.B)	237k	237k	0.37	0.045	0.059	0.51	1600	13min36s
Figure 16(a)(S.A)	237k	237k	0.38	—	0.060	0.46	1600	12min16s
Figure 16(a)(VSSPH)	237k	237k	0.41	—	0.015	0.47	1600	12min32s
Figure 16(a)(PBF)	237k	237k	0.08	—	0.014	0.15	1600	4min
Figure 24(S.C)	144k	0.2-1.0M	0.35	0.010	0.032	0.41	600	4min6s
Figure 24(S.B)	144k	144k	0.18	0.022	0.029	0.24	600	2min24s
Figure 24(S.A)	144k	144k	0.16	—	0.030	0.21	600	2min6s
Figure 23(a)(S.C)	72K	97-280k	0.20	0.006	0.024	0.24	600	2min24s
Figure 23(a)(S.B)	72k	72k	0.077	0.019	0.019	0.12	600	1min12s
Figure 23(a)(S.A)	72k	72k	0.078	—	0.019	0.11	600	1min6s
Figure 26	1.5M	1.8-4.1M	4.5	0.086	0.29	5.0	1900	2h38min20s
Figure 27(Our Method)	520k	0.6-1.5M	1.08	0.033	0.10	1.6	1600	42min40s
Figure 27(VSSPH)	520k	0.6-1.5M	0.62	—	0.024	0.97	1600	25min52s
Figure 27(PBF)	520k	0.6-1.5M	0.23	—	0.025	0.66	1600	17min36s

¹S.A., S.B., S.C. respectively represent colocalization strategy, particle shifting strategy and spatially adaptive strategy. ² N_{real} represents the number of real particles; ³ $N_{virt.}$ represents the number of virtual particles; ⁴ $t_{press.}$ represents the average computational cost for solving Equation (4); ⁵ $t_{virt.}$ represents the average computational cost for generating virtual particles; ⁶ $t_{neigh.}$ represents the average computational cost for searching neighboring particles; ⁷ t_{total} represents the average total computational cost for one frame; ⁸“Frame Number” represents the total number of frames for the scenario; ⁹“Total Time” represents the total computational cost for the scenario. The spacing distances of real and virtual particles are all set to $0.005m$, the smoothing lengths are all set to $0.0125m$.

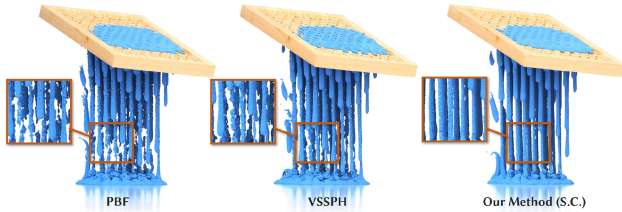


Fig. 27. High viscous fluid. A high viscous fluid consisting of 500k particles is simulated by the PBF Method [Macklin and Müller 2013] (Left), VSSPH method [He et al. 2020] (Middle) and our method with S.C (Right). The viscous model in Liu et al. [2021] and Weiler et al. [2018] and the surface tension model in He et al. [2014] are used for this example.

accelerate generating virtual particles. **Fifth**, it is worthwhile to explore the applicability of the particle splitting strategy [Ando et al. 2012] with the potential to notably improve the stability of small, thin features in fluids. Nevertheless, implementing the particle splitting strategy poses challenges for parallelization on GPUs and requires meticulous integration with the SPH discretization method [Winchenbach and Kolb 2021]. Addressing these challenges will be a focus of our future work. **Finally**, since the dual-particle approach has been proven effective at removing tensile instability for solving fluid incompressibility, we would like to extend this approach to address tensile instability in other physical terms, such as viscosity, elasticity as well as their interference.

ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for their valuable advices on improving the paper.

REFERENCES

- Nadir Akinci, Gizem Akinci, and Matthias Teschner. 2013. Versatile surface tension and adhesion for SPH fluids. *ACM Trans Graph (TOG)* 32, 6, Article 182 (nov 2013), 8 pages. DOI: <https://doi.org/10.1145/2508363.2508395>
- Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile rigid-fluid coupling for incompressible SPH. *ACM Trans Graph (TOG)* 31, 4, Article 62 (jul 2012), 8 pages. DOI: <https://doi.org/10.1145/2185520.2185558>
- Ryoichi Ando, Nils Thurey, and Reiji Tsuruno. 2012. Preserving fluid sheets with adaptively sampled anisotropic particles. *IEEE Transactions on Visualization and Computer Graphics* 18, 8 (2012), 1202–1214.
- Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans Graph (TOG)* 26, 3 (07 2007), 100:1–100:7. DOI: <https://doi.org/10.1145/1275808.1276502>
- Markus Becker and Matthias Teschner. 2007. Weakly compressible SPH for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '07)*. Eurographics Association, Goslar, DEU, 209–217.
- T. Belytschko and Shaoping Xiao. 2002. Stability analysis of particle methods with corrected derivatives. *Computers & Mathematics with Applications* 43, 3 (2002), 329–350. DOI: [https://doi.org/10.1016/S0898-1221\(01\)00290-5](https://doi.org/10.1016/S0898-1221(01)00290-5)
- Jan Bender and Dan Koschier. 2015. Divergence-free smoothed particle hydrodynamics. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '15)*. Association for Computing Machinery, New York, NY, 147–155. DOI: <https://doi.org/10.1145/2786784.2786796>
- J. Brackbill and H. Ruppel. 1986. Flip: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *J Comp Phys* 65, 2 (1986), 314–343.
- Robert Bridson. 2015. *Fluid Simulation for Computer Graphics, Second Edition*. CRC press. 1–256 pages.

- Caitlin Chalk, Manuel Pastor, J. Peakall, Duncan Borman, Andrew Sleight, William Murphy, and Raul Fuentes. 2020. Stress-particle smoothed particle hydrodynamics: An application to the failure and post-failure behaviour of slopes. *Computer Methods in Applied Mechanics and Engineering* 366, 7 (03 2020), 113034. DOI: <https://doi.org/10.1016/j.cma.2020.113034>
- J. K. Chen and J. E. Beraun. 2000. A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems. *Computer Methods in Applied Mechanics and Engineering* 190, 1 (2000), 225–239. DOI: [https://doi.org/10.1016/S0045-7825\(99\)00422-3](https://doi.org/10.1016/S0045-7825(99)00422-3)
- J. K. Chen, J. E. Beraun, and C. J. Jih. 1999. An improvement for tensile instability in smoothed particle hydrodynamics. *Computational Mechanics* 23, 4 (1999), 279–287.
- Xiao Song Chen, Chen Feng Li, Geng Chen Cao, Yun Tao Jiang, and Shi Min Hu. 2020a. A moving least square reproducing kernel particle method for unified multiphase continuum simulation. *ACM Trans Graph (TOG)* 39, 6 (2020), 1–15.
- Yilu Chen, Jonathan Meier, Barbara Solenthaler, and Vinicius C. Azevedo. 2020b. An extended cut-cell method for sub-grid liquids tracking with surface tension. *ACM Trans Graph (TOG)* 39, 6, Article 169 (nov 2020), 13 pages. DOI: <https://doi.org/10.1145/3414685.3417859>
- Andrea Colagrossi. 2005. A meshless Lagrangian method for free-surface and interface flows with fragmentation. *These, Università di Roma* (2005).
- Andrea Colagrossi, Matteo Antuono, and David Le Touzé. 2009. Theoretical considerations on the free-surface role in the smoothed-particle-hydrodynamics model. *Phys. Rev. E* 79, 5 (May 2009), 056701. DOI: <https://doi.org/10.1103/PhysRevE.79.056701>
- Jens Cornelis, Markus Ihmsen, Andreas Peer, and Matthias Teschner. 2014. IISPH-FLIP for incompressible fluids. *Computer Graphics Forum* 33, 2 (2014), 255–262. DOI: <https://doi.org/10.1111/cgf.12324>
- Sharen J. Cummins and Murray Rudman. 1999. An SPH projection method. *J. Comput. Phys.* 152, 2 (1999), 584–607.
- Mathieu Desbrun and Marie-Paule Gascuel. 1996. Smoothed Particles: A new paradigm for animating highly deformable bodies. In *Computer Animation and Simulation '96*. Ronan Boulic and Gerard Héron (Eds.). Springer Vienna, Vienna, 61–76.
- C. T. Dyka and R. P. Ingel. 1995. An approach for tension instability in smoothed particle hydrodynamics (SPH). *Computers & Structures* 57, 4 (1995), 573–580. DOI: [https://doi.org/10.1016/0045-7949\(95\)00059-P](https://doi.org/10.1016/0045-7949(95)00059-P)
- C. T. Dyka, P. W. Randles, and R. P. Ingel. 1997. Stress points for tension instability in SPH. *Internat. J. Numer. Methods Engrg.* 40, 13 (1997), 2325–2341.
- Yun (Raymond) Fei, Qi Guo, Rundong Wu, Li Huang, and Ming Gao. 2021. Revisiting integration in the material point method: A scheme for easier separation and less dissipation. *ACM Trans Graph (TOG)* 40, 4, Article 109 (jul 2021), 16 pages. DOI: <https://doi.org/10.1145/3450626.3459678>
- N. Foster and D. Metaxas. 1996. Realistic animation of liquids. *Graph Mod Imag Proc* 58, 5 (1996), 471–483.
- Thomas-Peter Fries and Ted Belytschko. 2008. Convergence and stabilization of stress-point integration in mesh-free and particle methods. *Internat. J. Numer. Methods Engrg.* 74, 7 (2008), 1067–1087.
- Chuyuan Fu, Qi Guo, Theodore Gast, Chenfanfu Jiang, and Joseph Teran. 2017. A polynomial particle-in-cell method. *ACM Trans Graph (TOG)* 36, 6, Article 222 (Nov. 2017), 12 pages. DOI: <https://doi.org/10.1145/3130800.3130878>
- Jan-Philipp Fürstenau, Bircan Avci, and Peter Wriggers. 2017. A comparative numerical study of pressure-Poisson-equation discretization strategies for SPH. In *12th International SPHERIC Workshop*.
- Ming Gao, Xinlei Wang, Kui Wu, Andre Pradhana, Eftychios Sifakis, Cem Yuksel, and Chenfanfu Jiang. 2018. GPU optimization of material point methods. *ACM Trans. Graph.* 37, 6, Article 254 (dec 2018), 12 pages. DOI: <https://doi.org/10.1145/3272127.3275044>
- Dan Gerszewski and Adam W. Bargteil. 2013. Physics-based animation of large-scale splashing liquids. *ACM Trans Graph (TOG)* 32, 6, Article 185 (nov 2013), 6 pages. DOI: <https://doi.org/10.1145/2508363.2508430>
- Robert A. Gingold and Joseph J. Monaghan. 1977. Smoothed particle hydrodynamics—theory and application to non-spherical stars. *mnras* 181, 3 (11 1977), 375–389. DOI: <https://doi.org/10.1093/mnras/181.3.375>
- Christoph Gissler, Andreas Peer, Stefan Band, Jan Bender, and Matthias Teschner. 2019. Interlinked SPH pressure solvers for strong fluid-rigid coupling. *ACM Trans Graph (TOG)* 38, 1, Article 5 (jan 2019), 13 pages. DOI: <https://doi.org/10.1145/3284980>
- Hitoshi Gotoh and Abbas Khayyer. 2016. Current achievements and future perspectives for projection-based particle methods with applications in ocean engineering. *Journal of Ocean Engineering and Marine Energy* 2, 3 (2016), 251–278.
- Francis H. Harlow. 1962. The particle-in-cell method for numerical solution of problems in fluid dynamics. United States: N, 1962. DOI: [10.2172/4769185](https://doi.org/10.2172/4769185)
- Francis H. Harlow and E. Welch. 1965. Numerical calculation of time dependent viscous flow of fluid with a free surface. *Phys Fluid* 8, 12 (1965), 2182–2189.
- Xiaowei He, Ning Liu, Sheng Li, Hongan Wang, and Guoping Wang. 2012a. Local poisson SPH for viscous incompressible fluids. *Computer Graphics Forum* 31, 6 (2012), 1948–1958.
- Xiaowei He, Ning Liu, Guoping Wang, Fengjun Zhang, Sheng Li, Songdong Shao, and Hongan Wang. 2012b. Staggered meshless solid-fluid coupling. *ACM Trans Graph (TOG)* 31, 6 (2012), 149.
- Xiaowei He, Huamin Wang, Guoping Wang, Hongan Wang, and Enhua Wu. 2020. A variational staggered particle framework for incompressible free-surface flows. arXiv:2001.09421. Retrieved from <https://arxiv.org/abs/2001.09421>
- Xiaowei He, Huamin Wang, Fengjun Zhang, Hongan Wang, Guoping Wang, and Kun Zhou. 2014. Robust simulation of sparsely sampled thin features in SPH-based free surface flows. *ACM Trans Graph (TOG)* 34, 1, Article 7 (Dec. 2014), 9 pages.
- Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. 2018. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans Graph (TOG)* 37, 4, Article 150 (July 2018), 14 pages. DOI: <https://doi.org/10.1145/3197517.3201293>
- Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. 2014a. Implicit incompressible SPH. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (2014), 426–435.
- Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. 2014b. SPH fluids in computer graphics. *Eurographics (State of the Art Reports)* (2014), 21–42.
- Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Trans Graph (TOG)* 35, 4 (Aug. 2015), 51:1–51:10. DOI: <https://doi.org/10.1145/2766996>
- Abbas Khayyer and Hitoshi Gotoh. 2011. Enhancement of stability and accuracy of the moving particle semi-implicit method. *J. Comput. Phys.* 230, 8 (2011), 3093–3118. DOI: <https://doi.org/10.1016/j.jcp.2011.01.009>
- Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2019. Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids. *The Eurographics Association* (2019).
- Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2022. A survey on SPH methods in computer graphics. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 737–760.
- Egor Larionov, Christopher Batty, and Robert Bridson. 2017. Variational stokes: A unified pressure-viscosity solver for accurate viscous liquids. *ACM Trans Graph (TOG)* 36, 4 (07 2017), 101:1–101:11. DOI: <https://doi.org/10.1145/3072959.3073628>
- Moubin Liu and Guiren Liu. 2010. Smoothed particle hydrodynamics (SPH): An overview and recent developments. *Archives of Computational Methods in Engineering* 17, 1 (2010), 25–76.
- Shusen Liu, Xiaowei He, Wencheng Wang, and Enhua Wu. 2021. Adapted SIMPLE algorithm for incompressible SPH fluids with a broad range viscosity. *IEEE Transactions on Visualization and Computer Graphics* 28, 9 (2021), 3168–3179.
- Wing Kam Liu, Sukky Jun, Shaofan Li, Jonathan Adee, and Ted Belytschko. 1995. Reproducing kernel particle methods for structural dynamics. *Internat. J. Numer. Methods Engrg.* 38, 10 (1995), 1655–1679.
- Xiaoxing Liu, Koji Morita, and Shuai Zhang. 2018. An advanced moving particle semi-implicit method for accurate and stable simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 339 (2018), 467–487.
- Leon B. Lucy. 1977. A numerical approach to the testing of the fission hypothesis. *The Astrophysical Journal* 8, 12 (01 1977), 1013–1024.
- Hong-Guan Lyu, Peng-Nan Sun, Xiao-Ting Huang, Shun-Hua Chen, and A-Man Zhang. 2021. On removing the numerical instability induced by negative pressures in SPH simulations of typical fluid–structure interaction problems in ocean engineering. *Applied Ocean Research* 117 (2021), 102938.
- Miles Macklin and Matthias Müller. 2013. Position based fluids. *ACM Trans Graph (TOG)* 32, 4 (2013), 104.
- Joseph J. Monaghan. 1992. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics* 30, 1 (1992), 543–574.
- Joseph J. Monaghan. 2000. SPH without a tensile instability. *J. Comput. Phys.* 159, 2 (2000), 290–311. DOI: <https://doi.org/10.1006/jcph.2000.6439>
- Matthias Müller, David Charypar, and Markus Gross. 2003. Particle-based fluid simulation for interactive applications. In *Proceedings of SCA*. 154–159.
- Ken Museth. 2021. NanoVDB: A GPU-friendly and portable VDB data structure for real-time rendering and simulation. In *ACM SIGGRAPH 2021 Talks (SIGGRAPH '21)*. Association for Computing Machinery, New York, NY, Article 1, 2 pages. DOI: <https://doi.org/10.1145/3450623.3464633>
- Ken Museth, Jeff Lait, John Johanson, Jeff Budsberg, Ron Henderson, Mihai Alden, Peter Cucka, David Hill, and Andrew Pearce. 2013. OpenVDB: An open-source data structure and toolkit for high-resolution volumes. In *ACM SIGGRAPH 2013 Courses (SIGGRAPH '13)*. Association for Computing Machinery, New York, NY, Article 19, 1 pages. DOI: <https://doi.org/10.1145/2504435.2504454>
- Prapanch Nair and Gaurav Tomar. 2014. An improved free surface modeling for incompressible SPH. *Computers and Fluids* 102, 10 (2014), 304–314.
- Rafael Nakanishi, Filipe Nascimento, Rafael Campos, Paulo Pagliosa, and Afonso Paiva. 2020. RBF liquids: An adaptive PIC solver using RBF-FD. *ACM Trans Graph (TOG)* 39, 6 (2020), 170:1–170:13.
- G. Oger, M. Doring, B. Alessandrini, and P. Ferrant. 2007. An improved SPH method: Towards higher order convergence. *J. Comput. Phys.* 225, 2 (2007), 1472–1492. DOI: <https://doi.org/10.1016/j.jcp.2007.01.039>

- P. W. Randles and L. D. Libersky. 2000. Normalized SPH with stress points. *Internat. J. Numer. Methods Engrg.* 48, 10 (2000), 1445–1462. DOI: [https://doi.org/10.1002/1097-0207\(20000810\)48:10<1445::AID-NME831>3.0.CO;2-9](https://doi.org/10.1002/1097-0207(20000810)48:10<1445::AID-NME831>3.0.CO;2-9)
- Karthik Raveendran, Chris Wojtan, and Greg Turk. 2011. Hybrid smoothed particle hydrodynamics. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 33–42.
- Stefan Reinhardt, Tim Krake, Bernhard Eberhardt, and Daniel Weiskopf. 2019. Consistent shepard interpolation for SPH-based fluid animation. *ACM Trans Graph (TOG)* 38, 6, Article 189 (nov 2019), 11 pages. DOI: <https://doi.org/10.1145/3355089.3356503>
- Hagit Schechter and Robert Bridson. 2012. Ghost SPH for animating water. *ACM Trans Graph (TOG)* 31, 4, Article 61 (jul 2012), 8 pages. DOI: <https://doi.org/10.1145/2185520.2185557>
- Jonathan R. Shewchuk. 1994. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Technical Report. Carnegie Mellon University.
- Weixin Si, Jing Qin, Zhuchao Chen, Xiangyun Liao, Qiong Wang, and Pheng-Ann Heng. 2018. Thin-feature-aware transport-velocity formulation for SPH-based liquid animation. *IEEE Transactions on Multimedia* 20, 11 (2018), 3033–3044. DOI: <https://doi.org/10.1109/TMM.2018.2825888>
- Barbara Solenthaler and Renato Pajarola. 2009. Predictive-corrective incompressible SPH. *ACM Trans Graph (TOG)* 28, 3, Article 40 (jul 2009), 6 pages. DOI: <https://doi.org/10.1145/1531326.1531346>
- Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A material point method for snow simulation. *ACM Trans Graph (TOG)* 32, 4 (2013), 102:1–102:10.
- Peng-Nan Sun, Andrea Colagrossi, Salvatore Marrone, Matteo Antuono, and A-Man Zhang. 2018. Multi-resolution Delta-plus-SPH with tensile instability control: Towards high Reynolds number flows. *Computer Physics Communications* 224 (2018), 63–80.
- Peng-Nan Sun, Andrea Colagrossi, Salvatore Marrone, and A-Man Zhang. 2017. The δ plus-SPH model: Simple procedures for a further improvement of the SPH scheme. *Computer Methods in Applied Mechanics and Engineering* 315 (2017), 25–49.
- Peng-Nan Sun, David Le Touze, Guillaume Oger, and A-Man Zhang. 2021. An accurate FSI-SPH modeling of challenging fluid-structure interaction problems in two and three dimensions. *Ocean Engineering* 221 (2021), 108552.
- Peng-Nan Sun, Min Luo, David Le Touzé, and A. Zhang. 2019. The suction effect during freak wave slamming on a fixed platform deck: Smoothed particle hydrodynamics simulation and experimental study. *Physics of Fluids* 31, 11 (2019), 117108.
- Jeff W. Swegle, Darrell L. Hicks, and S. W. Attaway. 1995. Smoothed particle hydrodynamics stability analysis. *J. Comput. Phys.* 116, 1 (1995), 123–134.
- Tetsuya Takahashi, Yoshinori Dobashi, Tomoyuki Nishita, and Ming C. Lin. 2018. An efficient hybrid incompressible SPH solver with interface handling for boundary conditions. *Computer Graphics Forum* 37, 1 (2018), 313–324. DOI: <https://doi.org/10.1111/cgf.13292>
- Naoki Tsuruta, Abbas Khayyer, and Hitoshi Gotoh. 2013. A short note on dynamic stabilization of moving particle semi-implicit method. *Computers and Fluids* 82, 15 (2013), 158–164. DOI: <https://doi.org/10.1016/j.compfluid.2013.05.001>
- Kiwon Um, Seungho Baek, and JungHyun Han. 2014. Advanced hybrid particle-grid method with sub-grid particle correction. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 209–218.
- Renato Vacondio, Corrado Altomare, Matthieu De Lefte, Xiangyu Hu, David Le Touzé, Steven Lind, Jean-Christophe Marongiu, Salvatore Marrone, Benedict D. Rogers, and Antonio Souto-Iglesias. 2021. Grand challenges for Smoothed Particle Hydrodynamics numerical schemes. *Computational Particle Mechanics* 8, 3 (2021), 575–588.
- Xinlei Wang, Yuxing Qiu, Stuart R. Slattery, Yu Fang, Minchen Li, Song-Chun Zhu, Yixin Zhu, Min Tang, Dinesh Manocha, and Chenfanfu Jiang. 2020. A massively parallel and scalable multi-GPU material point method. *ACM Trans Graph (TOG)* 39, 4 (2020), 30–1.
- Marcel Weiler, Dan Koschier, Magnus Brand, and Jan Bender. 2018. A physically consistent implicit viscosity solver for SPH fluids. *Computer Graphics Forum* 37, 2 (2018), 145–155.
- Rene Winchenbach and Andreas Kolb. 2021. Optimized refinement for spatially adaptive SPH. *ACM Trans Graph (TOG)* 40, 1 (2021), 1–15.
- Rui Xu, Peter Stansby, and Dominique Laurence. 2009. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *J. Comput. Phys.* 228, 18 (2009), 6703–6725.
- Meng Yang, Xiaosheng Li, Youquan Liu, Yang Gang, and Enhua Wu. 2017a. A novel surface tension formulation for SPH fluid simulation. *The Visual Computer* 33 (05 2017). DOI: <https://doi.org/10.1007/s00371-016-1274-4>
- Sheng Yang, Xiaowei He, Huamin Wang, Sheng Li, Guoping Wang, Enhua Wu, and Kun Zhou. 2016. Enriching SPH simulation by approximate capillary waves. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '16)*. Eurographics Association, Goslar, DEU, 29–36.
- Tao Yang, Ralph R. Martin, Ming C. Lin, Jian Chang, and Shi-Min Hu. 2017b. Pairwise force SPH model for real-time multi-interaction applications. *IEEE Transactions on Visualization and Computer Graphics* 23, 10 (2017), 2235–2247. DOI: <https://doi.org/10.1109/TVCG.2017.2706289>
- Ling Zhan, Chong Peng, Bingyin Zhang, and Wei Wu. 2019. A stabilized TL-WC SPH approach with GPU acceleration for three-dimensional fluid-structure interaction. *Journal of Fluids and Structures* 86 (2019), 329–353.
- Chi Zhang, Xiangyu Y. Hu, and Nikolaus A. Adams. 2017. A generalized transport-velocity formulation for smoothed particle hydrodynamics. *J. Comput. Phys.* 337, 15 (2017), 216–232.

Received 2 October 2022; revised 7 February 2024; accepted 10 February 2024