

An Adaptive Particle Fission-Fusion Approach for Dual-Particle SPH Fluid

Shusen Liu¹ , Yuzhong Guo¹ , Ying Qiao¹ , Xiaowei He^{†1}

¹ Institute of Software, Chinese Academy of Sciences, China

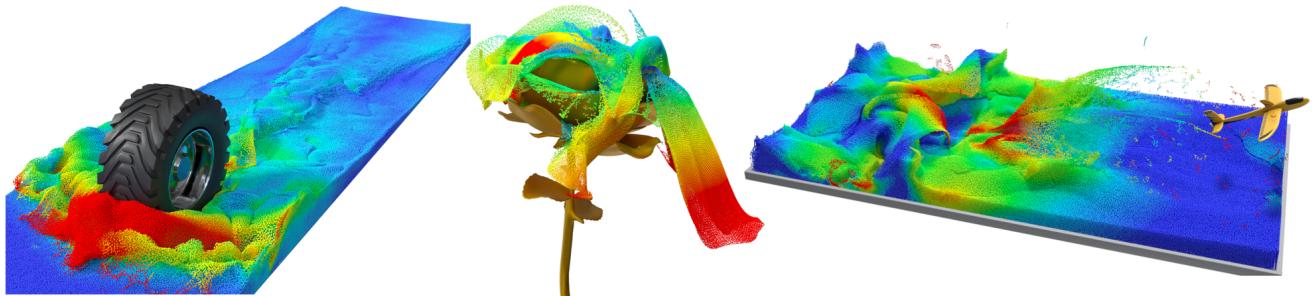


Figure 1: Our method can enhance the stability of fluids under both compression and tension states, and enable efficient and realistic simulation of complex incompressible fluid dynamics. Left: A tire rolling through a water puddle. Middle: Fluids striking a metal rose. Right: A plane crashing into a fluid.

Abstract

Smoothed Particle Hydrodynamics (SPH) is a classical and popular method for fluid simulation, yet it is inherently susceptible to instabilities under tension or compression, which leads to significant visual artifacts. To overcome the limitation, an adaptive particle fission-fusion approach is proposed within the Dual-particle SPH framework. Specifically, in tension-dominant regions (e.g., fluid splashing), the velocity and pressure calculation points are decoupled to enhance tension stability, while in compression-dominant regions (e.g., fluid interiors), the velocity and pressure points are colocated to preserve compression stability. This adaptive configuration, together with modifications to the Dual-particle projection solver, allows for a unified treatment of fluid behavior across different stress regimes. Additionally, due to the reduced number of virtual particles and an optimized solver initialization, the proposed method achieves significant performance improvements compared to the original Dual-particle SPH method.

CCS Concepts

• Animation → Fluid Modeling; Physically Based Animation; • Modeling → Physically Based Modeling;

1. Introduction

Physics-based fluid simulation is a classic and significant research area in Computer Graphics (CG). Among the various methodologies of fluid simulations, the Smoothed Particle Hydrodynamic (SPH) method is an intuitive and widely used approach. SPH utilizes particles to depict the fluid and applies kernel function to discretize the fluid governing equations, offering numerous advantages such as meshless, simplicity, efficiency, low-dissipation,

and mass-conservation [KBST22, KBST19, HJL*25]. However, compared to Eulerian grid methods, SPH is known to face specific numerical instability problems. Specifically, SPH fluids inherently exhibit instability in certain stress ranges. When the fluid stress falls into the regions of instability, the particles may tend to clump together, resulting in significant errors and visual artifacts [SHA95, GK16, VADL*21].

The SPH fluid method is generally susceptible to instabilities in regions experiencing tension or compression states. In conventional SPH methods, negative pressure is usually induced by fluid

[†] Corresponding: xiaowei@iscas.ac.cn

tension state [SHA95], which consequently triggers tension instability. To mitigate this issue, most of existing SPH methods adopt a negative pressure truncation strategy to avoid the tension instability [BK16, ICS^{*}13]. However, the lack of negative pressure hinders these methods from correctly modeling stretching behaviors of fluids, such as splashes and thin sheets. Accordingly, the Dual-particle SPH method [LHG^{*}24] was proposed recently. This method employs additional virtual particles arranged on an Eulerian grid for pressure computation while retaining negative pressures. As demonstrated by Liu et al. [LHG^{*}24], separating virtual particles from real particles effectively weaken tension instability, thereby enabling the realistic capture of small-scaled features of fluids. Nonetheless, the Dual-particle SPH method may exhibit numerical instabilities under positive pressure, leading to deteriorated particle distribution within the fluid and potentially cause non-physical volume loss under compression. (as shown in Fig. 12, 13 and 14). Besides, the excessive number of virtual particles in the Dual-particle SPH method [LHG^{*}24] also lead to a notable deterioration in performance.

In this paper, we present a simple and promising strategy to suppress the instability in tension and compression for SPH fluids, which is achieved through adaptive particle fission and fusion. In the tension region with negative pressure, we split the SPH particle into separate pressure and velocity points (i.e., particle **fission**), mitigating the instability under negative pressure. Conversely, in the compression region with positive pressure, we merge the pressure points with velocity points (named particle **fusion**) to suppress the instability in terms of positive pressure. Ultimately, adaptive particle fission and fusion enable fluids to maintain stress stability throughout most of the fluid domain.

The standard SPH method, which calculates pressure and velocity on the same particles, cannot support our particle fission proposal. Therefore, we opt for the Dual-particle SPH method [LHG^{*}24] as the framework for our particle fission-fusion strategy, which includes virtual particles for pressure calculation and real particles with velocity attributes. Unlike the original Dual-particle SPH method, where grid-based virtual particles cover the entire fluid, we only separate virtual particles from real particles at locations where the fluid is under tension. To better accommodate the uneven distribution of virtual particles and discontinuous transitions between fusion and fission modes, we further enhance the projection solver of the original Dual-particle SPH method [LHG^{*}24]. Additionally, we propose a warm start scheme for the Dual-particle projection solver to reduce the iterations required for pressure solving. The experiments show that our method not only achieves high-quality small-scale fluid details comparable to those of the original Dual-particle SPH method [LHG^{*}24], but also matches the standard SPH projection method in terms of compression stability. Additionally, due to significantly fewer virtual particles and the warm start scheme for projection, our approach achieves significantly better computational efficiency compared to the original Dual-particle SPH method [LHG^{*}24].

We summarize our main contributions as:

- proposing an adaptive particle fission-fusion strategy within the Dual-particle SPH framework that effectively addresses both tension and compression instability.

- adapting the projection-solving scheme based on the Dual-particle SPH framework to stably couple with the particle fission-fusion scheme.
- presenting a warm start scheme to improve the performance of the Dual-particle SPH framework.

2. Related Works

The modeling of incompressibility is one of the most classical and fundamental problems in SPH fluid, and numerous studies have been conducted in the CG community over the past three decades. Regarding strategies to address instability caused by specific stress states in SPH, we categorize three categories and discuss each category separately.

Unidirectional Incompressibility Method. The unidirectional method refers to incompressibility solvers that need to avoid negative pressures. Early SPH incompressibility methods focused on preventing volume loss in fluids, which calculated pressure by employing different forms of equations of state (EOS) to eliminate density errors. Among these schemes, the explicit approaches is easier to implement [MCG03a, BT07], while the implicit approaches can prevent density errors in fluids at larger time steps [HLL^{*}12, BLS11, ICS^{*}13, MM13, HLG^{*}25]. To enhance the accuracy and stability of fluid simulation further, the methods that remove velocity-divergence-errors in fluids are introduced [BK16, TDNL18]. When fluid is in tension state, positive velocity-divergence errors can cause negative pressures to arise. To address issues related to negative pressures, several strategies have been proposed: clamping negative-pressures [BK16] or treating non-negative pressure ($p > 0$) as the constraint in Linear Complementarity Problem (LCP) [ICS^{*}13, TDNL18]. The absence of negative pressure can suppress the tension instability, but it also means these methods are difficult to accurately and stably simulate stretching forms of fluids, such as splashes and other small-scale thin-featured details. In the research fields of mechanics and engineering, explicitly imposing a positive background pressure is a popular scheme to avoid negative pressure and tension instability [Mon00, LHA15a, ZHA17]. However, the background pressure can result in excessive numerical dissipation in SPH fluids [VADL^{*}21], and choosing the value of the artificial pressure also poses a considerable difficulty [CPP^{*}20].

Bidirectional Incompressibility Method. The bidirectional method refers to incompressibility solvers that retain negative pressures. The tension instability issue in the negative pressure region may cause particle clumps and voids. Therefore, such fluid-modeling methods are less favored in the CG area. Researches in mechanics and engineering fields offer several pathways capable of retaining negative stresses without triggering tension instability. The Total-Lagrangian method (TL-SPH method) can entirely prevent tension instability [VRC06, GSK^{*}25], but their kernel functions calculations use material coordinates, restricting their application to large deformation materials such as fluids. Furthermore, stress-point techniques, which compute stress on additional points outside the SPH particles, can alleviate tension instability [DRI97, RL00, FB08, CPP^{*}20]. Nevertheless, the requirement for a uniform-staggered layout between stress points and SPH particles makes it challenging to model fluid. Inspired by the stress-

point method, the Staggered-SPH method is proposed to model fluids [HNW^{*}12, HWW^{*}20], which assumes the existence of stress points only during discretization. However, the staggered-SPH method exhibits limited improvement in the stability of fluids under negative pressures. The Dual-particle SPH method [LHG^{*}24] introduces virtual particles, i.e., stress points, arranged in an Eulerian grid pattern, significantly mitigating the tension instability of fluid. However, Due to the need for a large number of virtual particles, the Dual-particle SPH method [LHG^{*}24] suffers from low computational efficiency. Moreover, its stability under compression remains unverified.

Other Methods to Cope with Stress-instability The particle shifting technique (PST) [LXR12] directly modifies particle positions to enforce a uniform distribution, which can slightly mitigate both tension and compression instabilities and is widely used in engineering fields. However, in small-scale scenarios, PST can lead to non-negligible momentum loss [KGS17]. Tension Instability Control (TIC) [SCM^{*}18, LSH^{*}21] designs a "switch" to suppress tension instability, which uses the symmetric gradient model to obtain an even particle distribution within the fluid domain, and employs the differential gradient approach to mitigate the instability caused by negative pressures. However, due to significant errors introduced by transitions between the two different gradient models, the method requires a small time step size and must be coupled with the PST [LXR12, KGS17]. Two-scale pressure estimation is another kind of "switch" scheme to handle tension instability [HWZ^{*}14]. It activates smaller kernel estimation and anisotropic pressure filtering in thin features of fluids to weaken the tension instability. Particle-based surface tension methods, which address the intermolecular cohesive forces underlying fluid surface tension, have been shown to partially suppress tensile instability [AAT13, YML^{*}17], and improve the fidelity of small-scale fluid features [JWL^{*}23].

Similar to the SPH method, the grid-based method and particle-grid hybrid method may also suffer from tension instability or non-uniform particle distribution. Gerszewski and Bargteil [GB13] proposed eliminating negative pressure to stabilize the stability of FLIP-based fluid simulations under tension; Losasso et al. [LTKF08] introduced SPH solver within grid-based fluids to simulate diffuse regions such as sprays; Kugelstadt et al. [KLTB19] further incorporated a constant density constraint during the projection step to maintain particle distribution uniformity.

3. Revisiting Stress Instability

Before introducing our approach, we briefly review the SPH incompressibility method and its instability in terms of the stress state. Ignoring viscous effects and volume force, the motion of an incompressible fluid should satisfy the Navier-Stokes (NS) equations along with the divergence-free velocity condition. In SPH fluids, the two equations can be written as:

$$\frac{\mathbf{v}_i - \mathbf{v}_i^*}{\delta t} = -\frac{1}{\rho_i} \nabla_i p, \quad (1)$$

$$\nabla_i \cdot \mathbf{v} = 0. \quad (2)$$

Table 1: Table of notation

Variable	Description
i, j	SPH particle or real particle index
I, J	virtual particle index
\mathbf{v}_i	velocity of particle i
\mathbf{v}_i^*	intermediate velocity of particle i
\mathbf{x}_i	position of particle i
r_{ij}	distance between particle i and j
p_j, p_J	pressure of particle j and virtual particle J
p_0	artificial background pressure
h	SPH smoothing length
m_i	mass of particle i
δt	time step size
ρ_i	density on real particle i position
ρ_I	density on virtual particle I position
V_I^v	volume of virtual particle I
ρ_0	rest density
W	SPH kernel function
W'	1st-order derivative of SPH kernel function
W''	2nd-order derivative of SPH kernel function
α, η, ξ	positive constant

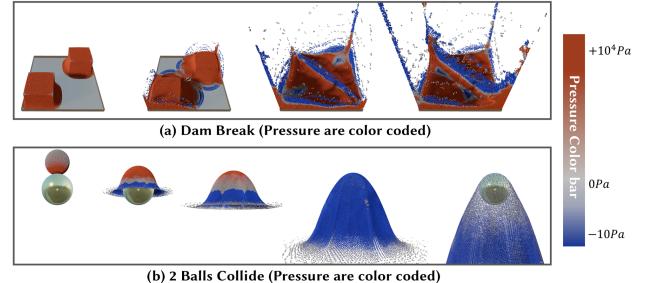


Figure 2: The negative (blue) and positive (red) pressure areas in two fluids. The fluids are modeled by ISPH method [CR99], and the zero-pressure condition are imposed near the fluid free surface [NT14].

By taking the divergence of both sides of Equation 1, we can derive the Pressure Poisson Equation (PPE) for solving the fluid pressure, which takes the following form:

$$\nabla_i^2 p = \frac{\rho_i}{\delta t} \nabla_i \cdot \mathbf{v}^*. \quad (3)$$

Variable descriptions are provided in Table 1. To solve for the pressure, Equation 3 needs to be discretized and transformed into a linear system. Additionally, it is necessary to apply the Dirichlet boundary condition, which typically sets the pressure at the fluid's free surface to zero, to ensure the linear system is solvable [TDNL18, NT14]. Under these conditions, a positive velocity divergence caused by fluid tension will result in negative particle pressure, whereas a negative velocity divergence due to fluid compression will lead to positive particle pressure [SHA95, SCMZ17, CPP^{*}20]. We use the ISPH method [CR99] to model fluids, as

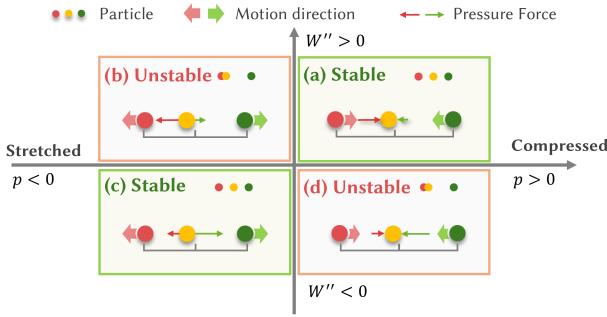


Figure 3: Instability of SPH fluid method. In the pressure gradient calculation (Equation 4), the particle pressure p and the second-derivative of kernel W'' plays a critical role in determining numerical stability [SHA95]. In these sub-figures, red and green particles move toward or away from a yellow particle that is slightly displaced from their central axis. For cases (a) and (c), where $pW'' > 0$, the balanced pressure force guides the yellow particle back to the central position; In contrast, for cases (b) and (d), where $pW'' < 0$, the lopsided pressure forces amplify the positional deviation of the yellow particle, leading to unphysical particle clustering or void formation.

shown in Figure 2. It can be observed that the negative pressure distributions are primarily located in splash areas, whereas positive pressures are confined to the bulk fluid regions.

After pressure calculation, the particle velocity needs to be updated using pressure force, which is computed from the discretization of the pressure gradient:

$$\mathbf{v}_i = \mathbf{v}_i^* - \frac{\delta t}{\rho_i} \nabla_i p = \mathbf{v}_i^* - \frac{\delta t}{\rho_i} \sum_j \frac{m_j}{\rho_j} \left(p_j W'_{ij} \right) \frac{\mathbf{x}_{ij}}{r_{ij}}. \quad (4)$$

During this step, the instability issues become evident. The main cause of the instability is the red term on the right-hand side of Equation 4, namely $p_j W'_{ij} = p_j \frac{dW_{ij}}{dr_{ij}}$ [SHA95]. The instability caused by specific stress states is considered as the mismatch between particle pressure and SPH Kernel. Swegle et al. [SHA95] identified the sufficient conditions leading to instability in pressure gradient approximation, which can be expressed by a very concise

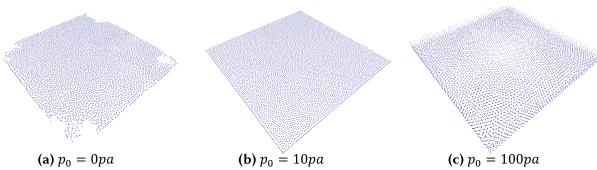


Figure 4: The fluid is modeled by Weakly Compressible SPH method with artificial background pressure (p_0) [AHA13]. (a) No artificial background pressure; (b) Small artificial background pressure ($p_0 = 10\text{pa}$); (c) Excessive artificial background pressure ($p_0 = 100\text{pa}$).

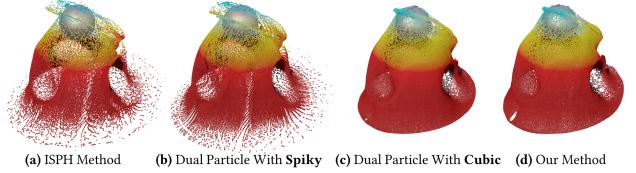


Figure 5: Comparison of different SPH kernel functions. In this test, a fluid fish collides with a rigid ball. (a) The ISPH method [CR99] with the cubic kernel; (b) The original Dual-particle SPH method [LHG*24] with the spiky kernel; (c) The original Dual-particle SPH method with the cubic kernel; (d) Our method with cubic kernel. Compared to the spiky kernel [MCG03b], the cubic kernel possesses regions with negative second derivatives [DA12], which are essential for maintaining stability under negative pressure [SHA95].

inequation:

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

where fluid pressure p is positive in compression and negative in tension [SHA95] in most cases. If W'_{ij} is increasing with r_{ij} (i.e. $W''_{ij} > 0$), particles under compression due to positive pressure cannot maintain the proper distribution (Figure 3(d)). Conversely, if W'_{ij} is decreasing (i.e. $W''_{ij} < 0$), particles under tension due to negative pressure also cannot maintain a correct distribution (Figure 3(b)).

Although the standard incompressible methods for SPH fluids exhibit good stability under compression, they are usually prone

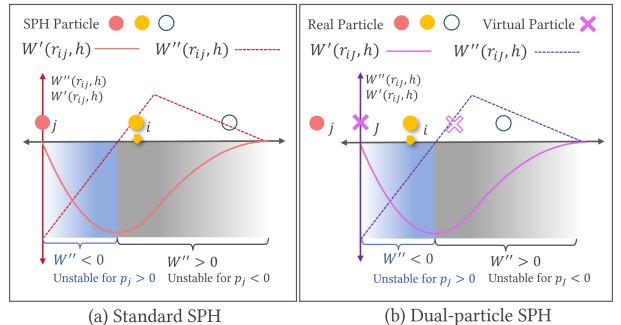


Figure 6: Instability caused by the neighborhood particle pressure in the standard SPH and Dual-particle SPH methods. A cubic kernel function is used in the schematic illustration, with $h = 2.5\text{dx}$. Refer to Swegle et al. [SHA95], in gradient calculation, the neighborhood particle j or J can affect the stability of particle i . (a) In the standard SPH method, a negative pressure of neighboring particle j may lead particle i into an unstable zone, and cause the tension instability of particle i (Figure 3 (b)); (b) In the Dual-particle SPH method, a positive pressure of the virtual particle J may cause the real particle i to enter an unstable zone, and result in the compression instability of particle i (Figure 3 (d)).

Method	Compression Stability	Tension Stability
ISPH	✓	✗
DFSPH	✓	✗
Dual-particle SPH	✗	✓
Our Method	✓	✓

Table 2: Stress stability of different methods. Based on the results shown in Figure 7, 12 and 17, it can be observed that both the DFSPH and ISPH methods are instability under tension, and the original Dual-particle SPH method demonstrates instability under compression. Our approach is equivalent to the ISPH method in the compression region and to the original Dual-particle method in the tension region.

to instability under tension, which is caused by negative pressures [Mon00, MSH04, LSH*21] (The causes of tension instability are illustrated in Figure 6). Therefore, these methods aim to prevent the emergence of negative pressures to weaken tension instability. In CG, the majority of incompressible SPH methods [BK16, ICS*13, TDNL18] achieve this by imposing non-negative pressure constraints (i.e., $p > 0$) during the pressure solving process. However, the absence of negative pressures can lead to the loss of small-scale details in the fluid (as shown in Figure 17 (PBF, DFSPH, VSSPH and ISPH methods)). Alternatively, the Weakly Compressible SPH method [Mon00, SCMZ17, MMC12] employs artificial positive background pressure to counteract negative pressures. Notably, the background pressure can enhance the particle distribution uniformity [SCM*18, CBAM12, LHA15b], but an inappropriate value of background pressure may introduce additional dissipation [VADL*21, VADL*21], as shown in Figure 4.

Diverging from the above conventional methods, the Dual-particle SPH method [LHG*24] deliberately retains negative pressure to improve the quality of fluid splashing. The method [LHG*24] demonstrates that separating the pressure computation points (i.e., virtual particles) from the velocity computation points (i.e., real particles) is an effective approach to suppress tension instability. Therefore, it introduces a grid pattern of virtual particles as pressure nodes, which are separated from real particles that carry velocity and mass properties. When integrated with suitable kernel functions, the Dual-particle approach demonstrates enhanced tension stability (Figure 5). Nevertheless, as shown in Figure 6(b), its efficacy in maintaining compression stability requires further validation.

We designed a simple three-particle test involving both compression and tension scenarios for typical methods, as shown in Figure 7. The results show that ISPH [CR99] and DFSPH [BK16] maintain relatively uniform particle distributions under compression, but fail to do so under tension conditions. In contrast, the original Dual-particle SPH method [LHG*24] exhibits the opposite behavior. In table 2, we compare the three methods and our method in terms of the stability.

4. Adaptive Particle Fission-Fusion Strategy

From the above discussion, it can be seen that existing SPH incompressibility methods can typically only maintain compression

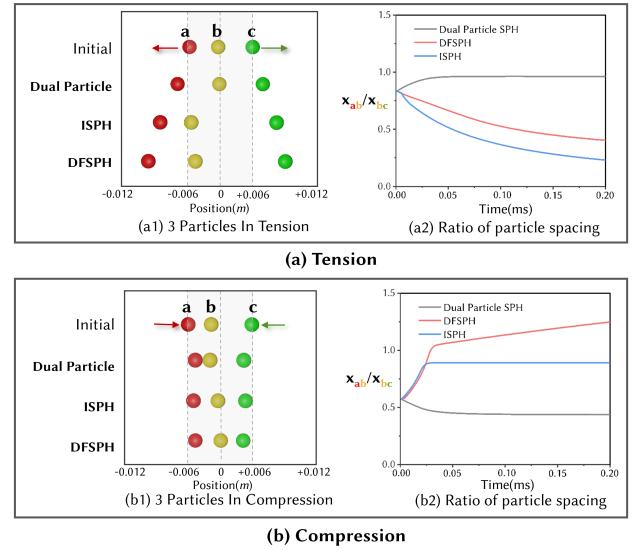


Figure 7: Tug-of-war. The ISPH [CR99], DFSPH [BK16] and original Dual-particle SPH method [LHG*24] are utilized to simulate the tension and compression behaviors of three particles arranged in one dimension. (a) The red and green particles are moved apart; (b) the red and green particles are approached each other. In both comparative scenarios, the yellow particle is initially positioned slightly off-center relative to the red and green particles.

stability or tension stability. So the question is: how can we mitigate instability under both compression and tension region in SPH Fluid?

Although the original Dual-particle SPH method [LHG*24] cannot guarantee stability under compression, the generality of this framework makes it possible to address the aforementioned issues. In the Dual-particle SPH framework, two distinct types of particles are defined: virtual particles serve as pressure calculation points, while real particles carry velocities and masses. When virtual particles are separated from real particles, the method can suppress the tension instability effectively [LHG*24]; When virtual particles coincide with real particles [LHG*24], the method becomes equivalent to the standard ISPH method [CR99], which generally can mitigate compression instability. Therefore, a simple and direct strategy emerges: in the tension regions of the fluid, separate virtual particles from real particles (referred to as **fission**); in the compression regions, let virtual particles overlap with real particles (referred to as **fusion**). Theoretically, this strategy can mitigate instability in most regions of the fluid.

To mitigate the stress instability across the majority of the fluid domain, we first need to develop a method to delineate the particle fission region and particle fusion region. The particle fission region should cover the tension region of fluid, the key criterion for assessing the stretching behaviors of fluids is the velocity divergence approximation:

$$\nabla_i \cdot \mathbf{v} = \frac{1}{\rho_i} \sum_j m_i (\mathbf{v}_j - \mathbf{v}_i) \cdot \nabla_j W_{ij}. \quad (6)$$

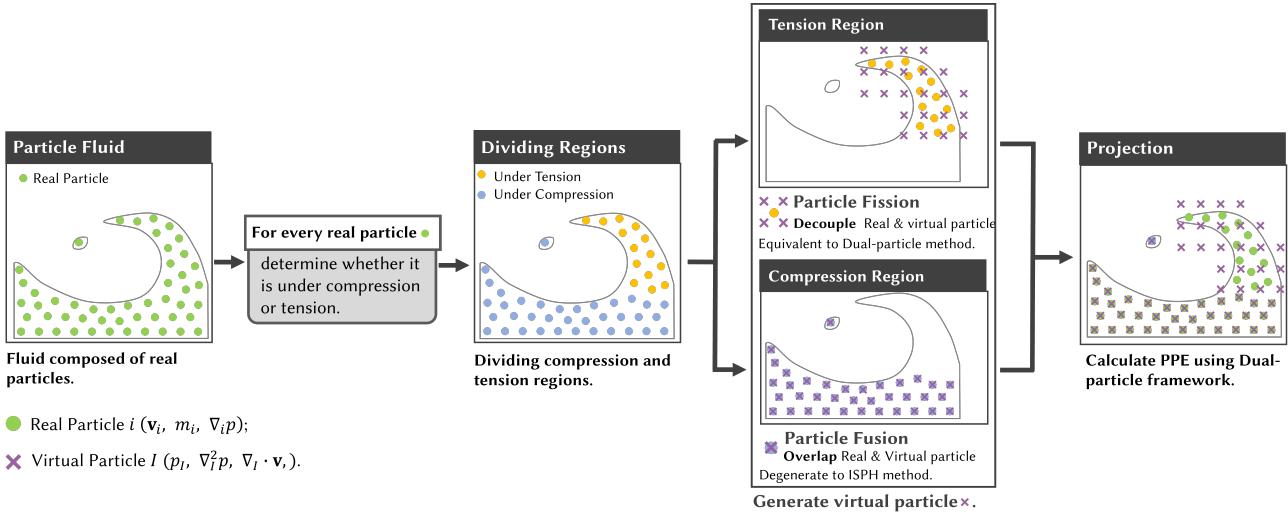


Figure 8: The overview of our method. Our method is based on the Dual-particle SPH framework [LHG*24], where the real particles store the particle masses, velocities and pressure gradients, while the virtual particles carry the pressures, and velocity-divergences, etc. We determine whether a fluid particle is in a tension region based on its velocity divergence and thin-layer feature. In the tension-dominant region, the fission scheme is employed for fluid particles, i.e., decoupling virtual particles from real particles to improve tension stability; Conversely, in the compression-dominant region, virtual particles are made to overlap with real particles to maintain compression stability. After the generation of virtual particles, the projection solver based on the Dual-particle SPH framework [LHG*24] is applied to enforce incompressibility across both tension and compression regions.

$\nabla_i \cdot \mathbf{v} > 0$ means that particle i is under a tension state, accordingly, the particle should be split into real particle and virtual particle to mitigate the tension instability.

Previous research [HWZ*14, LHG*24] and our experiments (Figure 2) show that tension stability is crucial for splashing and thin-sheets features of fluids, and compression stability plays a significant role in maintaining regular particle distribution and ensuring volume conservation within fluids. Therefore, we propose a more robust scheme to trigger particle fission based on thin features of fluids. For each real particle, whether it is within a thin-sheet of fluid is determined by its neighborhood distribution. Following [LHG*23], we employ the equation to estimate the local

distribution properties of each real particle:

$$C_i = \frac{\sum_j \mathbf{n}_{ij} \otimes \mathbf{n}_{ij} W_{ij}}{\sum_j W_{ij}}, \quad (7)$$

where $\mathbf{n}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{r_{ij}}$, and \otimes is tensor product. Next, we compute the singular value decomposition (SVD) of the matrix C_i and extract its singular values, denoted as $\lambda_i = (\lambda_i^1, \lambda_i^2 \dots \lambda_i^d)$, where d is the dimension of space. If $\forall \lambda_i^n \leq \xi$, we will trigger real particle i fission.

To better identify tension-dominant regions, both of the aforementioned criteria for triggering particle splitting are employed. Specifically, whenever a real particle has negative velocity divergence or is positioned in a thin fluid sheet, it undergoes splitting. After determining particle fission, another consideration is the placement of virtual particles. To facilitate parallel generation and deduplication of virtual particles on the GPU, we adopt the same approach as [LHG*24], which involves generating virtual particles at the Eulerian grid nodes adjacent to the real particles undergoing fission. Finally, the implementation of our triggered fission scheme is illustrated in Figure 8.

To evaluate the differences between our proposed method and the original Dual-particle SPH method [LHG*24], we simulated fluid dynamics and visualized the virtual particles, as shown in Figure 9(a) and (b). We also compared the variations in the number of virtual particles used by both methods, as shown in Figure 9(c).

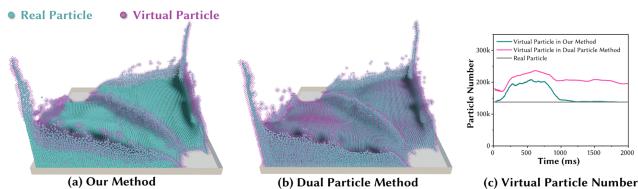


Figure 9: Visualization of virtual particles in the dam-break simulation. (a) Our method decouples virtual particles from real particles in tension areas of a fluid. (b) The original Dual-particle SPH method separates virtual particles from real particles across all areas of the fluid. (c) The variations in virtual particle counts between our method and the original Dual-particle SPH method during this experiment are depicted.

5. Integration with Dual-particle SPH Framework

The original Dual-particle SPH method [LHG^{*}24] is a more general SPH framework that allows pressure computations to be performed not only on conventional SPH particles. Liu et al. [LHG^{*}24] have demonstrated that, within this framework, placing virtual particles away from real particles significantly mitigates the tension instability of fluids, whereas when virtual and real particles coincide, the method reduces to the standard ISPH [GK16]. However, the original Dual-particle SPH method cannot be stably integrated with our adaptive fission-fusion strategy and therefore requires further modification.

5.1. Projection

In our adaptive particle fission-fusion strategy, virtual particles located at the boundary between fission and fusion regions may be in close proximity or overlap. Additionally, certain particles may undergo abrupt transitions between fission and fusion modes during simulations. Integrating this strategy directly into the Dual-particle SPH framework [LHG^{*}24] could result in convergence problems and sudden spikes in the pressure forces exerted on real particles. Therefore, it is necessary to improve the projection solver in the Dual-particle SPH framework.

In the adapted Dual-particle SPH framework, discretizing differential operators using virtual particles requires the use of SPH kernel estimation. If the spatial volume occupied by virtual particles is large, greater weights should be assigned in the SPH estimation to reduce linear errors introduced by the distribution of virtual particles, as shown in Figure 10. Therefore, the volume of the virtual particle should be achieved by SPH kernel summation:

$$V_I^v = \frac{1}{\sum_J W_{IJ}}. \quad (8)$$

And then, the left-hand Laplacian term of PPE (Equation 3) can be discretized as:

$$\nabla_I^2 p = \sum_J V_J^v \frac{p_I - p_J}{r_{IJ} + \eta} \frac{dW_{IJ}}{dr_{IJ}}; \quad (9)$$

the source term of PPE (Equation 3) can be defined as:

$$\frac{\rho_I}{\delta t} \nabla_I \cdot \mathbf{v}^* = \frac{1}{\delta t} \sum_j m_I (\mathbf{v}_j^* - \bar{\mathbf{v}}_I^*) \cdot \nabla_j W_{IJ}, \quad (10)$$

where ρ_I is the fluid mass density at virtual particle position, i.e., $\rho_I = \sum_j m_j W_{IJ}$, and $\bar{\mathbf{v}}_I$ is the intermediate virtual particle velocity refers to [CB00], which can be computed by

$$\bar{\mathbf{v}}_I^* = \frac{\sum_j V_j \mathbf{v}_j^* W_{IJ}}{\sum_j V_j W_{IJ}}, \quad (11)$$

V_j is the volume of real particle j . To prevent the loss of fluid volume, according to the ECS (Error Compensating Source) in MPS (Moving Particle Semi-implicit) method [KG11], the source term in the PPE (Equation 3) can be compensated through the following equation:

$$\frac{\rho_I}{\delta t} \nabla_I \cdot \mathbf{v}^* := \frac{\rho_I}{\delta t} \nabla_I \cdot \mathbf{v}^* + \alpha \frac{\rho_I^v - \rho_0}{\rho_0}, \quad (12)$$

where α is a positive constant, and we recommend that the α be set at the same order of magnitude as $\frac{\rho_I}{\delta t}$.

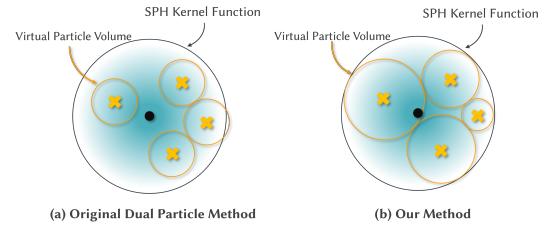


Figure 10: Estimation of virtual particle volume. (a) In the original Dual-particle SPH method [LHG^{*}24], the volume occupied by virtual particles in physical space is not considered in the discretization of the differential operators. (b) In our approach, the volume of virtual particles is calculated by the SPH density estimation to improve the kernel normalization property during differential operator discretizations.

Based on equations 9, 10 and 12, the PPE (Equation 3) has been transformed into a linear system. After applying free-surface boundary conditions (as shown in section 5.2), the linear system becomes symmetric positive definite, enabling efficient pressure calculations for virtual particles via iterative solvers like Jacobi or conjugate gradient methods. After obtaining the virtual particle pressures, the pressure gradient at real particle positions is required to update their velocity:

$$\mathbf{v}_i = \mathbf{v}_i^* - \frac{\delta t}{\rho_i} \nabla_i p = \mathbf{v}_i^* - \frac{\delta t}{\rho_i} \sum_j V_j^v p_j \nabla W_{ij}. \quad (13)$$

From the projection method described above, it can be seen that when the virtual particles coincide with the real particles—i.e., by replacing the uppercase index I, J with the lowercase i, j in the Equations, the projection method is very similar to the standard ISPH method [CR99, YHW^{*}16, GK16, KBST19].

5.2. Boundary

Fluid simulation scenarios typically include fluid regions, solid regions, and air regions. The handling of the fluid-gas boundary (i.e., free surface) and fluid-solid boundary is crucial for stable fluid simulations.

Free-surface boundary. In projection methods, the free-surface zero-pressure boundary condition serves as the Dirichlet boundary for the PPE, which ensures that the equation becomes well-posed with a unique solution. Our approach retains the semi-analytical fluid boundary conditions employed in the original Dual-particle SPH method [LHG^{*}24], which is proposed by [NT14]. We first split the right-hand side of the Equation 9 as follows:

$$\nabla_I^2 p = A_I p_I - \sum_J A_{IJ} p_J, \quad (14)$$

where $A_{IJ} = \frac{V_J^v}{r_{IJ} + \eta} \frac{dW_{IJ}}{dr_{IJ}}$ and $A_I = \sum_J A_{IJ}$. At the simulation beginning, we calculate the coefficient A_I for every virtual particle, and denote the coefficient A_I of a virtual particle in the fluid interior

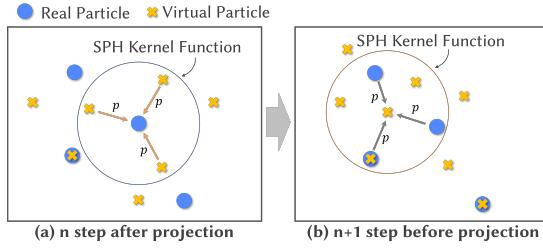


Figure 11: Warm Start. (a) After calculating the pressures for virtual particles, the pressures are mapped to real particles in the fission region; (b) In the subsequent time step, the pressures are interpolated from real particles to virtual particles in the fission region and serve as an initial guess for solving the PPE.

with uniform and sufficient neighborhoods as A^0 . In every simulation step, we check each virtual particle's A_I . If a virtual particle's A_I is smaller than A^0 , the coefficient A_I should be replaced by A^0 :

$$A_I = \max(A_I, A^0). \quad (15)$$

According to [NT14, YHW*16], the aforementioned semi-analytical boundary method is equivalent to generating a sufficient and uniformly distributed set of zero-pressure ghost-air particles outside the fluid's free surface, which significantly improves the convergence rate of the PPE iterative solver.

Solid boundary. For modeling the solid wall boundaries of the fluid, we employ the same ghost-solid particle as used in the original Dual-particle SPH method [LHG*24]. However, when fluid particles move at high speeds or when large time steps are adopted, unphysical penetration of particles into solid boundaries may still occur. To address this issue, we also incorporate a signed distance field (SDF)-based boundary treatment to correct the positions of penetrating fluid particles [CLH*20].

5.3. Warm Start

For iterative solvers of linear systems, a proper initial guess can significantly reduce the number of iterations required for convergence. In SPH-based projection methods, the particle pressures from the previous time step are commonly used as the initial values for discretized PPE calculations [ICS*13]. However, in our framework, the discontinuities of virtual particle positions caused by adaptive particle fission between successive frames pose significantly challenges for initializing the pressure solver.

To obtain a proper initial guess for virtual particles in projection calculation, we propose a warm start operating by storing extra pressures on real particles (Figure 11). For fusion particles that overlap with virtual particles, their pressure is taken directly from the corresponding virtual particle. In the particle fission region, the pressure of the real particle is interpolated from nearby virtual particles using an SPH kernel function. In the next time step, the pressures carried by real particles are interpolated back to nearby virtual particles and used as the initial guess for solving the PPE (Equation 3).

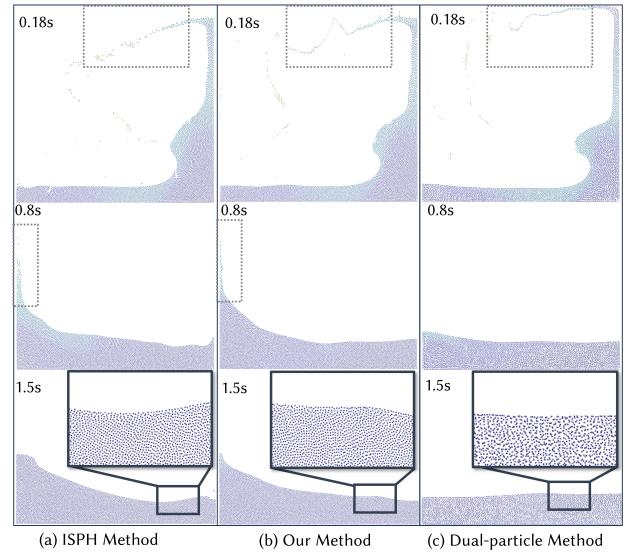


Figure 12: A side-by-side comparison of the ISPH [CR99], our method and the original Dual-particle SPH method [LHG*24]. Due to the compression instability, the fluid modeled by the original Dual-particle SPH method [LHG*24] exhibits highly non-uniform particle distribution under the compression state (1.5s). In contrast, both our method and the ISPH method maintain a more uniform particle distribution during compression(1.5s).

6. Results and Discussions

Our simulation framework was implemented on a workstation equipped with an NVIDIA GeForce RTX 4080 GPU (16GB GDDR), an Intel Core i7-14700K processor (20 cores, 3.4GHz), and 32GB DDR5 RAM (5600MHz). The complete simulation pipeline - including neighborhood queries, incompressibility solving, viscosity calculations [SB12], and solid-fluid interactions - was parallelized on the GPU using CUDA 11.7. For all scenarios, the particle spacing dx is all set to $0.005m$, and the smoothing length h is $0.0125m$. The artificial viscosity, i.e., XSPH [SB12], is used, and **the XSPH parameter is set to 0.05 in all experiments**. In comparison, the ISPH method [CR99], the original Dual-particle SPH method [LHG*24], and our method all employ the standard conjugate gradient method [S*94] to solve the PPE. The implementation of our method is as shown in Appendix (Algorithm 1). The computational costs for the scenarios can be found in Table 3.

6.1. Dam Break

To verify the effectiveness of our method in improving the stability at compression and tension states, we model the dam-break scenarios using the ISPH method [CR99, YHW*16], the original Dual-particle SPH method [LHG*24] and our method.

Figure 12 is a comparison in 2D space. The ISPH method [CR99, YHW*16] is prone to tension instability, resulting in the premature breakup of the splashing fluid (0.18s). Meanwhile, the original Dual-particle SPH method [LHG*24] exhibits significant compres-

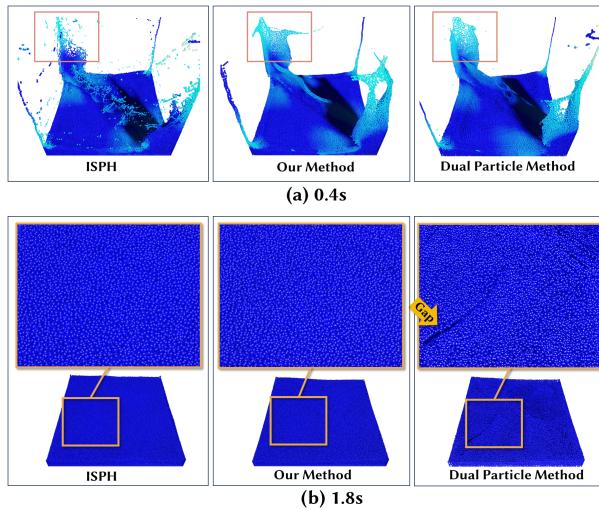


Figure 13: Dam-break of two fluid blocks. The fluid is modeled by the ISPH method, the original SPH Dual-particle method, and our method. (a) The original Dual-particle SPH method and our method produce similar thin-feature details. (b) When the fluid motion slows down and the overall state transitions into compression, the surface of the fluid molded by the original Dual-particle SPH method shows significant gaps.

sion instability, leading to rapid energy dissipation and early stagnation (0.8s), as well as poor particle distribution (1.5s). In this comparison, our method demonstrates better stability in both aspects: it preserves the integrity of the splashing region (0.18s) and maintains a more uniform particle distribution during the compression phase (1.5s).

Figure 13 shows dam-break test in 3D space. In the fluid splashing region, both our method and the original Dual-particle SPH method [LHG*24] successfully maintain the thin-layer structure of the fluid. In contrast, the ISPH method shows significant tension instability, leading to particle clustering. However, as the fluid motion approaches a standstill, the original Dual-particle SPH method [LHG*24] exhibits noticeable void formation, whereas our

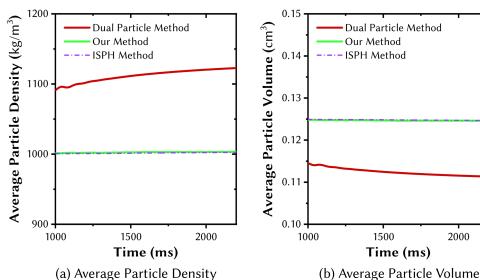


Figure 14: Variation of average density and volume of (real) particle in Figure 13. The rest density is set to $1000\text{kg}/\text{cm}^3$. The volume loss occurs in the original Dual-particle SPH method [LHG*24].

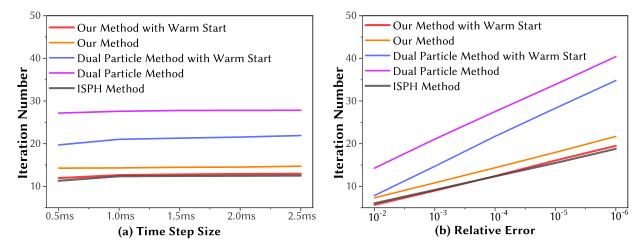


Figure 15: Performance comparison. (a) Average iteration numbers of different methods using a variation of time step size, the relative error threshold for convergence is all set to 10^{-4} , and the total simulation time is 1.5s; (b) Average iteration numbers of different methods using a variation of relative error threshold for convergence, the time step size is all set to 1ms.

method and the ISPH method do not suffer from this issue. Figure 14 depicts the curves of the average particle density changes in the dam-break test (Figure 13), it demonstrates that the original Dual-particle SPH method [LHG*24] fails to maintain the rest density stably within a short period, while our method and the ISPH method do not encounter this difficulty.

Performance. Using the same scenario as in Figure 13, we evaluated the iteration counts of the ISPH method [CR99, YHW*16], the original Dual-particle SPH method [LHG*24], and our method under different time step size and convergence tolerance settings, as illustrated in Figure 15. In our implementation, the ISPH method [CR99, YHW*16] incorporates the same semi-analytical free surface boundary treatment [NT14] used in our method for handling boundaries, and the three methods are all used conjugate gradient method to solve the PPE. The curves indicate that our method achieves notably faster convergence rates compared to the original Dual-particle SPH method [LHG*24]. When integrated with the warm start scheme proposed in this work, the con-

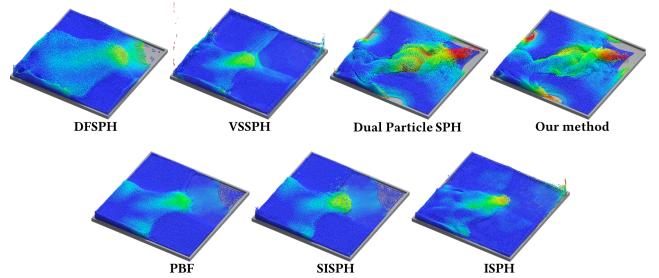


Figure 16: Comparison of dam-break with different methods. In this scenario with 68.9k particles, the original Dual-particle SPH method [LHG*24] and our proposed method both exhibit significantly richer fluid motion details than other methods. The iteration counts of PBF [MM13] and SISPH [HLG*25] are both set to 25; The relative error thresholds of the DFSPH [BK16], VSSPH [HNW*12, HWW*20], Dual-particle SPH [LHG*24], ISPH [CR99] and Our method are all 10^{-3} .

vergence speed is further enhanced and comparable to that of the ISPH method [CR99].

6.2. Comparison among Different SPH Methods

In order to provide a comprehensive evaluation of our approach, we conduct a series of comparisons with a variety of existing methods. Figure 16 shows a fluid modeled by the PBF [MM13], SISPH [HLG*25], ISPH [CR99, YHW*16], DFSPH [BK16], VSSPH [HNW*12, HWW*20] original Dual-particle SPH method [LHG*24] and our method, respectively. In the scenario with a limited number of particles, the fluid simulated by our method and the original Dual-particle [LHG*24] exhibit significantly richer details, due to the better tension stability. Figure 17 illustrates a test with significant fluid splashing. In this scenario, our method and the original Dual-particle SPH method [LHG*24] both simulate nearly identical small-scale fluid details, whereas other methods, including PBF [MM13], DFSPH [BK16], VSSPH [HNW*12, HWW*20], and ISPH [CR99, YHW*16], cannot maintain the thin-sheets of fluid stably. Furthermore, the comparison between our method and the ISPH method combined with surface tension model or implicit viscosity model is shown in the Appendix (Figure 20).

6.3. More Demonstrations

Figure 1(left) and Figure 18 show a simulation of a tire rolling over a puddle with 2.1M real particles, generating significant splashing. Benefiting from the numerical stability of our method in terms of stress, the fluid particles remain uniformly distributed in the initial static state, and rich fluid details are produced after the tire passes through. Figure 1(middle) demonstrates a fluid collides with a metal rose, and generates many splashes. In Figure 1(right) and 19, an airplane is crashing into a turbulent water with 1.69M real particles, and countless small-scale thin features of water are generated.

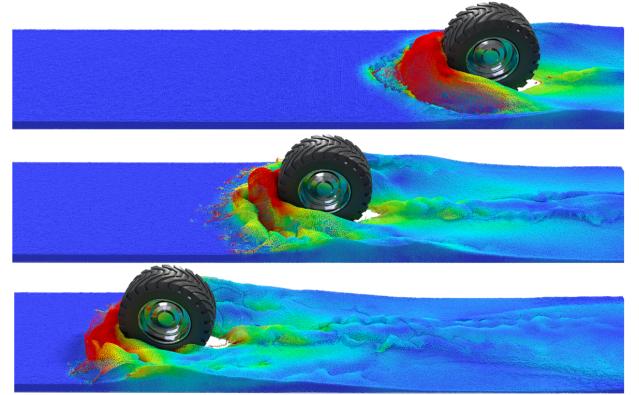


Figure 18: A tyre rolls over a puddle, and generates rich fluid details. There are 2.1M real particles in this scenario, and the particle velocities are color-coded.

Figure	N _p	t _{virtual}	t _{neighbor}	t _{solver}	t _{total}
Fig.1 left and Fig. 18.	2.10M	23.8ms	240ms	1.95s	2.30s
Fig. 1 middle.	1.08M	9.36ms	51.6ms	537ms	630ms
Fig. 1 right and Fig. 19.	1.69M	16.7ms	204ms	1.57s	1.83s
Fig. 5 Our method.	243k	4.83ms	20.1ms	103ms	137ms
Fig. 13 ISPH.	137k	-	3.41ms	35.2ms	44.6ms
Fig. 13 Our method.	137k	3.74ms	10.3ms	35.1ms	52.6ms
Fig. 13 Dual-particle.	137k	3.52ms	12.6ms	102.7ms	135.8ms
Fig. 17 Our method	243k	3.90ms	19.2ms	89.2ms	121ms
Fig. 16 PBF	68.9k	-	1.71ms	14.8ms	19.4ms
Fig. 16 SISPH	68.9k	-	1.69ms	11.7ms	16.5ms
Fig. 16 DFSPH	68.9k	-	1.58ms	17.2ms	21.1ms
Fig. 16 VSSPH	68.9k	-	2.31ms	14.9ms	20.3ms
Fig. 16 ISPH	68.9k	-	6.80ms	22.3ms	33.2ms
Fig. 16 Dual-particle	68.9k	2.62ms	8.42ms	47.0ms	60.4ms
Fig. 16 Our method	68.9k	3.85ms	7.01ms	25.5ms	39.6ms

Table 3: Statistics. N_p represents the number of SPH particles or real particle; t_{virtual} represents the average computational cost for generating virtual particles; t_{neighbor} is the average computational cost to find neighbors; t_{solver} is the average computational cost for the projection solver; t_{total} represents the average total computational cost per time step.

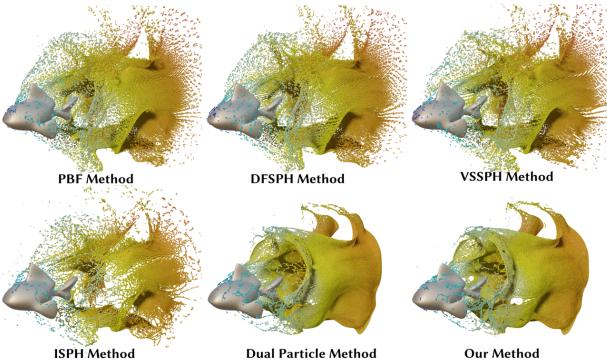


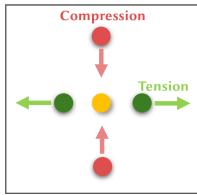
Figure 17: Kissing Fish. During the collision between a fluid fish and a rigid fish, extensive splashing occurs. In this scenario where tension dominates, our approach produces results similar to those of the Dual-particle approach [LHG*24].

7. Conclusion and Future Work

In incompressible SPH fluids, there are inherent numerical instability issues that affect the quality of fluid simulation. Specifically, tension instability can lead to the loss of small-scale motion details in the fluid, while compression instability results in poor particle distribution and often leads to volume loss. To address these issues, we propose an adaptive particle fission–fusion scheme based on the Dual-particle SPH framework, aiming to mitigate both tension and compression instability. In this method, pressure and velocity computation points are dynamically aligned in compression regions of fluids, while separated in regions undergoing tension. When combined with an improved Dual-particle SPH projection solver, the method significantly improves numerical stability across most of the fluid domain. Compared to the standard SPH, our approach demonstrates superior tension stability. Compared to the original Dual-particle SPH method, our approach suppresses compression instability, and achieves higher computational efficiency

by employing fewer virtual particles and a well-designed warm start scheme.

There are still areas that need further refinement in our future work. First, neighboring particles of a given particle may exhibit mixed stress states (both compression and tension). For example, in the right figure, the yellow particle's green neighborhoods are under tension, while its red neighborhoods are under compression. However, our current methodology applies a uniform treatment (i.e., fission or fusion) to the entire neighboring region of the particle. This region-level treatment prevents the complete elimination of numerical instabilities. This issue may be addressed by developing an improved, non-local virtual particle sampling approach. Second, while our method already employs significantly fewer virtual particles compared to the original Dual-particle SPH method, there remains potential for further reduction, which will enhance the computational efficiency. Third, in our implementation, we found that appropriately sampled virtual particles in the Dual-particle SPH framework can greatly improve simulation accuracy. Developing data-driven frameworks for generating optimized virtual particle distributions aligned with fluid dynamics characteristics represents a highly promising research direction.



8. Acknowledgments

We would like to thank anonymous reviewers for their valuable comments. The project was partially supported by the National Natural Science Foundation of China (No. 62302490), the Key Project of the ISCAS (No. ISCAS-ZD-202401), and the Basic Research Project of ISCAS (No. ISCAS-JCMS-202403).

References

- [AAT13] AKINCI N., AKINCI G., TESCHNER M.: Versatile surface tension and adhesion for sph fluids. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–8. [3](#), [13](#)
- [AHA13] ADAMI S., HU X., ADAMS N. A.: A transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics* 241 (2013), 292–307. [4](#)
- [BK16] BENDER J., KOSCHIER D.: Divergence-free sph for incompressible and viscous fluids. *IEEE Transactions on Visualization and Computer Graphics* 23, 3 (2016), 1193–1206. [2](#), [4](#), [5](#), [9](#)
- [BLS11] BODIN K., LACOURSIERE C., SERVIN M.: Constraint fluids. *IEEE Transactions on Visualization and Computer Graphics* 18, 3 (2011), 516–526. [2](#)
- [BT07] BECKER M., TESCHNER M.: Weakly compressible sph for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 209–217. [2](#)
- [CB00] CHEN J., BERNAUN J.: A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems. *Computer Methods in Applied Mechanics and Engineering* 190, 1-2 (2000), 225–239. [7](#)
- [CBAM12] COLAGROSSI A., BOUCASSE B., ANTUONO M., MARONE S.: Particle packing algorithm for sph schemes. *Computer Physics Communications* 183, 8 (2012), 1641–1653. [4](#)
- [CLH*20] CHANG Y., LIU S., HE X., LI S., WANG G.: Semi-analytical solid boundary conditions for free surface flows. *Computer Graphics Forum* 39, 7 (2020), 131–141. [8](#)
- [CPP*20] CHALK C., PASTOR M., PEAKALL J., BORMAN D., SLEIGH A., MURPHY W., FUENTES R.: Stress-particle smoothed particle hydrodynamics: An application to the failure and post-failure behaviour of slopes. *Computer Methods in Applied Mechanics and Engineering* 366 (03 2020). [doi:10.1016/j.cma.2020.113034](#). [2](#), [3](#)
- [CR99] CUMMINS S. J., RUDMAN M.: An SPH projection method. *J. Comput. Phys.* 152, 2 (1999), 584–607. [3](#), [4](#), [5](#), [7](#), [8](#), [9](#), [13](#)
- [DA12] DEHNEN W., ALY H.: Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. *Monthly Notices of the Royal Astronomical Society* 425, 2 (2012), 1068–1082. [4](#)
- [DRI97] DYKA C., RANDLES P., INGEL R.: Stress points for tension instability in sph. *International Journal for Numerical Methods in Engineering* 40, 13 (1997), 2325–2341. [2](#)
- [FB08] FRIES T.-P., BELYTSCHKO T.: Convergence and stabilization of stress-point integration in mesh-free and particle methods. *International Journal for Numerical Methods in Engineering* 74, 7 (2008), 1067–1087. [2](#)
- [GB13] GERSZEWSKI D., BARGTEIL A. W.: Physics-based animation of large-scale splashing liquids. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–6. [3](#)
- [GK16] GOTOH H., KHAYYER A.: Current achievements and future perspectives for projection-based particle methods with applications in ocean engineering. *Journal of Ocean Engineering and Marine Energy* 2 (2016), 251–278. [1](#), [6](#), [7](#)
- [GSK*25] GOTOH T., SAKODA D., KHAYYER A., LEE C. H., GIL A., GOTOH H., BONET J.: An enhanced total lagrangian sph for non-linear and finite strain elastic structural dynamics. *Computational Mechanics* (2025), 1–33. [2](#)
- [HJL*25] HOLZ D., JESKE S. R., LÖSCHNER F., BENDER J., YANG Y., ANDREWS S.: Multiphysics Simulation Methods in Computer Graphics. *Computer Graphics Forum* 44, 2 (2025). [doi:10.1111/cgf.70082](#). [1](#)
- [HLG*25] HE X., LIU S., GUO Y., SHI J., QIAO Y.: A semi-implicit sph method for compressible and incompressible flows with improved convergence. In *COMPUTER GRAPHICS forum* (2025), vol. 44. [2](#), [9](#)
- [HLL*12] HE X., LIU N., LI S., WANG H., WANG G.: Local poisson sph for viscous incompressible fluids. *Computer Graphics Forum* 31, 6 (2012), 1948–1958. [2](#)
- [HNW*12] HE X., NING L., WANG G., ZHANG F., LI S., SHAO S., WANG H.: Staggered meshless solid-fluid coupling. *ACM Trans Graph (TOG)* 31, 6CD (2012), 1–12. [2](#), [9](#)
- [HWW*20] HE X., WANG H., WANG G., WANG H., WU E.: A variational staggered particle framework for incompressible free-surface flows. *arXiv:2001.09421*. (2020). [2](#), [9](#)
- [HWZ*14] HE X., WANG H., ZHANG F., WANG H., WANG G., ZHOU K.: Robust simulation of sparsely sampled thin features in SPH-based free surface flows. *ACM Trans Graph (TOG)* 34, 1 (Dec. 2014), 7:1–7:9. [3](#), [5](#)
- [ICS*13] IHMSEN M., CORNELIS J., SOLENTHALER B., HORVATH C., TESCHNER M.: Implicit incompressible sph. *IEEE transactions on visualization and computer graphics* 20, 3 (2013), 426–435. [2](#), [4](#), [8](#)
- [JWL*23] JESKE S. R., WESTHOVEN L., LÖSCHNER F., FERNÁNDEZ-FERNÁNDEZ J. A., BENDER J.: Implicit surface tension for sph fluid simulation. *ACM Transactions on Graphics* 43, 1 (2023), 1–14. [3](#)
- [KBST19] KOSCHIER D., BENDER J., SOLENTHALER B., TESCHNER M.: Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids. [1](#), [7](#)
- [KBST22] KOSCHIER D., BENDER J., SOLENTHALER B., TESCHNER M.: A survey on sph methods in computer graphics. In *Computer graphics forum* (2022), vol. 41, Wiley Online Library, pp. 737–760. [1](#)

- [KG11] KHAYYER A., GOTOH H.: Enhancement of stability and accuracy of the moving particle semi-implicit method. *J. Comput. Phys.* 230, 8 (2011), 3093–3118. URL: <https://www.sciencedirect.com/science/article/pii/S0021999111000271>, doi:<https://doi.org/10.1016/j.jcp.2011.01.009>. 7
- [KGS17] KHAYYER A., GOTOH H., SHIMIZU Y.: Comparative study on accuracy and conservation properties of two particle regularization schemes and proposal of an optimized particle shifting scheme in isph context. *J. Comput. Phys.* 332 (2017), 236–256. 3
- [KLTB19] KUGELSTADT T., LONGVA A., THUERÉY N., BENDER J.: Implicit density projection for volume conserving liquids. *IEEE Transactions on Visualization and Computer Graphics* 27, 4 (2019), 2385–2395. 3
- [LHA15a] LITVINOV S., HU X., ADAMS N.: Towards consistence and convergence of conservative sph approximations. *Journal of Computational Physics* 301 (2015), 394–401. URL: <https://www.sciencedirect.com/science/article/pii/S0021999115005690>, doi:<https://doi.org/10.1016/j.jcp.2015.08.041>. 2
- [LHA15b] LITVINOV S., HU X., ADAMS N. A.: Towards consistence and convergence of conservative sph approximations. *Journal of Computational Physics* 301 (2015), 394–401. 4
- [LHG*23] LU Z., HE X., GUO Y., LIU X., WANG H.: Projective peridynamic modeling of hyperelastic membranes with contact. *IEEE Transactions on Visualization and Computer Graphics* 30, 8 (2023), 4601–4614. 5
- [LHG*24] LIU S., HE X., GUO Y., CHANG Y., WANG W.: A dual-particle approach for incompressible sph fluids. *ACM Transactions on Graphics* 43, 3 (2024), 1–18. 2, 3, 4, 5, 6, 7, 8, 9, 10
- [LSH*21] LYU H.-G., SUN P.-N., HUANG X.-T., CHEN S.-H., ZHANG A.-M.: 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. 3, 4
- [LTKF08] LOSASSO F., TALTON J., KWATRA N., FEDKIW R.: Two-way coupled sph and particle level set fluid simulation. *IEEE transactions on visualization and computer graphics* 14, 4 (2008), 797–804. 3
- [LXSR12] LIND S. J., XU R., STANSBY P. K., ROGERS B. D.: Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *J. Comput. Phys.* 231, 4 (2012), 1499–1523. 3
- [MCG03a] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), Citeseer, pp. 154–159. 2
- [MCG03b] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), Citeseer, pp. 154–159. 4
- [MM13] MACKLIN M., MÜLLER M.: Position based fluids. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12. 2, 9
- [MMC12] MEHRA V., MISHRA V., CHATURVEDI S.: Tensile instability and artificial stresses in impact problems in sph. In *Journal of Physics: Conference Series* (2012), vol. 377, IOP Publishing, p. 012102. 4
- [Mon00] MONAGHAN J. J.: Sph without a tensile instability. *J. Comput. Phys.* 159, 2 (2000), 290–311. URL: <https://www.sciencedirect.com/science/article/pii/S0021999100964398>, doi:<https://doi.org/10.1006/jcph.2000.6439>. 2, 4
- [MSH04] MELEÁN Y., SIGALOTTI L. D. G., HASMY A.: On the sph tensile instability in forming viscous liquid drops. *Computer Physics Communications* 157, 3 (2004), 191–200. 4
- [NT14] NAIR P., TOMAR G.: An improved free surface modeling for incompressible sph. *Computers and Fluids* 102 (2014), 304–314. 3, 7, 9
- [RL00] RANDLES P. W., LIBERSKY L. D.: Normalized sph with stress points. *International Journal for Numerical Methods in Engineering* 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). 2
- [S*94] SHEWCHUK J. R., ET AL.: An introduction to the conjugate gradient method without the agonizing pain. 8
- [SB12] SCHECHTER H., BRIDSON R.: Ghost sph for animating water. *ACM Trans Graph (TOG)* 31, 4 (jul 2012). URL: <https://doi.org/10.1145/2185520.2185557>, doi:<https://doi.org/10.1145/2185520.2185557>. 8
- [SCM*18] SUN P.-N., COLAGROSSI A., MARRONE S., ANTUONO M., ZHANG A.-M.: Multi-resolution delta-plus-sph with tensile instability control: Towards high reynolds number flows. *Computer Physics Communications* 224 (2018), 63–80. 3, 4
- [SCMZ17] SUN P.-N., COLAGROSSI A., MARRONE S., ZHANG A.-M.: The δplus-sph model: Simple procedures for a further improvement of the sph scheme. *Computer Methods in Applied Mechanics and Engineering* 315 (2017), 25–49. 3, 4
- [SHA95] SWEGLE J. W., HICKS D. L., ATTAWAY S. W.: Smoothed particle hydrodynamics stability analysis. *Journal of computational physics* 116, 1 (1995), 123–134. 1, 3, 4
- [TDNL18] TAKAHASHI T., DOBASHI Y., NISHITA T., LIN M. C.: An efficient hybrid incompressible sph solver with interface handling for boundary conditions. *Computer Graphics Forum* 37, 1 (2018), 313–324. 2, 3, 4
- [VADL*21] VACONDIO R., ALTOMARE C., DE LEFFE M., HU X., LE TOUZÉ D., LIND S., MARONGIU J.-C., MARRONE S., ROGERS B. D., SOUTO-IGLESIAS A.: Grand challenges for smoothed particle hydrodynamics numerical schemes. *Computational Particle Mechanics* 8, 3 (2021), 575–588. 1, 2, 5
- [VRC06] VIGNJEVIC R., REVELES J. R., CAMPBELL J.: Sph in a total lagrangian formalism. *CMC-Tech Science Press-* 4, 3 (2006), 181. 2
- [WKBB18] WEILER M., KOSCHIER D., BRAND M., BENDER J.: A physically consistent implicit viscosity solver for sph fluids. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 145–155. 13
- [YHW*16] YANG S., HE X., WANG H., LI S., WANG G., WU E., ZHOU K.: Enriching sph simulation by approximate capillary waves. In *Symposium on Computer Animation* (2016), vol. 29, p. 36. 7, 8, 9
- [YML*17] YANG T., MARTIN R. R., LIN M. C., CHANG J., HU S.-M.: Pairwise force sph model for real-time multi-interaction applications. *IEEE transactions on visualization and computer graphics* 23, 10 (2017), 2235–2247. 3
- [ZHA17] ZHANG C., HU X. Y., ADAMS N. A.: A generalized transport-velocity formulation for smoothed particle hydrodynamics. *J. Comput. Phys.* 337 (2017), 216–232. 2

9. Appendix

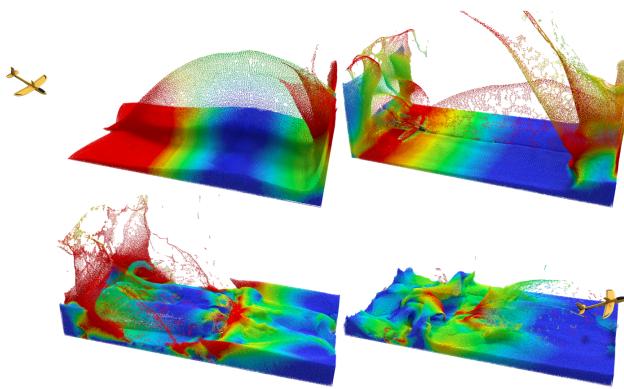


Figure 19: The aircraft plunges dramatically into the tumultuous sea, generating intense splashing. There are 1.69M real particles in this scenario, and the particle velocities are color-coded.

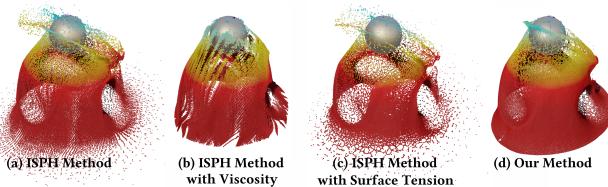


Figure 20: A comparison between our method and the ISPH method with viscosity or surface tension. The result shows that relying solely on viscosity and adhesion-based surface tension fails to effectively enhance the tension stability. (a) The ISPH method [CR99]; (b) The ISPH method with the implicit viscosity [WKBB18] ($\mu = 200 \text{ Pa} \cdot \text{s}$); (c) The ISPH method with adhesion-based surface tension [AAT13]; (d) Our method without the implicit viscosity and surface tension.

Algorithm 1: Adaptive Particle Fission-Fusion for Dual-Particle SPH Fluid

```

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
    for All Real particle  $i$  do
        Compute the thin-sheet feature  $C_i$  (Eq. 7);
        Compute the velocity divergence  $\nabla_i \cdot \mathbf{v}$  (Eq. 6);
        if  $\nabla_i \cdot \mathbf{v} < 0$  or  $\forall \lambda_i^n \leq \xi$  ( $\lambda_i$  is eigenvalue of  $C_i$ ) then
            Determine particle  $i$  undergoes particle fission;
        end
        else
            Determine particle  $i$  undergoes particle fusion;
        end
    end
    for All real particle  $i$  do
        if Real particle  $i$  need to undergo fission then
            generate virtual particles at adjacent grid sites of
            the real particle  $i$ ;
        end
        else
            Generate a virtual particle on the real particle  $i$ 
            position;
        end
    end
Search for the neighborhood of each real and virtual
particle;
Compute the volume of each virtual and real particle;
Compute velocity divergence  $\nabla_I \cdot \mathbf{v}^*$  (Eq. 10)
Compensate the source term of PPE (Equation 12);
Discretize the pressure Laplacian  $\nabla_I^2(p)$  (Eq. 14);
Impose the free-surface boundary condition (Eq. 15);
Solve the discretized PPE (Eq. 3) using the conjugate
gradient method;
for All real particle  $i$  do
    Compute pressure gradient  $\nabla_i p$  (Eq. 13);
     $\mathbf{v}_i^{n+1} \leftarrow \mathbf{v}_i^n - \frac{\delta t}{\rho_0} \cdot \nabla_i p;$ 
end
end

```
