EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 2

A Semi-Analytical Energy Model for Particle-Based Fluid Simulation Involving Complex Moving Boundaries

Junyuan Liu[1], Shusen Liu[2], Yuzhong Guo[2], Ruikai Liang[1], Yin Li[1], and Xiaowei He[2]†

[1]University of Chinese Academy of Sciences, Beijing, China
[2]Institute of Software, Chinese Academy of Sciences, Beijing, China

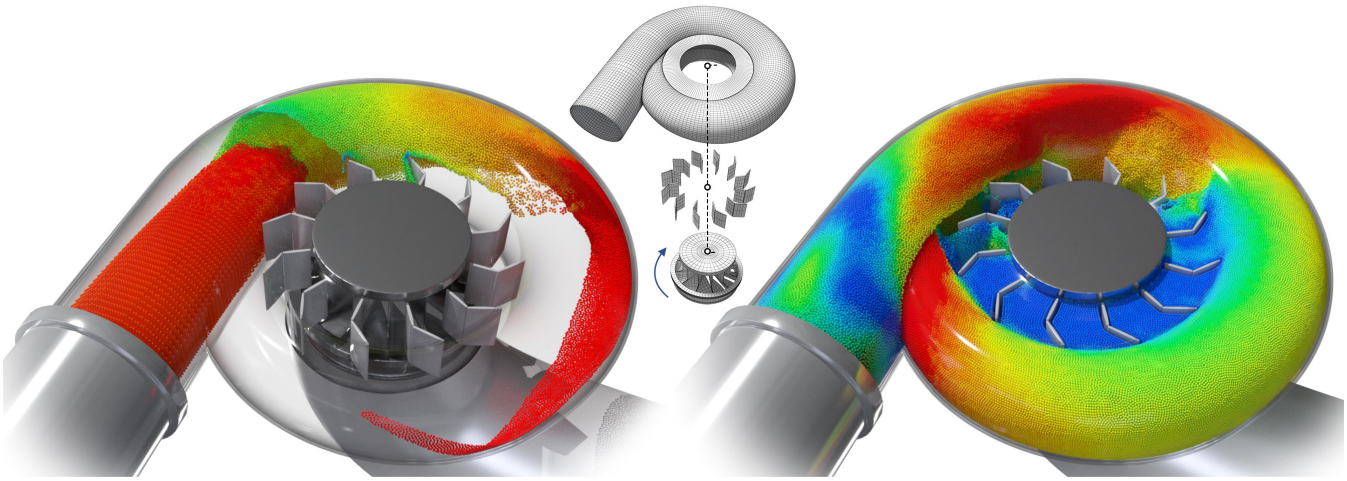


Figure 1: Simulation of water flowing though a hydroturbine consisting of three parts. Our method is capable of coupling fluid particles with dynamic boundaries represented by intricate triangular meshes. In this figure, velocities of particles are color coded.

Abstract

While semi-analytical boundary handling techniques have proven effective for modeling particle-based fluid-solid interactions, they can become unstable when applied to mesh boundaries undergoing dynamic motion or featuring complex, sharp geometries. We propose a novel semi-analytical energy model for boundary handling that unifies fluid simulation and boundary interactions within a variational framework. The model comprises two key components: a semi-analytical bulk energy formulation that mitigates particle deficiency issues in the evaluation of bulk energy, and a nonlocal contact potential that effectively prevents particle penetration into boundaries. Both energy terms are naturally compatible with the Semi-Implicit SPH (SISPH), and a unified Hessian-free solver combined with reduced-order collision detection enables an efficient and stable GPU-based implementation for both fluid dynamics and nonlinear fluid-solid interactions. Furthermore, the unified treatment of fluid bulk energy and boundary energy via the semi-analytical formulation robustly corrects penetrations in practice, even under severe compression scenarios involving complex moving boundaries. Compared with existing semi-analytical boundary treatments, our method is more robust under fast boundary motion and strong compression. Across challenging benchmarks with sharp features, narrow gaps, and moving meshes, it remains stable and penetration-free where prior methods often fail.

CCS Concepts

• ***Computing methodologies*** → ***Physical simulation;***

1. Introduction

Accurate modeling of complex boundaries in fluid simulation is crucial for both computer graphics and engineering applications.

† Corresponding: xiaowei@iscas.ac.cn

Smoothed Particle Hydrodynamics (SPH), a particle-based Lagrangian method, has become a popular choice for simulating free-surface flows, large deformations, and multiphysics scenarios [HJL*25]. Despite its flexibility, boundary handling remains a persistent challenge, particularly when fluid simulation is coupled with complex moving boundaries. Two core difficulties arise: first, fast-moving, complex boundaries can cause particle penetration when large time steps are used. Second, the lack of the Kronecker delta property leads to the underestimation of physical quantities near the boundary, resulting in inaccuracy and instability in regions near the boundary [FM03; KBST22].

To prevent interpenetration between objects, Continuous Collision Detection (CCD) is widely employed [XLYJ23]. While effective at resolving collisions between moving objects, direct adjustments to particle positions using CCD may lead to an unphysical loss of kinetic energy. Moreover, CCD requires continuous prevention of interpenetration throughout the simulation; if violated, the method becomes unreliable and computationally expensive to restore to a feasible state. Alternatively, other boundary handling strategies try to mitigate the challenges with improved boundary integral strategy. *Ghost particle methods* approximate missing boundary contributions by sampling additional particles on or near the solid surface [GPB*19; BGI*18; PT23], but they incur sampling overhead and struggle with sharp or detailed geometries. *Semi-analytical boundary handling* instead replaces the missing kernel support with surface integrals over boundary geometry [FM15; CLH*20; WAK20], improving geometric fidelity, yet penetration remains challenging under large time steps or extreme compression (e.g., as shown in Figure 2).

We propose to model both fluid simulation and boundary interactions under an unified variational framework. We first introduce a semi-analytical energy model for boundary handling, which consists of two key components. The semi-analytical bulk energy term is introduced to address the particle deficiency in bulk energy evaluation, while a finite-horizon nonlocal contact potential is introduced to robustly correct penetrations. Here “nonlocal” follows the standard usage in nonlocal continuum mechanics (Peridynamics): the response at a point is determined by integrating interactions over a finite neighborhood (support radius h), rather than by purely local derivatives [Sil00; LHG*23]. Unlike other contact potential models, such as the IPC model [LFS*20a], the nonlocal contact potential uses a signed distance value rather than a strictly positive distance. Thus, to account for complex moving boundaries, there is no need to perform exact CCD tests at every iteration during numerical optimization. We therefore propose a reduced-order CCD algorithm to detect collisions between particles and boundary triangular meshes, which needs to be performed only once at the beginning of each time step. As both the semi-analytical bulk energy and the nonlocal contact potential are adapted into a semi-implicit successive substitution method, we finally obtain a Hessian-free, GPU-friendly substitution-type solver to simulate particle-based fluids involving complex moving boundaries.

To summarize, the contributions of this work include

- **A semi-analytical energy model** for boundary handling that consists of the semi-analytical bulk energy addressing the particle deficiency issue and the nonlocal contact potential addressing the particle penetration issue;
- **A reduced-order CCD algorithm** which is simplified to solving a quadratic equation, achieving better robustness against collisions between particles and boundary triangles;
- **A substitution-type iterative solver** that is Hessian-free and GPU-friendly for solving nonlinear optimization problems, capable of handling complex, moving boundaries in particle-based fluid simulations.

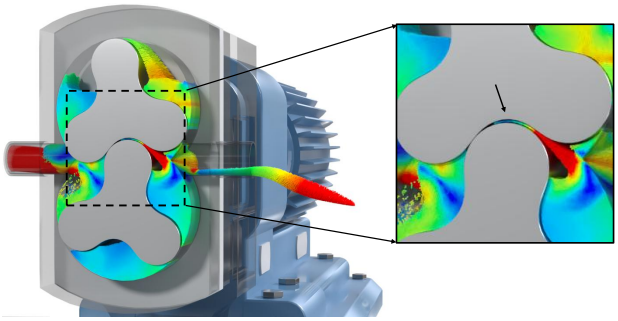


Figure 2: *Simulation of fluid flowing through a rotor pump with velocity color-coded. The proposed semi-analytical boundary handling method achieves stable, intersection-free motion even when the cavity is strongly compressed by the two rotors.*

Through experiments, our method demonstrates improved robustness under compression, accurate capture of boundary details, and stable performance with moving or deforming boundaries, while retaining computational scalability.

2. Related Work

2.1. Incompressibility in SPH

To enforce fluid incompressibility, a common approach is the non-iterative Weakly Compressible SPH (WCSPH) method [BT07; Mon94], which has been extended with techniques like adaptive particle resolution [WHK17; WK21], asynchronous time integration [RHEW17] and higher-order kernels [CLC*20]. Nevertheless, the WCSPH method relies on an explicit equation of state (EOS) for pressure and suffers from significant stability limitations, particularly when a large time step is employed. To address the stability issues, "Predictive–corrective" methods [SP09; HLL*12] have been proposed to enforce incompressibility by iteratively correcting density errors. Alternatively, the projection method [IOS*14], which solves a linearized global pressure Poisson equation, achieves improved efficiency and stability. The divergence-free SPH (DFSPH) method [BK15] further demonstrates that enforcing both constant density and divergence-free velocity constraints on fluid particles enhances numerical stability. To capture finer small-scale fluid motions, the dual-particle SPH method [LHG*24] proposes a framework to retain negative pressures without inducing tensile instability.

In contrast to the aforementioned pressure-solving methods, the Position-Based Fluid (PBF) method [MM13], inspired by Position-Based Dynamics (PBD) [BKCW14; MMC16], directly adjusts particle positions to enforce constant density constraints. PBF is stable, fast, and easy to implement, but it exhibits sensitivity to pa-

rameters such as particle radius and time step size. To address this limitation, the SISPH method [HLG*25] is introduced, which employs the semi-implicit successive-substitution method to optimize the density-related variational energy of fluids. Compared to PBF, SISPH is more robust to variations in the time step size and particle radius, and achieves faster convergence.

2.2. Boundary Handling in SPH

In the particle-based method, the treatment of the fluid-solid boundary is crucial for simulation stability. Compared to the grid-based method (including Eulerian grid-based solvers such as the lattice Boltzmann method (LBM)) [SHM22; FGW*21; AC10; CD98], the particle-based method using kernel-based approximations requires a larger support radius [HBFR04; KBST22]. Near the boundary, kernel supports of fluid particles will be truncated, which can lead to underestimated density and visible artifacts. Therefore, naive boundary handling techniques, such as distance-based penalty forces or position corrections [Mon94; LHWW21], often result in particle clustering and oscillations near solid interfaces, motivating more robust boundary treatments, such as kernel renormalization corrections to mitigate truncation errors [BL99; RL96b]. The ghost-solid particle approach is effective and widely used to treat solid boundaries by placing ghost particles within the solid domain [AIA*12; SB12; BGPT18; BGI*18], for instance via generalized wall boundary conditions based on boundary particles [AHA12]. Additionally, the ghost particle approach can be easily extended to model two-way coupling of fluid-solid interactions [GPB*19].

The ghost particle approach improves the accuracy of calculations near the boundary but requires careful sampling. To avoid explicit sampling, implicit solid boundary methods based on eulerian grids are devised. Koschier and Bender [BKWK19] propose to store density contributions of solid boundaries on a spatial grid, and correct the estimations of fluid particle densities. After that, Bender et al. [BKWK20] suggest using the volume contributions of boundaries instead of the density contributions in grids, which can better integrate with existing SPH methods and tolerate lower grid resolutions. Winchenbach et al. [WAK20] propose to use signed-distance-field (SDF) based corrections assuming locally planar walls, which integrate well with the adaptive particle resolution SPH method.

To model fluid–solid interactions involving complex and sharp geometries, a straightforward approach is to use surface meshes for boundary representation, which requires semi-analytical boundary techniques to ensure accurate near-wall fluid behavior. Early semi-analytical methods can only treat simple geometries [KBLP04], Ferrand et al. [FLR*13] extend the method to unified 3D cases. Fujisawa et al. [FM15] integrate the semi-analytical boundary into position-based methods. To further improve computational efficiency and robustness in handling complex boundaries, Chang et al. [CLH*20] convert volume integrals inside solids into surface integrals over mesh triangles.

Compared to other methods, the semi-analytical boundary is well-suited for handling complex solid boundaries, and can directly utilize surface geometric models for simulation without requiring pre-computation [CLH*20]. Nevertheless, since solids are represented as surface meshes, fluid particles may easily penetrate into the solid region due to excessively large time steps or high velocities. The approaches to prevent unphysical penetration between fluid and solid [XLYJ23] rely on a combination of two techniques: the Continuous Collision Detection (CCD) [RKC02; WTTM15], and barrier-energy methods such as incremental potential contact (IPC) [LFS*20b], which ensure robust non-intersection guarantees for solid objects. Nevertheless, the CCD typically incurs significant computational overhead [BEB12].

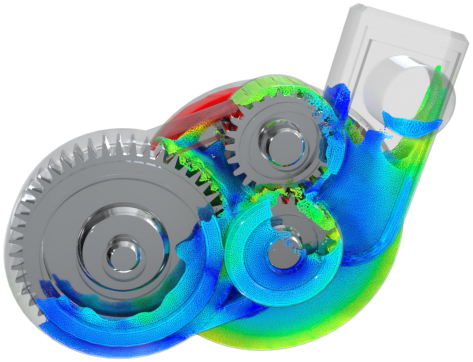


Figure 3: *Fluid flow through a multi-gear assembly with velocity color-coded. The narrow clearances between teeth and the simultaneous motion of multiple gears exemplify sharp features and multi-contact interactions that challenge boundary handling methods. Notably, the gear teeth include many narrow convex tips that undergo high-speed fluid impact, serving as a complementary convex sharp-corner stress test to Figure 16.*

3. Background

3.1. Semi-Analytical Boundary

In SPH, the value of a function $f(\mathbf{x})$ at particle i is approximated as

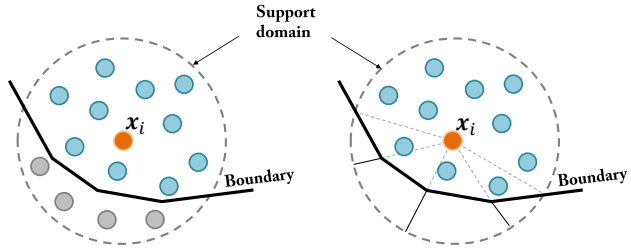
$$f(\mathbf{x}_i) = \sum_j V_j f(\mathbf{x}_j) W(r_{ij}, H), \tag{1}$$

where W is the kernel function, V_j is the volume associated with particle j, $r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the distance between particles i and j, and H is the smoothing length. In regions away from boundaries, the particle approximation can provide accurate estimates under sufficiently smooth fields and uniform particle distributions. However, near boundaries the kernel support domain is truncated, leading to the well-known *particle deficiency problem* [RL96a; FLR*13; KBST22]. This issue has motivated a wide range of remedies, including approaches that explicitly integrate over the boundary geometry as well as methods that densely sample the boundary interior using ghost particles.

Ghost particle methods [SB12; GPB*19] address this problem by sampling particles outside the boundary, as shown in Fig. 4a. The corrected approximation is expressed as

$$f(\mathbf{x}_i) = \underbrace{\sum_j V_j f(\mathbf{x}_j) W(r_{ij}, H)}_{f^{\mathcal{F}}(\mathbf{x}_i)} + \underbrace{\sum_{j'} V_{j'} f(\mathbf{x}_{j'}) W(r_{ij'}, H)}_{f^{\mathcal{B}}(\mathbf{x}_i)}, \tag{2}$$

where $f^{\mathcal{F}}$ and $f^{\mathcal{B}}$ denote contributions from the interior fluid and



(a) Ghost particle (b) Semi-analytical boundary

Figure 4: *(a) Ghost particle sampling; (b) semi-analytical boundary integral.*

the boundary particles, respectively. To avoid sampling the boundary, the semi-analytical method [CLH*20] instead computes the boundary contribution via integral:

$$\begin{aligned} f^{\mathcal{B}}(\mathbf{x}) &= \int_{\mathcal{B}} f(\mathbf{x}')W(r,H)\,dV \\ &= \int_{\mathcal{B}} \nabla_{\mathbf{x}'}\cdot\mathbf{G}(\mathbf{x},\mathbf{x}')\,dV \\ &= \int_{\Omega}\left(\int_{r(\theta,\varphi)}^{H} g(r)r^2 dr\right)\sin\theta\, d\theta d\varphi \\ &= \int_{\Omega} G(r)|_{r(\theta,\varphi)}^{H}\, d\Omega, \end{aligned} \tag{3}$$

where $\mathbf{G}(\mathbf{x},\mathbf{x}')$ satisfies $\nabla_{\mathbf{x}'}\cdot\mathbf{G}(\mathbf{x},\mathbf{x}') = f(\mathbf{x}')W(r,H)$, Ω is the solid angle covered by boundary $\mathcal{B}$, $G'(r) = g(r)r^2$, $G(r)|_{r(\theta,\varphi)}^{H} = G(H) - G(r(\theta,\varphi))$, and $d\Omega = \sin\theta\, d\theta d\varphi$. For $f(\mathbf{x}') \equiv 1$, Eq. 3 simplifies to the volume enclosed by the boundary. When the boundary is discretized using a triangular mesh, the integral can be evaluated over the discrete boundary elements, as illustrated in Fig. 4b.

3.2. Semi-Implicit SPH

Standard SPH methods enforce incompressibility either by solving global pressure equations or iteratively correcting density errors to enforce a constant density field. Since constraint-based solvers are sensitive to the time step and the number of solver iterations, the Semi-Implicit SPH (SISPH) method reformulates the enforcement of fluid incompressibility as an energy minimization problem [HLG*25], where the total energy is defined as

$$E(\mathbf{x}) = \frac{1}{2(\Delta t)^2}\mathbf{M}\|\mathbf{x}-\mathbf{x}^*\|^2 + \mu\sum_i B(\lambda_i), \tag{4}$$

where Δt is the time step, μ is a constant controlling fluid compressibility, $\mathbf{M}$ is the mass matrix, and $\mathbf{x}^*$ is the predicted position, $B(\lambda_i)$ is the bulk energy and λ_i is the density ratio given by

$$\lambda_i = \frac{\rho_i}{\rho_0}, \qquad \rho_i = \sum_j m_j W(\|\mathbf{x}_i - \mathbf{x}_j\|, H) \tag{5}$$

with ρ_0 denoting the rest density and m_j the mass of particle j , and H is the kernel support radius. Solving the optimization problem in Eq. 4 is equivalent to finding the stationary point for the following equation

$$\mathbf{x}_i = \mathbf{x}_i^* - \mu\frac{(\Delta t)^2}{m_i}\dot{B}(\lambda_i)\frac{\partial\lambda_i}{\partial\mathbf{x}_i}. \tag{6}$$

Here $\dot{B}(\lambda) = \frac{dB}{d\lambda}$ denotes the derivative of B with respect to its scalar argument λ for compactness.

Since Eq. 6 contains nonlinear terms, how to solve the above nonlinear problem efficiently still remains a challenging task. To overcome this issue, He et al. [HLG*25] propose to incorporate the *Semi-Implicit Successive Substitution Method* (SISSM), which splits the nonlinear term $\dot{B}(\lambda_i)$ into a positive part $\dot{B}^+(\lambda_i) \geq 0$ and a negative part $\dot{B}^-(\lambda_i) \leq 0$. The positive part is treated implicitly while the negative part is kept explicit, making each iteration simple and intuitive.

Substituting $\dot{B}(\lambda_i) = \dot{B}^+(\lambda_i) + \dot{B}^-(\lambda_i)$ into Eq. 6 gives

$$\mathbf{x}_i^{k+1} = \frac{1}{A_i^{k+}}\left[\mathbf{x}_i^* + \sum_j \left(A_{ij}^{k-}(\mathbf{x}_j^k - \mathbf{x}_i^k) + A_{ij}^{k+}\mathbf{x}_j^k\right)\right], \tag{7}$$

where A_i^{k+} denotes the normalization factor $A_i^{k+} = 1 + \sum_j A_{ij}^{k+}$, $\mathbf{x}_i^* = \mathbf{x}_i^n + \Delta t\,\mathbf{v}_i^*$ is the predicted position, $\mathbf{x}_i^k$ and $\mathbf{x}_i^{k+1}$ denote the positions of particle i at iteration k and $k+1$. The coefficients A_{ij}^{k+} and A_{ij}^{k-} follow directly from the decomposition of $\dot{B}(\lambda_i)$ and are written as

$$A_{ij}^{k\pm} = -\mu\frac{(\Delta t)^2}{m_i}\dot{B}^{\pm}(\lambda_i^k)\frac{\partial W}{r_{ij}\partial r_{ij}}, \tag{8}$$

where m_i is the particle mass, and λ_i^k is the normalized density ratio at iteration k.

Equation 7 therefore gives a purely fixed-point iteration in which positive terms are treated implicitly and negative terms are treated explicitly; convergence is consistently observed in our experiments with a simple line search. This strategy eliminates the need to assemble a large coefficient matrix, resulting in efficient, GPU-friendly iterations. Moreover, it can be extended to incorporate energy terms arising from complex solid boundaries.

4. A Semi-Analytical Energy Model for Boundary Handling

We model boundary handling within a purely Lagrangian framework, where the fluid domain is discretized into particles and the solid boundary is represented by a triangular mesh. Our objective is to unify two essential requirements: first, the enforcement of incompressibility both inside the fluid and near the boundary; and second, the prevention of particle penetration into the solid. Directly applying position-based dynamics often introduces sensitivity to numerical parameters (e.g., the time step size Δt) and can lead to nonphysical energy dissipation. To avoid this, we instead formulate the problem variationally, inspired by projective dynamics [BML*14] and the semi-implicit SPH method [HLG*25].

The governing minimization problem is written as

$$\min_{\mathbf{x}} \frac{1}{2(\Delta t)^2}\mathbf{M}\|\mathbf{x}-\mathbf{x}^*\|^2 + \mu\sum_i B(\lambda_i) + \kappa\sum_c C(\gamma_{ic}), \tag{9}$$

where $B(\lambda_i)$ represents the bulk energy and $\lambda_i = \rho_i/\rho_0$ denotes the density ratio. Here, $C(\gamma_{ic})$ represents the contact potential, which prevents particle penetration, with $\gamma_{ic} = d_{ic}/\hat{d}$ defined as the ratio of the signed distance d_{ic} to the boundary over a user-specified thickness $\hat{d}$. The parameters μ and κ are weighting coefficients that balance the contributions of bulk and contact energies.

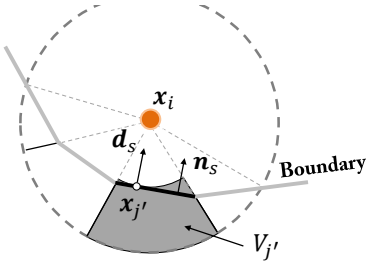


Figure 5: *Illustration of a virtual boundary particle for semi-analytical bulk energy. For a fluid particle* $\mathbf{x}_i$*, each boundary element s intersecting its support contributes a virtual boundary particle* $\mathbf{x}_{j'}$ *at the closest point on s. The shaded arc-shaped sector approximates its effective volume* $V_{j'}$*, which determines its mass* $m_{j'}$*.*

To facilitate semi-analytical boundary integral, two different types of virtual particles are introduced: ***virtual boundary particles*** and ***virtual contact particles***. Virtual boundary particles are created to correct the density approximation for fluid particles near the boundary, while virtual contact particles are introduced to prevent particle penetration into solid boundaries.

4.1. Semi-Analytical Bulk Energy

The bulk energy for particle i is expressed as

$$B(\lambda_i) = b(\lambda_i)V_i, \tag{10}$$

where V_i is the particle volume and $b(\cdot)$ is typically selected as a polynomial function [HLG*25]. According to a standard particle approximation, λ_i is formulated as

$$\lambda_i = \frac{\sum_j m_j W_{ij}}{\rho_0}. \tag{11}$$

The issue arises when particle i is positioned near the boundary, Eq. 11 underestimates the density due to the truncation of the support domain by the boundary. To address this problem, we propose to generate one virtual boundary particle for each boundary element that intersects the support domain of particle i. More specifically, for a triangle s, we define a corresponding virtual boundary particle j' at $\mathbf{x}_{j'}$, where $\mathbf{x}_{j'}$ denotes the closest point on triangle s to $\mathbf{x}_i$. By applying the semi-analytical boundary formulation [CLH*20] to calculate the volume integral(as the arc-shaped intersection sector shown in Figure 5), the mass of particle j' is approximated as

$$m_{j'} = \rho_0 \tilde{W}\big|_{r=\|\mathbf{x}_{j'}-\mathbf{x}_i\|}^{H} \frac{(\mathbf{n}_s \cdot \mathbf{d}_s)A_s}{A_0}\Omega_0, \tag{12}$$

where $\tilde{W}$ is the radial integral of the kernel satisfying $\tilde{W}(H) = 0$. The direction $\mathbf{d}_s$ is defined as a normalized vector pointing from $\mathbf{x}_{j'}$ to $\mathbf{x}_i$, $\mathbf{n}_s$ is the triangle normal, A_s represents the intersected area between triangle s and the support domain. A_0 represents the intersected area between the plane containing triangle s and the support domain, and Ω_0 is the corresponding solid angle.

Incorporating Eq. 12 into Eq. 11 results in the following corrected density ratio

$$\lambda_i = \frac{\sum_j m_j W_{ij} + \sum_{j'} m_{j'} W_{ij'}}{\rho_0}. \tag{13}$$

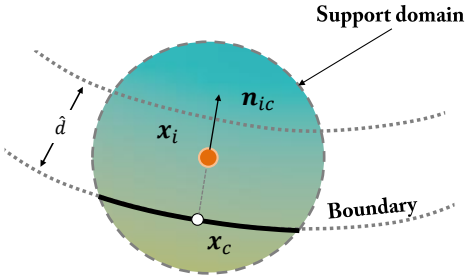


Figure 6: *Illustration of a virtual contact particle for nonlocal contact potential. For particle* $\mathbf{x}_i$*, a virtual contact particle is created at the closest point* $\mathbf{x}_c$ *to* $\mathbf{x}_i$ *on each group of triangles that intersecting the support domain. The vector* $\mathbf{n}_{ic}$ *denotes the normal vector at* $\mathbf{x}_i$ *in the signed distance field reconstructed from the boundary. If more than one group of triangles intersect the support domain of particle i, one virtual contact particle should be created for each group.*

The above correction ensures that bulk energy is compensated near the boundaries. Note the number of virtual boundary particles is equal to the number of triangles that intersect the support domain of particle i. In practical experiments, we observed merely addressing the particle deficiency problem in the calculation of bulk energy is insufficient to prevent particle penetration into the solid (e.g., when the fluid is under strong compression as shown in Figure 2). This observation motivates the introduction of virtual contact particles.

4.2. Nonlocal Contact Potential

Unlike creating a virtual boundary particle for each boundary element that intersects a particle's support domain, we propose creating a single virtual contact particle for each group of triangles that intersects the particle's support domain. Here, a group of triangles refers to all triangles that are interconnected. The position of the virtual contact particle $\mathbf{x}_c$ is defined to be the closest point on the triangle group. However, a notable issue is that multiple points may be equidistant from $\mathbf{x}_i$. To resolve this ambiguity, our practical implementation for calculating $\mathbf{x}_c$ follows two procedures. First, we calculate the signed distance d and the normal vector $\mathbf{n}_{ic}$ at position $\mathbf{x}_i$ with respect to the triangle group. Then, we set $\mathbf{x}_c$ as $\mathbf{x}_c = \mathbf{x}_i - d\mathbf{n}_{ic}$. This strategy enables the identification of a unique position for both concave and convex boundaries.

We then define a nonlocal contact potential for each virtual contact particle as a function of the signed distance from the position of the virtual contact particle, $\mathbf{x}_c$, to the position of the fluid particle $\mathbf{x}_i$ (as shown in Figure 6)

$$C(\gamma) = c(\gamma)V_i. \tag{14}$$

$\gamma = d/\hat{d}$ represents the ratio between the signed distance to a user defined contact threshold, as motivated by Incremental Potential Contact(IPC) [LFS*20a], and $c(\gamma)$ is a barrier function of γ. Note direct application of the logarithmic barrier function is not suitable in this context because the log barrier function only accepts positive values. Therefore, we propose using the following polynomial

barrier function for $c(\gamma)$

$$c(\gamma) = \begin{cases} \sum_{n=1}^{N} \frac{(1-\gamma)^n}{n}, & \gamma < 1, \\ 0, & \gamma \geq 1, \end{cases} \tag{15}$$

where N is the polynomial order. As N increases, note Eq. 15 converges to the logarithmic barrier used in IPC. The advantage of using the polynomial barrier function is that it accepts negative values, which means the computationally expensive exact CCD can be removed during iterations. Furthermore, another advantage of formulating the nonlocal contact potential in terms of γ rather than d is that it can be easily adapted into a GPU-friendly semi-implicit solver, the details of which will be presented in Section 5.

5. Nonlinear Optimization using the Semi-Implicit Successive Substitution Method

To solve the nonlinear optimization problem in Eq. 9, one must deal with the nonquadratic nature of both the bulk energy and the contact potential. Standard solvers for linear systems, such as conjugate gradient, cannot be applied directly, and Newton's method, though theoretically suitable, is impractical in GPU-parallel frameworks because it requires evaluating second-order derivatives and solving global linear systems. To avoid these difficulties, we adopt the *Semi-Implicit Successive Substitution Method* (SISSM) [LHG*23; HLG*25], a Hessian-free approach based on coefficient decomposition.

Taking the first-order derivative of Eq. 9 with respect to $\mathbf{x}_i$ yields

$$\mathbf{x}_i = \mathbf{x}_i^* - \underbrace{\mu \frac{h^2}{m_i} \dot{B}(\lambda_i) \frac{\partial \lambda_i}{\partial \mathbf{x}_i}}_{\mathbf{f}^{\mathcal{B}}(\mathbf{x}_i)} - \underbrace{\kappa \frac{h^2}{m_i} \sum_c \dot{C}(\gamma_{ic}) \frac{\partial \gamma_{ic}}{\partial \mathbf{x}_i}}_{\mathbf{f}^{\mathcal{C}}(\mathbf{x}_i)}, \tag{16}$$

where $\mathbf{x}_i^*$ in the right hand side is the predicted position, the second term $\mathbf{f}^{\mathcal{B}}(\mathbf{x}_i)$ is the force resulting from the bulk energy in Eq. 10, and the third term $\mathbf{f}^{\mathcal{C}}(\mathbf{x}_i)$ is the force resulting from the contact potential in Eq. 14.

The direct update of Eq. 16 does not guarantee convergence because both $\dot{B}(\lambda_i)$ and $\dot{C}(\gamma_c)$ contain nonlinear functions. Following [HLG*25], the key point is to reformulate each term into a nonlocal form as $f(\mathbf{x},\mathbf{x}')(\mathbf{x}'-\mathbf{x})$, and then decompose the coefficient into a positive and a negative part

$$f(\mathbf{x},\mathbf{x}') = f^+(\mathbf{x},\mathbf{x}') + f^-(\mathbf{x},\mathbf{x}'), \tag{17}$$

so that $f^+ \geq 0$ can be handled implicitly while $f^- \leq 0$ handled explicitly.

5.1. Reformulating $\mathbf{f}^{\mathcal{B}}(\mathbf{x}_i)$

Inserting the corrected density ratio of Eq. 13 into $\mathbf{f}^{\mathcal{B}}(\mathbf{x}_i)$ gives

$$\mathbf{f}^{\mathcal{B}}(\mathbf{x}_i) = \mu \frac{h^2}{\rho_0} \dot{B}(\lambda_i) \left(\sum_j \dot{W}_{ij} \frac{\mathbf{x}_j - \mathbf{x}_i}{r_{ij}} + \sum_{j'} \eta_{j'} \dot{W}_{ij'} \frac{\mathbf{x}_{j'} - \mathbf{x}_i}{r_{ij'}} \right), \tag{18}$$

where $\eta_{j'} = m_{j'}/m_i$ is the relative mass of the virtual boundary particle and $\dot{W}_{ij} = \partial W(r_{ij},H)/\partial r_{ij}$. When Desbrun's spiky kernel [DG96] is used, the decomposition of $\mathbf{f}^{\mathcal{B}}(\mathbf{x}_i)$ is formulated as

$$\mathbf{f}^{\mathcal{B}}(\mathbf{x}_i) = A_{ij}^+(\mathbf{x}_j - \mathbf{x}_i) + A_{ij}^-(\mathbf{x}_j - \mathbf{x}_i) + A_{ij'}^+(\mathbf{x}_{j'} - \mathbf{x}_i) + A_{ij'}^-(\mathbf{x}_{j'} - \mathbf{x}_i), \tag{19}$$

where the coefficients A_{ij}^+, A_{ij}^-, $A_{ij'}^+$ and $A_{ij'}^-$ are calculated as

$$\begin{aligned} A_{ij}^+ &= \mu \frac{h^2}{\rho_0} \dot{B}^-(\lambda_i) \frac{\dot{W}_{ij}}{r_{ij}}, & A_{ij}^- &= \mu \frac{h^2}{\rho_0} \dot{B}^+(\lambda_i) \frac{\dot{W}_{ij}}{r_{ij}}, \\ A_{ij'}^+ &= \mu \frac{h^2}{\rho_0} \eta_{j'} \dot{B}^-(\lambda_i) \frac{\dot{W}_{ij'}}{r_{ij'}}, & A_{ij'}^- &= \mu \frac{h^2}{\rho_0} \eta_{j'} \dot{B}^+(\lambda_i) \frac{\dot{W}_{ij'}}{r_{ij'}}. \end{aligned} \tag{20}$$

$\dot{B}^-(\lambda_i)$ and $\dot{B}^+(\lambda_i)$ represent the negative and positive parts, respectively. For the special case of $b(\lambda) = \frac{1}{2}(\lambda-1)^2$, $\dot{B}(\lambda_i)$ splits into $\dot{B}^-(\lambda_i) = -V_i$ and $\dot{B}^+(\lambda_i) = \lambda_i V_i$.

5.2. Reformulating $\mathbf{f}^{\mathcal{C}}(\mathbf{x}_i)$

For the contact force, substituting $\gamma_{ic} = d_{ic}/\hat{d}$ with $d_{ic} = \|\mathbf{x}_c - \mathbf{x}_i\|$ into $\mathbf{f}^{\mathcal{C}}(\mathbf{x}_i)$ yields

$$\mathbf{f}^{\mathcal{C}}(\mathbf{x}_i) = \kappa \frac{h^2}{\rho_0} \sum_c \frac{\dot{C}(\gamma_{ic})}{\hat{d}\, d_{ic}} (\mathbf{x}_c - \mathbf{x}_i), \tag{21}$$

To ensure smoothness as $d_{ic} \to 0$ and maintain consistent signs for coefficients, we propose to reformulate Eq. 21 as

$$\mathbf{f}^{\mathcal{C}}(\mathbf{x}_i) = \kappa \frac{h^2}{\rho_0} \sum_c \left(\frac{\dot{C}^-(\gamma_{ic})}{\hat{d}} \frac{\mathbf{x}_c - \mathbf{x}_i}{d_{ic}} + \frac{\dot{C}^+(\gamma_{ic})}{\hat{d}\, d_{ic}} (\mathbf{x}_c - \mathbf{x}_i) \right). \tag{22}$$

Note that the term $(\mathbf{x}_c - \mathbf{x}_i)/d_{ic}$ in the first term on the right-hand side of Eq. 22 can be interpreted as the inward normal vector $\mathbf{n}_{ic}$. To enhance numerical stability by avoiding division by zero when $\mathbf{x}_c$ lies on the boundary, a more robust approach is to compute $\mathbf{n}_{ic}$ directly. In our implementation, we adopt a straightforward method: we identify the triangle containing $\mathbf{x}_c$ and assign its normal vector to $\mathbf{n}_{ic}$. Alternative techniques, such as deriving the normal from a signed distance field, are also viable.

To avoid division by zero for the second term of Eq. 22, we first take the derivative of Eq. 15 as

$$\dot{c}(\gamma) = \begin{cases} \sum_{n=1}^{N} -(1-\gamma)^{n-1}, & \gamma < 1, \\ 0, & \gamma \geq 1, \end{cases} \tag{23}$$

and then take the decomposition as follows

$$\begin{aligned} \dot{c}^+(\gamma) &= \frac{(1-\gamma)^N + (1+\gamma)^N - 2}{2\gamma}, \\ \dot{c}^-(\gamma) &= \frac{(1-\gamma)^N - (1+\gamma)^N}{2\gamma}. \end{aligned} \tag{24}$$

Note γ in the denominator of $\dot{c}^-(\gamma)$ cancels out with the γ in the numerator, thus $\dot{C}^-(\gamma) = c(\gamma)^- V_i$ remains valid for all values of d_c. Similarly, the numerator of $\dot{c}^+(\gamma)$ includes a multiplicative factor of γ^2 for $N > 1$, meaning that the relation $\dot{C}^+(\gamma)/d = c(\gamma)^+ V_i/d$ holds for all values of d.

Summarizing the above equations with a certain kind of simpli-

fication can yield the following equation

$$\mathbf{f}^{\mathcal{C}}(\mathbf{x}_i) = \sum_c \left(A_{ic}^- \mathbf{n}_{ic} + A_{ic}^+ (\mathbf{x}_c - \mathbf{x}_i) \right), \tag{25}$$

where the coefficients are written as

$$\begin{aligned} A_{ic}^- &= \frac{(1-\gamma)^N - (1+\gamma)^N}{2d_{ic}}, \\ A_{ic}^+ &= \frac{(1-\gamma)^N + (1+\gamma)^N - 2}{2d_{ic}^2}. \end{aligned} \tag{26}$$

It can be verified that $A_{ic}^- \leq 0$ and $A_{ic}^+ \geq 0$, which ensures stability and continuity for all $d_{ic} < \hat{d}$.

5.3. Final Update Rule

Combining the decompositions in Eq. 19 and Eq. 25, Eq. 16 becomes

$$\begin{aligned} \mathbf{x}_i = \mathbf{x}_i^* &+ \sum_j (A_{ij}^- + A_{ij}^+)(\mathbf{x}_j - \mathbf{x}_i) \\ &+ \sum_{j'} (A_{ij'}^- + A_{ij'}^+)(\mathbf{x}_{j'} - \mathbf{x}_i) \\ &+ \sum_c (A_{ic}^- \mathbf{n}_{ic} + A_{ic}^+ (\mathbf{x}_c - \mathbf{x}_i)). \end{aligned} \tag{27}$$

Following the SISSM procedure to treat items with positive coefficients implicitly and negative ones explicitly, a semi-implicit update strategy reads as

$$\begin{aligned} \mathbf{x}_i^{k+1} = \frac{1}{A_i^{k+}} \Big[\mathbf{x}_i^* &+ \sum_j \left(A_{ij}^{k-} (\mathbf{x}_j^k - \mathbf{x}_i^k) + A_{ij}^{k+} \mathbf{x}_j^k \right) \\ &+ \sum_{j'} \left(A_{ij'}^{k-} (\mathbf{x}_{j'}^k - \mathbf{x}_i^k) + A_{ij'}^{k+} \mathbf{x}_{j'}^k \right) \\ &+ \sum_c \left(A_{ic}^{k-} \mathbf{n}_{ic}^k + A_{ic}^{k+} \mathbf{x}_c^k \right) \Big], \end{aligned} \tag{28}$$

with diagonal normalization

$$A_i^{k+} = 1 + \sum_j A_{ij}^{k+} + \sum_{j'} A_{ij'}^{k+} + \sum_c A_{ic}^{k+}. \tag{29}$$

Equation 28 represents the final form of the Semi-Implicit Successive Substitution update. It combines fluid incompressibility and boundary handling into a unified iteration, eliminating the need for Hessian computations, relying solely on local kernel evaluations, and being well-suited for GPU acceleration.

6. Convergence Behavior

To ensure and accelerate the convergence of Eq. 28, two key issues must be addressed. First, the initial point $\mathbf{x}^0$ should be carefully selected to prevent penetration from occurring at the time integration stage. Second, the update strategy from one iterate $\mathbf{x}^k$ to the next should effectively incorporate information from previous iterates to ensure a decrease in the total energy, i.e., $E(\mathbf{x}^{k+1}) < E(\mathbf{x}^k)$.

6.1. Initial Guess with Reduced-order CCD (RCCD)

Continuous collision detection (CCD) is widely used to ensure non-intersection between fast-moving objects by detecting the exact moment of collision and applying a response. The Incremental Potential Contact (IPC) method [LFS*20a; LKJ21], for instance, uses CCD to compute the time of impact and applies a log-barrier potential to prevent penetration. Because the log-barrier function grows unbounded as contact is approached, IPC also requires a line-search filter to clamp step sizes. However, computing exact times of impact between vertex–face or edge–edge pairs in 3D is difficult, especially under mixed translation and rotation [WFS*21].

Instead of directly performing an elementary test between a fluid particle and a boundary triangle, we first conduct continuous collision detection (CCD) between the fluid particle and the plane containing the boundary triangle, as demonstrated in Figure 7. If a collision is detected, we then check whether the intersection point lies within the boundary triangle. Our solution simplifies the CCD problem by reducing it from solving roots of a univariate cubic polynomial to solving roots of a univariate quadratic function.

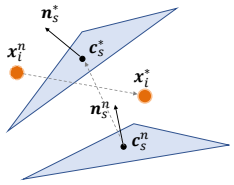


Figure 7: *RCCD*

Assume the plane is defined by the barycenter $\mathbf{c}_s$ of the triangle and its normal vector $\mathbf{n}_s$. The trajectory of the triangle can be identified by the directions formed by $\Delta\mathbf{c}_s^n = \mathbf{c}_s^* - \mathbf{c}_s^n$ and $\Delta\mathbf{n}_s^n = \mathbf{n}_s^* - \mathbf{n}_s^n$. Equivalently, the trajectory of particle i is defined as $\mathbf{x}_i^n + t\Delta\mathbf{x}_i^n$ with $\Delta\mathbf{x}_i^n = \mathbf{x}_i^* - \mathbf{x}_i^n$ representing its search direction. The corresponding CCD is then to find a real solution of $t \in [0,1]$ satisfying the following univariate quadratic function

$$\left[\mathbf{x}_i^n + t\Delta\mathbf{x}_i^n - \left(\mathbf{c}_s^n + t\Delta\mathbf{c}_s^n\right)\right] \cdot \left(\mathbf{n}_s^n + t\Delta\mathbf{n}_s^n\right) = 0. \tag{30}$$

If a real solution for t is found, an additional step is further applied to verify whether the intersection point lies within the triangle; see Algorithm 2 for the complete pseudocode. It is worth noting that finding the roots of the quadratic function in Eq. 30 is significantly simpler than finding roots of a univariate cubic polynomial [TTWM14; WTTM15]. However, the planes defined by the linear trajectories of the barycenter and normal vectors differ slightly from those formed by the triangle's vertices. This minor discrepancy is not problematic, as the nonlocal contact potential permits zero and negative distances and continues to guide the fluid particle inward in cases of interpenetration.

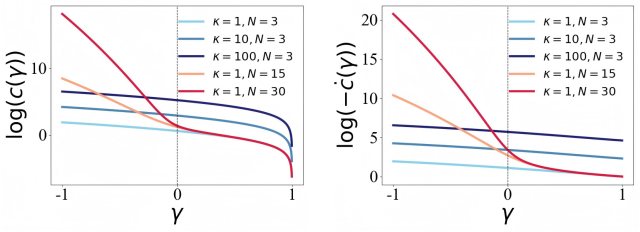
6.2. Parameter Learning for κ and N

According to He et al. [HLG*25], the parameter μ controls the compressibility of fluids. In our current implementation, μ is typically set as a constant value. The challenge lies in determining appropriate values for κ and N for the nonlocal contact potential to generate sufficient forces that prevent particle penetration. To give more insights, Figure 8 provides an intuitive illustration of the relationship between κ and N. Notably, when $\gamma \leq 0$, increasing the value of N proves more effective in enhancing the contact force. However, when $\gamma > 0$, increasing κ is more effective.

Since increasing the values of κ and N will correspondingly in-

Algorithm 1 A Substitution-Type Algorithm

while $t < t_{stop}$ **do**
  $\mathbf{v}_i^* \leftarrow \mathbf{v}_i^n + \Delta t \cdot f$
  $\mathbf{x}_i^* \leftarrow \mathbf{x}_i^n + \Delta t \cdot \mathbf{v}_i^*$
  Search neighbor particles and triangles
  // CCD Phase
  for all particles i **do**
    for all neighbor triangles j **do**
      $\Delta\mathbf{x}_i^n \leftarrow \mathbf{x}_i^* - \mathbf{x}_i^n$
      Call reduced-order CCD in Algorithm 2
    end for
    Initialize $\mathbf{x}_i^0 \leftarrow \mathbf{x}_i^n + t_{\min}\Delta\mathbf{x}_i^n$
  end for
  // SISSM Iteration Phase
  $k \leftarrow 0$
  while $k < N$ **do**
    for all particles i **do**
      Compute fluid density using Equation 1
      Compute boundary density using Equation 3
      Compute $\mathbf{x}_i^{k+1}$ using Equation 28
      Compute α_i^k using Equation 35
      $\mathbf{x}_i^{k+1} \leftarrow \mathbf{x}_i^k + \alpha_i^k\left(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\right)$
      $k \leftarrow k+1$
    end for
  end while
  for all particles i **do**
    $\mathbf{x}_i^{n+1} \leftarrow \mathbf{x}_i^N$
    $\mathbf{v}_i^{n+1} \leftarrow \mathbf{v}_i^* + \frac{\mathbf{x}_i^{n+1} - \mathbf{x}_i^n}{\Delta t}$
  end for
end while



(a) *Polynomial barrier* $c(\gamma)$ **(b)** *Negative derivative* $-\dot{c}(\gamma)$

Figure 8: *Effect of stiffness* κ *and polynomial order* N*. When* $\gamma < 0$*, the order* N *steepens the function. When* $\gamma > 0$*, the stiffness* κ *dominates.*

crease the stiffness of the optimization problem and slow the convergence, our objective is to choose sufficiently large values for both κ and N, but not too large. Staring with an initial value of $\kappa_i = \mu_i$ and $N_i^0 = 3$, our solution to update N for each particle is as follows

$$N_i^k = \begin{cases} 2N_i^{k-1}, & \mathbf{x}_i^{k-1} \in \mathcal{B} \wedge \mathbf{x}_i^k \in \mathcal{B}, \\ N_i^{k-1}, & \text{otherwise}, \end{cases} \tag{31}$$

where $\mathbf{x}_i \in \mathcal{B}$ indicates particle i is located inside the boundary. In other words, when particle i penetrates into the boundary, we first choose to increase the value of N to accelerate the convergence.

Then, as the position of particle i is adjusted back to a non-

Algorithm 2 Reduced-order CCD

function REDUCEDORDERCCD($\mathbf{x}_i^n, \Delta\mathbf{x}_i^n, \mathbf{c}_s^n, \Delta\mathbf{c}_s^n, \mathbf{n}_s^n, \Delta\mathbf{n}_s^n$)
  $a \leftarrow (\Delta\mathbf{x}_i^n - \Delta\mathbf{c}_s^n) \cdot \Delta\mathbf{n}_s^n$
  $b \leftarrow (\mathbf{x}_i^n - \mathbf{c}_s^n) \cdot \Delta\mathbf{n}_s^n + (\Delta\mathbf{x}_i^n - \Delta\mathbf{c}_s^n) \cdot \mathbf{n}_s^n$
  $c \leftarrow (\mathbf{x}_i^n - \mathbf{c}_s^n) \cdot \mathbf{n}_s^n$
  if $\|a\| < \epsilon$ **then**
    if $\|b\| < \epsilon$ **then**
      return 1
    else
      $t \leftarrow \max(0, \min(-\frac{c}{b}, 1))$
    end if
  else
    Compute discriminant: $\Delta \leftarrow b^2 - 4ac$
    if $\Delta < 0$ **then**
      return 1
    else
      $t_0 \leftarrow \frac{-b-\sqrt{\Delta}}{2a}$
      $t_1 \leftarrow \frac{-b+\sqrt{\Delta}}{2a}$
      $t \leftarrow \max(0, \min((\min(t_0, t_1)), 1))$
    end if
  end if
  if $\mathbf{x}_i^n + t\Delta\mathbf{x}_i^n$ is inside triangle s at time t **then**
    return t
  else
    return 1
  end if
end function

penetrating state, we estimate a lower bound for κ such that the contact force is sufficiently strong relative to the pressure force, ensuring that the predicted position of $\mathbf{x}_i$ remains $\hat{d}$ away from the boundary.

$$\kappa_i = \max\left(\frac{\left\|\mathbf{f}^{\mathcal{B}}\left(\mathbf{x}_i^k\right)\right\| + \max_c\left(\hat{d} - d_{ic}^k\right)}{\left\|\frac{\Delta t^2}{m_i}\sum_c \dot{C}(\gamma_{ic}^k)\frac{\partial\gamma_{ic}^k}{\partial\mathbf{x}_i}\right\|}, \kappa_0\right), \tag{32}$$

where κ_0 represents a constant value to ensure convergence.

From the above discussion, it can be noted that all virtual contact particles interacting with the same fluid particle share the same kernel function. Therefore, we only need to store the values of κ and N on each particle.

6.3. Line Search

By initializing $\mathbf{x}_i^0 = \mathbf{x}_i^n + t\Delta\mathbf{x}_i^n$, a sequence of iterates $\mathbf{x}_i^k$ can be generated using Eq. 28. To ensure that each iteration decreases the total energy, we apply a line search filter of the form

$$\mathbf{x}_i \leftarrow \mathbf{x}_i^k + \alpha_i^k\left(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\right), \tag{33}$$

where $\alpha_i^k \in [0,1]$ denotes the step length.

Computing the exact optimal step size by backtracking or Wolfe conditions [NW99] is often expensive, and in practice provides only limited improvement since subsequent iterations further refine the solution. Instead, we adopt an inexpensive approximation

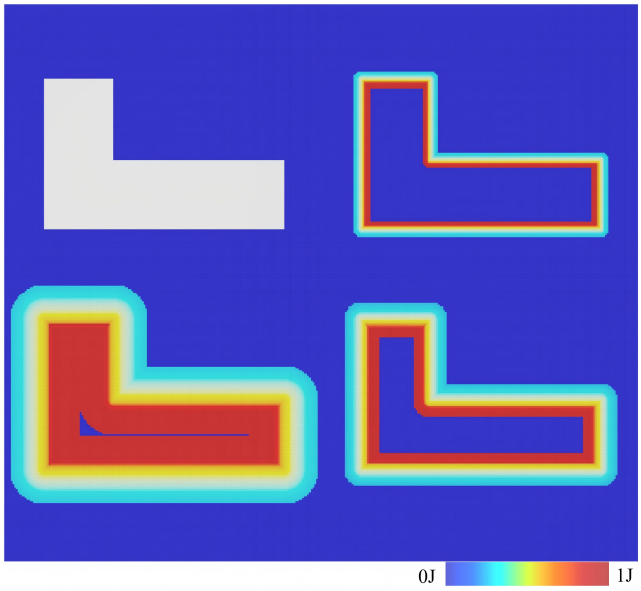


Figure 9: *Effect of the thickness $\hat{d}$ on contact band and feature capture with contact potential color coded from blue(0J) to red(1J) . Small $\hat{d}$ confines contact to a thin band, yielding steeper repulsive forces and preserving sharp corners and narrow gaps; large $\hat{d}$ spreads contact over a thicker band, softening forces and blurring small geometric features. If no boundary triangles are detected within the support (i.e., the sample is farther than $\hat{d}$ from the surface), the contact term is inactive.*

by linearizing the energy function around $\mathbf{x}_i^k$ using a first-order Taylor expansion:

$$\psi_i(\alpha) \approx \psi_i^k + \alpha \nabla \psi_i^k \cdot \left(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\right). \tag{34}$$

Following the Newton step intuition, we define α_i^k as the intersection point of this tangent with the horizontal axis:

$$\alpha_i^k = \min\left[-\frac{\psi_i^k}{\nabla\psi_i^k \cdot (\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) + \delta},\ 1\right], \tag{35}$$

where δ is a small positive constant to avoid division by zero.

Two cases arise in practice. If both $\mathbf{x}_i^k$ and $\mathbf{x}_i^{k+1}$ lie on the same side of the local minimum of ψ, then α_i^k is set to 1, meaning that the full update $\mathbf{x}_i^{k+1}$ is already a valid descent step. If $\mathbf{x}_i^{k+1}$ lies on the opposite side, the formula in Eq. 35 clamps α_i^k into $[0,1]$, avoiding overshooting while still ensuring a monotone energy decrease.

To measure convergence, we follow [HLG*25] and define the relative energy error rate as

$$\varepsilon^k = \frac{\psi(\mathbf{x}^k) - \psi(\mathbf{x}^N)}{\psi(\mathbf{x}^0) - \psi(\mathbf{x}^N)}, \tag{36}$$

where $\psi(\mathbf{x}^k)$ is the total energy at iteration k, and $\psi(\mathbf{x}^N)$ is the energy at the final iterate. This normalized measure quantifies how fast the iterative process reduces the energy relative to the initial gap, and provides a consistent criterion for evaluating convergence behavior.

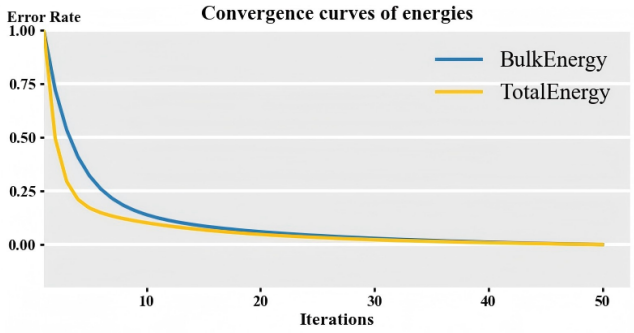


Figure 10: *Energy error rates over the first 50 iterations.*

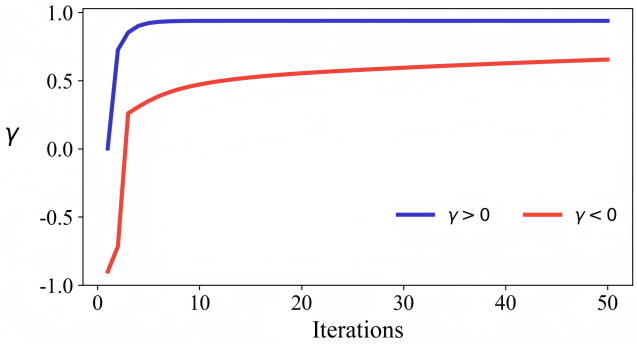


Figure 11: *Particle trajectory. The initial positions of the particles are at $\gamma = 0.001$ and $\gamma = -0.95$. The position changes of the particles over 50 iterations when using our method, where $\hat{d} = R, \kappa_0 = 1$, $N^0 = 3$.*

The effect of this line search strategy is illustrated in Figure 10, which plots the relative error rates of the bulk energy and the total energy over 50 iterations. Both curves exhibit a rapid decrease in the first few iterations, followed by a smooth asymptotic decay toward zero. The total energy consistently decreases without oscillation, confirming that the proposed line search filter successfully prevents overshooting and enforces monotone convergence. Meanwhile, the bulk energy error rate follows a nearly identical trend, showing that the stabilization achieved for the total energy naturally extends to the individual energy components. These results demonstrate that our iterative solver, combined with the inexpensive line search filter, achieves fast and stable convergence without the need for costly backtracking or exact Wolfe condition checks.

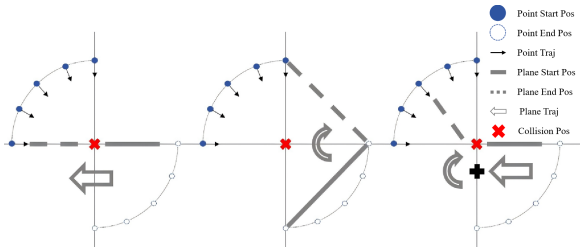
7. Experiments and Evaluation

To validate the effectiveness and robustness of our method, we conduct a series of carefully designed experiments. Our implementation follows Algorithm 1, and the code is developed in C++ and parallelized on the GPU using CUDA. All experiments are performed on a laptop computer equipped with an NVIDIA GeForce RTX 5080 GPU with 16 GiB of VRAM, an AMD Ryzen 9-7945HX processor (16-core, 2.5 GHz), and 32 GiB of RAM.

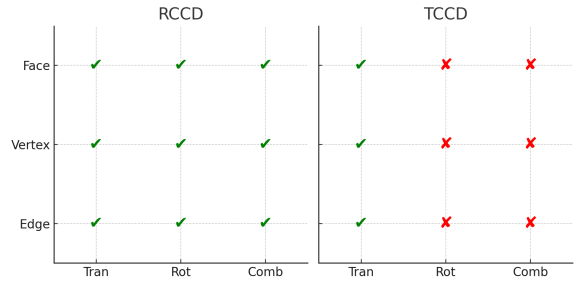
7.1. Ablation Study

7.1.1. CCD Test

To validate the reduced-order CCD introduced in Section 6.1, we designed point–plane collision scenarios as shown in Figure 12a.



(a) *2D schematic of point–plane collisions under translation, rotation, and combined motion.*



(b) *Side by side validation comparison between RCCD and TCCD.*

Figure 12: *RCCD vs. TightCCD comparison. RCCD achieves robust detection under rotational motion, where TightCCD fails due to linear trajectory assumptions, confirming the advantage predicted in Section 6.1.*

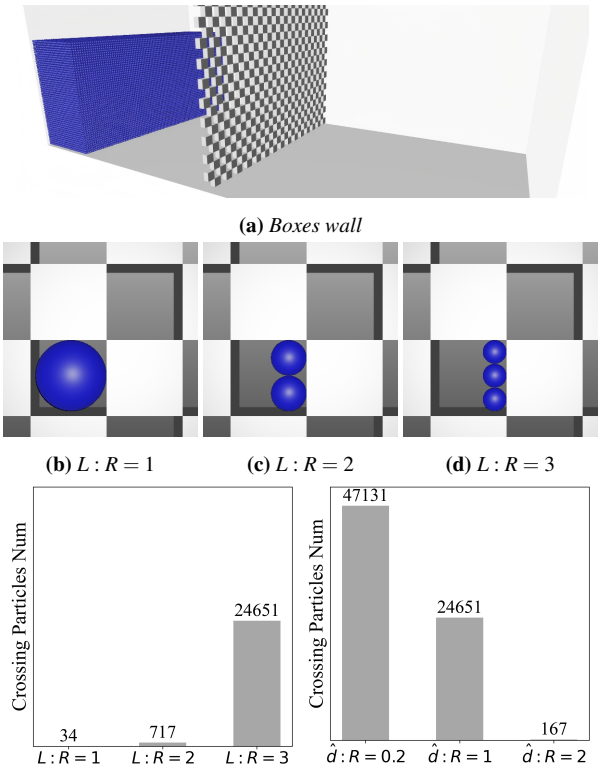
Method	Runtime (μs)	Mean TOI deviation	Miss rate (%)
RCCD	4.51	7.86×10^{-9}	0.00
TightCCD	2.47	3.70×10^{-8}	66.66
FullCCD	117.74	2.85×10^{-8}	0.00

Table 1: *Quantitative comparison among RCCD, TightCCD, and FullCCD.*

The plane moves under pure translation, pure rotation, and combined translation–rotation, with collisions expected at $t = 0.5$. Figure 12b shows side-by-side validation results for vertex, edge, and face cases. TightCCD [WTTM15] formulates collision events under an implicit assumption of linear vertex trajectories, which can be less robust under rotational motion. In contrast, RCCD directly accounts for the rotation-induced nonlinearities while avoiding the computational overhead of fully exact CCD. Table 1 summarizes the efficiency–accuracy tradeoff: TightCCD is the fastest (2.47 μs) but shows a higher miss rate (66.66%), whereas RCCD remains robust without misses at comparable runtime (4.51 μs); FullCCD (rigid-body CCD with rotation [RKC02]) also achieves no misses but with higher runtime (117.74 μs).

7.1.2. Boundary Handling

To evaluate the convergence behavior of our boundary handling formulation, we track the particle trajectory in terms of the normalized signed distance $\gamma = d/\hat{d}$ over successive iterations (Figure 11). Two representative initial conditions are considered: a particle starting just outside the boundary at $\gamma = 0.001$ and another starting deep inside at $\gamma = -0.95$. For both cases, our solver consistently drives the particle toward a stable equilibrium near the contact threshold $\gamma \approx 1$. When $\gamma > 0$, the nonlocal contact potential



(a) *Boxes wall*

(b) $L : R = 1$ **(c)** $L : R = 2$ **(d)** $L : R = 3$

(e) *Effect of* $L : R$ **(f)** *Effect of* $\hat{d} : R$

Figure 13: *Boundary feature capture depends jointly on particle size R, feature size L, and thickness $\hat{d}$. Results validate the theoretical trade-offs discussed in Section 7.1.3.*

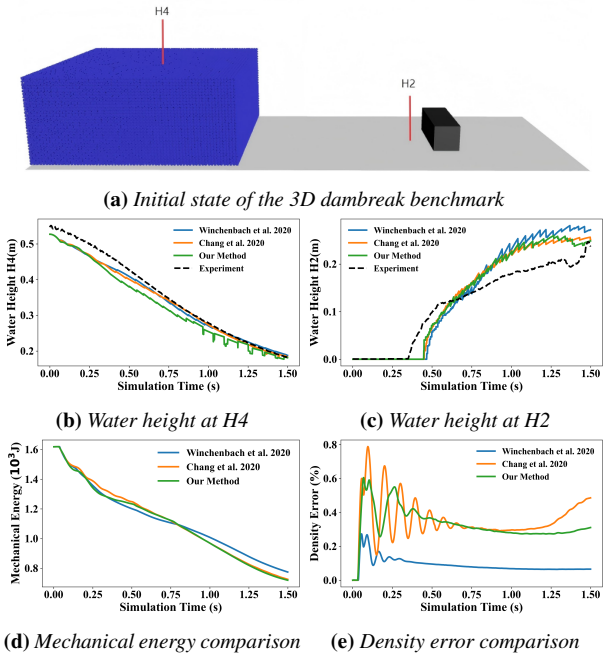
quickly damps the displacement, and the particle converges monotonically to the boundary without overshoot. When $\gamma < 0$, the particle is initially in penetration, but the adaptive update of N and κ steepens the repulsive response and gradually pushes the particle outward. This results in a smooth trajectory that asymptotically approaches the boundary from inside. The experiment demonstrates that our method can robustly handle both penetration and non-penetration cases, ensuring convergence without oscillations or divergence even under challenging initial conditions.

7.1.3. Boundary Feature Capturing

Finally, we evaluate the ability to capture geometric features, which depends on the particle size R, the thickness $\hat{d}$, and the minimum boundary feature size L. Figure 9 illustrates that larger $\hat{d}$ smooths the boundary response, while smaller $\hat{d}$ preserves fine-scale features. Quantitative tests using a wall composed of boxes (Figure 13a) show that particles resolve boundary gaps only when $L \geq 3R$ (Figure 13e). Varying $\hat{d}$ at fixed $L : R = 3$ (Figure 13f) further confirms that smaller thresholds enhance feature capture, consistent with Figure 9.

7.2. Comparison to Other Methods

We compare our method with the semi-analytical approaches of Winchenbach et al. [WAK20] and Chang et al. [CLH*20]. For fair-



(a) *Initial state of the 3D dambreak benchmark*

(b) *Water height at H4* **(c)** *Water height at H2*

(d) *Mechanical energy comparison* **(e)** *Density error comparison*

Figure 14: *3D dambreak benchmark [KFV*05]. All three semi-analytical boundary methods reproduce the experimental measurements with comparable accuracy. Mechanical energy curves are nearly identical. For density error $(\rho_{avg} - \rho_0)/\rho_0$, all methods show fluctuations at the beginning, but our method and Winchenbach et al. [WAK20] achieve stable state faster.*

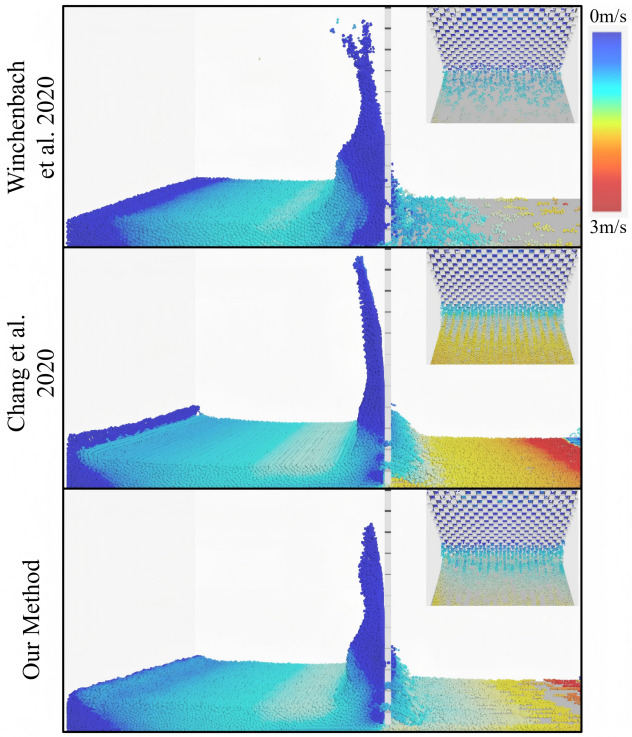


Figure 15: *A wall composed of boxes with scale ratio $L : R = 3$. All methods are able to preserve the boundary geometry and allow particles to pass smoothly through the narrow gaps.*

ness, we integrate their methods in PBF [MM13], using the same XSPH viscosity [SB12] and a timestep of $\Delta t = 1$ms.

7.2.1. Fixed Boundary

Benchmark The dambreak test (Figure 14a) evaluates global incompressibility under fixed boundaries. At H4 (Figure 14b), Winchenbach et al. [WAK20] matches experimental water heights most closely, while Chang et al. [CLH*20] and our method follow similar but slightly lower trajectories. At H2 (Figure 14c), all methods show a small delay relative to experiments yet remain consistent with one another. The three methods yield nearly overlapping mechanical energy curves (Figure 14d), confirming that fixed-boundary scenarios are well handled by all. The main difference appears in density error (Figure 14e): our method, like Chang et al. [CLH*20], exhibits fluctuations, but with smaller amplitude, while Winchenbach et al. [WAK20] provides superior volume preservation.

A wall of boxes To further assess complex fixed boundaries, we apply the test to a wall of boxes (Figure 13a). As shown in Figure 15, all three approaches succeed in maintaining the detailed geometry of the gaps and producing the expected particle flow-through behavior. This demonstrates that for static boundaries with moderate complexity, existing semi-analytical formulations as well as our method can achieve comparable results.

7.2.2. Moving Boundary

Rotating cone In the rotating cone experiment (Figure 16), Winchenbach et al. [WAK20] quickly fails after fluid injection, with particles escaping uncontrollably and the simulation breaking down. Chang et al. [CLH*20] performs more robustly but still exhibits noticeable penetrations and instabilities during rotation. In contrast, our method maintains stability throughout both the injection and the high-speed rotation phases. This demonstrates that our method can robustly handle narrow and moving boundary geometries while avoiding failure modes observed in prior semi-analytical methods.

Rotor pump The rotor pump in Figure 17 represents an extreme moving-boundary test. Winchenbach et al. [WAK20] becomes unstable in narrow gaps, and Chang et al. [CLH*20] exhibits severe penetration. Our method maintains stable performance, echoing the theoretical expectation of robustness in highly compressed boundary regions.

7.3. More Examples

Jeep wading Figure 18 shows a jeep traversing water at time steps of Δt = 3 ms, 6 ms, and 9 ms. Chang et al. [CLH*20] quickly loses splash intensity and suffers penetration, revealing sensitivity to time step size. Our method maintains sharp splashes and avoids penetration into the wheels even at $\Delta t = 9$ ms, reflecting the stability of our variational solver under large steps.

Hydroturbine We used three components, namely pipes, guide plates, and rotors (angular velocity 87.2 rad/s), to form a hydroturbine scene in Figure 1. The initial velocity of the water flow is 10 m/s, and the behavior of the water flow in the intricate boundary is stable throughout the simulation.

Scene	Figure	N_p	N_t	Δt	Iters	T_{tri}	T_{par}	T_{ccd}	T_{siss}	T_{vis}	T_{total}
Benchmark	Figures 14a	103K	24	1ms	5	2.224ms	5.973ms	0.082ms	6.904ms	1.490ms	16.681ms
Boxes wall	Figures 15(Bottom)	140K	5.4k	1ms	5	5.588ms	8.74ms	0.662ms	18.178ms	0.769ms	29.346ms
Rotating cone	Figures 16	25K	6	1ms	5	1.445ms	9.506ms	0.026ms	2.663ms	0.677ms	13.6348ms
Rotor pump	Figures 17c	1.1M	18k	1ms	5	6.128ms	21.814ms	8.614ms	22.132ms	2.069ms	72.191ms
Jeep wading	Figures 18b(Left)	1.59M	187k	3ms	10	4.699ms	9.787ms	6.334ms	80.463ms	1.670ms	113.930ms
Jeep wading	Figures 18b(Middle)	1.59M	187k	6ms	10	5.847ms	11.019ms	7.092ms	100.996ms	4.721ms	155.041ms
Jeep wading	Figures 18b(Right)	1.59M	187k	9ms	10	5.836ms	12.350ms	10.292ms	150.852ms	6.638ms	202.740ms
Hydroturbine	Figures 1	2.1M	49k	1ms	5	17.376ms	30.662ms	37.710ms	89.868ms	3.584ms	205.173ms
Pump render	Figures 2	1.1M	18k	1ms	5	9.221ms	42.112ms	16.736ms	68.253ms	2.050ms	121.763ms
Gear	Figures 3	484K	31k	1ms	5	8.038ms	24.699ms	8.844ms	71.631ms	4.797ms	134.598ms

Table 2: *In these scenes, the stiffness parameter μ of bulk energy is set to 1 and contact potential parameters $\kappa = \mu$, $N^0 = 3$, the smoothing length H is set to 2.0 times of the particle size R and the thickness $\hat{d}$ equals R all the time; N_p represents the number of particles; N_t represents the number of triangles; The number of iterations for every frame is set to a fixed value, i.e. Iters; T_{tri} is the computational cost to find triangle neighbors; T_{par} is the computational cost to find particle neighbors; T_{siss} is the computational cost for all SISSM iterations; T_{vis} is the computational cost for viscosity phase; T_{total} represents the average computational cost per time step.*

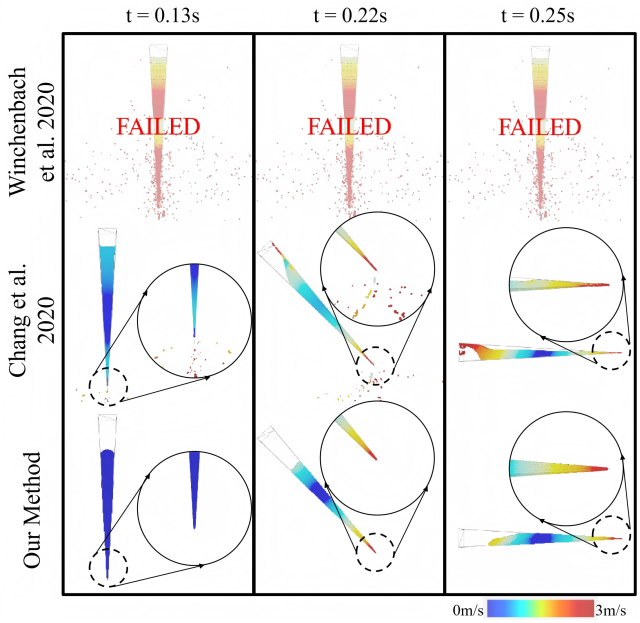
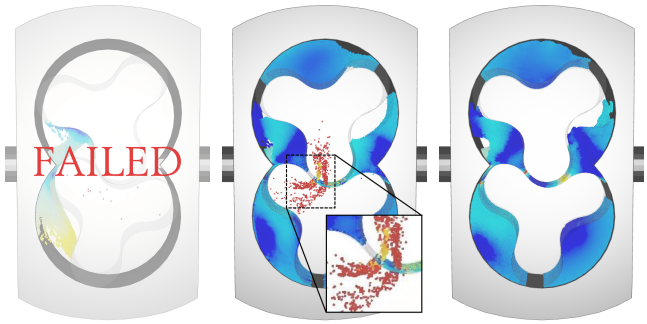


Figure 16: *Rotating cone test at angular velocity 87.2 rad/s. Winchenbach et al. [WAK20] fails shortly after injection (showing severe particle loss and instability). Chang et al. [CLH*20] remains functional but exhibits penetrations and distortions. Our method preserves stability across both injection and rotation, illustrating the robustness of our method for narrow and moving boundaries.*

Pump render In Figure 2, both the initial water velocity and the angular velocity of the rotor are set to twice the values shown in Figure 17. The water flow still remains stable at the cavity squeezed by rotors, and no fluid particles interpenetrate into the moving boundary.

Gear In Figure 3, the system contains a large number of sharp and narrow boundaries in motion (the angular velocities of the front, rear, and upper gears are 34.9 rad/s, and the angular velocity of the left gear is 17.4 rad/s). Despite this geometric complexity, the fluid flow does not exhibit any abnormal behavior.



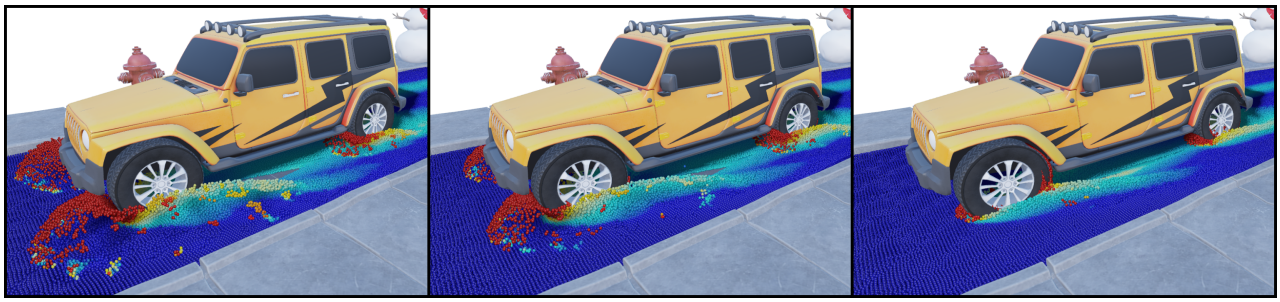
(a) *Winchenbach et al.* **(b)** *Chang et al.* **(c)** *Our method*

Figure 17: *Rotor pump with angular velocity 26.1 rad/s and particle velocities are color-coded. Winchenbach et al. [WAK20] fails in fast moving boundaries, Chang et al. [CLH*20] allows penetration, while our method remains stable. This validates the theoretical advantage of combining SABE and NCP for extreme moving and compression conditions.*

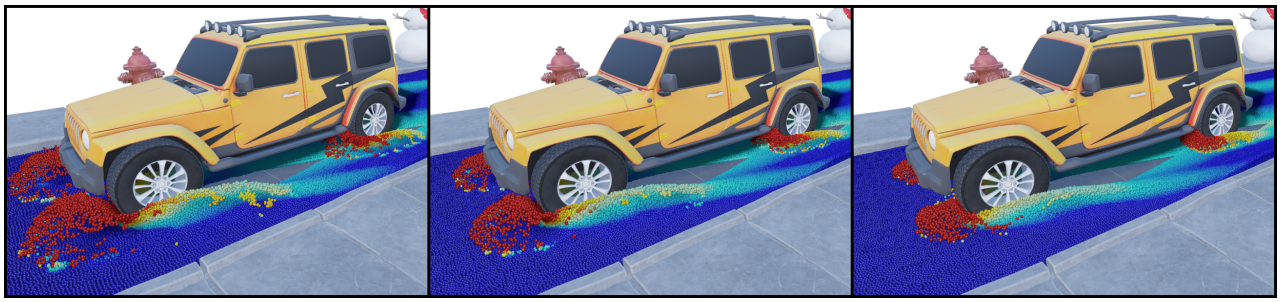
8. Conclusion and Limitations

We introduced a semi-analytical energy model for boundary handling within particle-based fluid simulation. The approach combines a semi-analytical bulk energy model with a pairwise nonlocal contact potential, ensuring consistent density correction near solid boundaries and preventing particle penetration. A Hessian-free, GPU-friendly optimization scheme together with a reduced-order CCD improves efficiency and stability under large time steps. By regenerating virtual boundary particles and contact particles at every iteration and adaptively adjusting stiffness parameters, the method achieves robustness across a wide range of boundary configurations. In particular, the ability to handle multi-boundary interactions allows particles to respond to several boundary features simultaneously, which enhances adaptability and enables accurate restoration of sharp boundary geometries.

Some limitations remain. First, although reduced-order CCD avoids the overhead of full geometric tests, it provides only an approximate collision condition. In highly dynamic scenes or under very large time steps, this approximation may reduce accuracy. Sec-



(a) *Chang et al. [CLH*20] with time steps increasing from left to right: Δt = 3 ms, 6 ms, 9 ms.*



(b) *Our method with time steps increasing from left to right: Δt = 3 ms, 6 ms, 9 ms.*

Figure 18: *Comparative evaluation of a jeep driving through water simulated using three time step sizes (Δt = 3ms, 6ms, and 9ms) with 1.59M particles. Particle velocities are color-coded. (a) The Chang et al. [CLH*20] method maintains splash features only at small time steps, but at larger steps, it suffers from noticeable penetration, indicating reduced stability. (b) Our method preserves splash shape and penetration-free across all time steps, demonstrating robustness to large time steps.*

ond, it is recommended to use a triangular mesh with an element size larger than that of the support domain. Otherwise, calculating the signed distance from fluid particles to the mesh boundary may become sensitive to particle positions, leading to fluctuations near the boundary. Finally, the current framework focuses on one-way boundary treatment, without modeling feedback forces on moving or deformable solids. Extending the approach to handle more general boundary motion remains a direction for future work.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback and suggestions that helped us improve the clarity, presentation, and experimental protocol of the paper. The project was partially supported by the National Natural Science Foundation of China (No. 92570206, No.62302490), the Key Project of the ISCAS (No. ISCAS-ZD-202401), and the Basic Research Project of ISCAS (No. ISCAS-JCMS-202403).

References

[AC10] AIDUN, CYRUS K. and CLAUSEN, JONATHAN R. "Lattice-Boltzmann Method for Complex Flows". *Annual Review of Fluid Mechanics* 42 (2010), 439–472. DOI: 10.1146/annurev-fluid-121108-145519 3.

[AHA12] ADAMI, S., HU, X. Y., and ADAMS, N. A. "A Generalized Wall Boundary Condition for Smoothed Particle Hydrodynamics". *J. Comput. Phys.* 231.21 (2012), 7057–7075. DOI: 10.1016/j.jcp.2012.05.005 3.

[AIA*12] AKINCI, NADIR, IHMSEN, MARKUS, AKINCI, GIZEM, et al. "Versatile Rigid-Fluid Coupling for Incompressible SPH". *ACM Trans. Graph.* 31.4 (2012), 1–8. DOI: 10.1145/2185520.2185558 3.

[BEB12] BROCHU, TYSON, EDWARDS, ESSEX, and BRIDSON, ROBERT. "Efficient Geometrically Exact Continuous Collision Detection". *ACM Transactions on Graphics* 31.4 (2012). DOI: 10.1145/2185520.2185592 3.

[BGI*18] BAND, STEFAN, GISSLER, CHRISTOPH, IHMSEN, MARKUS, et al. "Pressure Boundaries for Implicit Incompressible SPH". *ACM Trans. Graph.* 37.2 (2018), 1–11. DOI: 10.1145/3180486 2, 3.

[BGPT18] BAND, SAGI, GISSLER, CHRISTOPH, PEER, ANDREAS, and TESCHNER, MATTHIAS. "MLS Pressure Boundaries for Divergence-Free and Viscous SPH Fluids". *Computers & Graphics* 74 (2018), 103–113. DOI: 10.1016/j.cag.2018.08.001 3.

[BK15] BENDER, JAN and KOSCHIER, DAN. "Divergence-Free Smoothed Particle Hydrodynamics". *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2015, 147–155. DOI: 10.1145/2786784.2786796 2.

[BKCW14] BENDER, JAN, KOSCHIER, DAN, CHARRIER, PATRICK, and WEBER, DANIEL. "Position-based simulation of continuous materials". *Computers & Graphics* 44 (2014), 1–10 2.

[BKWK19] BENDER, JAN, KUGELSTADT, TASSILO, WEILER, MARCEL, and KOSCHIER, DAN. "Volume maps: An implicit boundary representation for SPH". *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games*. 2019, 1–10 3.

[BKWK20] BENDER, JAN, KUGELSTADT, TASSILO, WEILER, MARCEL, and KOSCHIER, DAN. "Implicit frictional boundary handling for SPH". *IEEE Transactions on Visualization and Computer Graphics* 26.10 (2020), 2982–2993. DOI: 10.1109/TVCG.2020.3004245 3.

[BL99] BONET, J. and LOK, T.-S. L. "Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations". *Computer Methods in Applied Mechanics and Engineering* 180.1-2 (1999), 97–115. DOI: 10.1016/S0045-7825(99)00051-1 3.

[BML*14] BOUAZIZ, SOFIEN, MARTIN, SEBASTIAN, LIU, TIANTIAN, et al. "Projective Dynamics: Fusing Constraint Projections for Fast Simulation". *ACM Trans. Graph.* 33.4 (2014), 1–11. DOI: 10.1145/2601097.2601116 4.

[BT07] BECKER, MARKUS and TESCHNER, MATTHIAS. "Weakly Compressible SPH for Free Surface Flows". *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2007, 209–217 2.

[CD98] CHEN, SHIYI and DOOLEN, GARY D. "Lattice Boltzmann Method for Fluid Flows". *Annual Review of Fluid Mechanics* 30 (1998), 329–364. DOI: 10.1146/annurev.fluid.30.1.329 3.

[CLC*20] CHEN, XIAO-SONG, LI, CHEN-FENG, CAO, GENG-CHEN, et al. "A Moving Least Square Reproducing Kernel Particle Method for Unified Multiphase Continuum Simulation". *ACM Trans. Graph.* 39.6 (2020), 1–15. DOI: 10.1145/3414685.3417804 2.

[CLH*20] CHANG, YUE, LIU, SHUSEN, HE, XIAOWEI, et al. "Semi-analytical Solid Boundary Conditions for Free Surface Flows". *Computer Graphics Forum* 39.7 (2020), 131–141. DOI: 10.1111/cgf.14132 2–5, 10–13.

[DG96] DESBRUN, MATHIEU and GASCUEL, MARIE-PAULE. "Smoothed particles: A new paradigm for animating highly deformable bodies". *Computer Animation and Simulation'96: Proceedings of the Eurographics Workshop in Poitiers*. Springer. 1996, 61–76 6.

[FGW*21] FEI, YUN, GUO, QI, WU, RUNDONG, et al. "Revisiting integration in the material point method: a scheme for easier separation and less dissipation". *ACM Transactions on Graphics (TOG)* 40.4 (2021), 1–16 3.

[FLR*13] FERRAND, MARTIN, LAURENCE, DOMINIQUE, ROGERS, BENEDICT D., et al. "Unified Semi-Analytical Wall Boundary Conditions for Inviscid, Laminar or Turbulent Flows in the Meshless SPH Method". *Int. J. Numer. Methods Fluids* 71.3 (2013), 446–472. DOI: 10.1002/fld.3666 3.

[FM03] FRIES, THOMAS-PETER and MATTHIES, HERMANN G. *Classification and Overview of Meshfree Methods*. Technical Report, Department of Mathematics and Computer Science, Technical University of Braunschweig. 2003 2.

[FM15] FUJISAWA, MAKOTO and MIURA, KENJIRO T. "An Efficient Boundary Handling with a Modified Density Calculation for SPH". *Computer Graphics Forum* 34.7 (2015), 155–162. DOI: 10.1111/cgf.12754 2, 3.

[GPB*19] GISSLER, CHRISTOPH, PEER, ANDREAS, BAND, STEFAN, et al. "Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling". *ACM Trans. Graph.* 38.1 (2019), 1–13. DOI: 10.1145/3284980 2, 3.

[HBFR04] HUERTA, ANTONIO, BELYTSCHKO, TED, FERNÁNDEZ-MÉNDEZ, SONIA, and RABCZUK, TIMON. "Meshfree methods". (2004) 3.

[HJL*25] HOLZ, DANIEL, JESKE, STEFAN RHYS, LÖSCHNER, FABIAN, et al. "Multiphysics Simulation Methods in Computer Graphics". *Comput. Graph. Forum* 44.1 (2025), e70082. DOI: 10.1111/cgf.70082 2.

[HLG*25] HE, XIAOWEI, LIU, SHUSEN, GUO, YUZHONG, et al. "A Semi-Implicit SPH Method for Compressible and Incompressible Flows with Improved Convergence". *Comput. Graph. Forum*. Vol. 44. 1. 2025, e70043. DOI: 10.1111/cgf.70043 3–7, 9.

[HLL*12] HE, XIAOWEI, LIU, NING, LI, SHENG, et al. "Local Poisson SPH for Viscous Incompressible Fluids". *Comput. Graph. Forum* 31.6 (2012), 1948–1958. DOI: 10.1111/j.1467-8659.2012.03191.x 2.

[IOS*14] IHMSEN, MARKUS, ORTHMANN, JENS, SOLENTHALER, BARBARA, et al. "Implicit Incompressible SPH". *IEEE Trans. Vis. Comput. Graph.* 20.3 (2014), 426–435. DOI: 10.1109/TVCG.2013.105 2.

[KBLP04] KULASEGARAM, SIVAKUMAR, BONET, JAVIER, LEWIS, ROLAND W., and PROFIT, MATTHEW. "A Variational Formulation Based Contact Algorithm for Rigid Boundaries in Two-Dimensional SPH Applications". *Comput. Mech.* 33.4 (2004), 316–325. DOI: 10.1007/s00466-003-0532-y 3.

[KBST22] KOSCHIER, DAN, BENDER, JAN, SOLENTHALER, BARBARA, and TESCHNER, MATTHIAS. "A Survey on SPH Methods in Computer Graphics". *Computer Graphics Forum* 41.2 (2022), 737–760. DOI: 10.1111/cgf.14508 2, 3.

[KFV*05] KLEEFSMAN, K. M. T., FEKKEN, G., VELDMAN, A. E. P., et al. "A Volume-of-Fluid Based Simulation Method for Wave Impact Problems". *J. Comput. Phys.* 206.1 (2005), 363–393. DOI: 10.1016/j.jcp.2004.12.007 11.

[LFS*20a] LI, MINCHEN, FERGUSON, ZACHARY, SCHNEIDER, TESEO, et al. "Incremental Potential Contact". *ACM Trans. Graph.* 39.4 (2020), 1–18. DOI: 10.1145/3386569.3392425 2, 5, 7.

[LFS*20b] LI, MINCHEN, FERGUSON, ZACHARY, SCHNEIDER, TESEO, et al. "Incremental potential contact: intersection-and inversion-free, large-deformation dynamics." *ACM Trans. Graph.* 39.4 (2020), 49 3.

[LHG*23] LU, ZIXUAN, HE, XIAOWEI, GUO, YUZHONG, et al. "Projective Peridynamic Modeling of Hyperelastic Membranes With Contact". *IEEE Trans. Vis. Comput. Graph.* 29.9 (2023), 1–14. DOI: 10.1109/TVCG.2023.3271511 2, 6.

[LHG*24] LIU, SHUSEN, HE, XIAOWEI, GUO, YUZHONG, et al. "A Dual-Particle Approach for Incompressible SPH Fluids". *ACM Trans. Graph.* 43.4 (2024), 1–15. DOI: 10.1145/3658132 2.

[LHWW21] LIU, SHUSEN, HE, XIAOWEI, WANG, WENCHENG, and WU, ENHUA. "Adapted SIMPLE Algorithm for Incompressible SPH Fluids with a Broad Range Viscosity". *IEEE Trans. Vis. Comput. Graph.* 28.9 (2021), 3168–3179. DOI: 10.1109/TVCG.2021.3055789 3.

[LKJ21] LI, MINCHEN, KAUFMAN, DANNY M., and JIANG, CHENFANFU. "Codimensional Incremental Potential Contact". *ACM Trans. Graph.* 40.4 (2021), 1–24. DOI: 10.1145/3450626.3459767 7.

[MM13] MACKLIN, MILES and MÜLLER, MATTHIAS. "Position Based Fluids". *ACM Trans. Graph.* 32.4 (2013), 1–12. DOI: 10.1145/2461912.2461984 2, 11.

[MMC16] MACKLIN, MILES, MÜLLER, MATTHIAS, and CHENTANEZ, NUTTAPONG. "XPBD: Position-Based Simulation of Compliant Constrained Dynamics". *Proceedings of the 9th International Conference on Motion in Games*. 2016, 49–54. DOI: 10.1145/2994258.2994272 2.

[Mon94] MONAGHAN, J. J. "Simulating free surface flows with SPH". *J. Comput. Phys.* Vol. 110. 2. 1994. DOI: 10.1006/jcph.1994.1034 2, 3.

[NW99] NOCEDAL, JORGE and WRIGHT, STEPHEN J. *Numerical Optimization*. New York: Springer, 1999. ISBN: 9780387987934 8.

[PT23] PROBST, TIMO and TESCHNER, MATTHIAS. "Monolithic Friction and Contact Handling for Rigid Bodies and Fluids Using SPH". *Computer Graphics Forum*. Vol. 42. 1. 2023, 155–179. DOI: 10.1111/cgf.14727 2.

[RHEW17] REINHARDT, STEFAN, HUBER, MARKUS, EBERHARDT, BERNHARD, and WEISKOPF, DANIEL. "Fully Asynchronous SPH Simulation". *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2017, 1–10. DOI: 10.1145/3099564.3099566 2.

[RKC02] REDON, STÉPHANE, KHEDDAR, ABDERRAHMANE, and COQUILLART, SABINE. "Fast Continuous Collision Detection Between Rigid Bodies". *Computer Graphics Forum* 21.3 (2002), 279–287. DOI: 10.1111/1467-8659.00678 3, 10.

[RL96a] RANDLES, P. W. and LIBERSKY, L. D. "Smoothed Particle Hydrodynamics: Some recent improvements and applications". *Proceedings of the International Conference on Computer Applications in Nuclear Engineering*. La Grange Park, IL: American Nuclear Society, 1996, 395–405 3.

[RL96b] RANDLES, P. W. and LIBERSKY, L. D. "Smoothed Particle Hydrodynamics: Some recent improvements and applications". *Computer Methods in Applied Mechanics and Engineering* 139.1-4 (1996), 375–408. DOI: 10.1016/S0045-7825(96)01090-0 3.

[SB12] SCHECHTER, HAGIT and BRIDSON, ROBERT. "Ghost SPH for Animating Water". *ACM Trans. Graph.* 31.4 (2012), 1–8. DOI: 10.1145/2185520.2185557 3, 11.

[SHM22] SHAO, HAN, HUANG, LIBO, and MICHELS, DOMINIK L. "A fast unsmoothed aggregation algebraic multigrid framework for the large-scale simulation of incompressible flow". *ACM Transactions on Graphics (TOG)* 41.4 (2022), 1–18 3.

[Sil00] SILLING, STEWART A. "Reformulation of Elasticity Theory for Discontinuities and Long-Range Forces". *Journal of the Mechanics and Physics of Solids* 48.1 (2000), 175–209. DOI: 10.1016/S0022-5096(99)00029-0 2.

[SP09] SOLENTHALER, BARBARA and PAJAROLA, RENATO. "Predictive-Corrective Incompressible SPH". *ACM Trans. Graph.* 28.3 (2009), 1–6. DOI: 10.1145/1531326.1531346 2.

[TTWM14] TANG, MIN, TONG, RUOFENG, WANG, ZHENDONG, and MANOCHA, DINESH. "Fast and Exact Continuous Collision Detection with Bernstein Sign Classification". *ACM Trans. Graph.* 33.6 (2014), 1–8. DOI: 10.1145/2661229.2661237 7.

[WAK20] WINCHENBACH, RENE, AKHUNOV, RUSTAM, and KOLB, ANDREAS. "Semi-Analytic Boundary Handling Below Particle Resolution for Smoothed Particle Hydrodynamics". *ACM Trans. Graph.* 39.6 (2020), 1–17. DOI: 10.1145/3414685.3417829 2, 3, 10–12.

[WFS*21] WANG, BOLUN, FERGUSON, ZACHARY, SCHNEIDER, TESEO, et al. "A Large-Scale Benchmark and an Inclusion-Based Algorithm for Continuous Collision Detection". *ACM Trans. Graph.* 40.5 (2021), 1–16. DOI: 10.1145/3460775 7.

[WHK17] WINCHENBACH, RENE, HOCHSTETTER, HENDRIK, and KOLB, ANDREAS. "Infinite Continuous Adaptivity for Incompressible SPH". *ACM Trans. Graph.* 36.4 (2017), 1–10. DOI: 10.1145/3072959.3073713 2.

[WK21] WINCHENBACH, RENE and KOLB, ANDREAS. "Optimized Refinement for Spatially Adaptive SPH". *ACM Trans. Graph.* 40.1 (2021), 1–15. DOI: 10.1145/3424633 2.

[WTTM15] WANG, ZHENDONG, TANG, MIN, TONG, RUOFENG, and MANOCHA, DINESH. "TightCCD: Efficient and Robust Continuous Collision Detection Using Tight Error Bounds". *Comput. Graph. Forum* 34.7 (2015), 289–298. DOI: 10.1111/cgf.12767 3, 7, 10.

[XLYJ23] XIE, TIANYI, LI, MINCHEN, YANG, YIN, and JIANG, CHENFANFU. "A Contact Proxy Splitting Method for Lagrangian Solid-Fluid Coupling". *ACM Trans. Graph.* 42.4 (2023), 1–14. DOI: 10.1145/3592115 2, 3.