

# Solid-Fluid Interaction with Surface-Tension-Dominant Contact

LIANGWANG RUAN\*, CFCS, Peking University & AICFVE, Beijing Film Academy

JINYUAN LIU\*, Dartmouth College

BO ZHU, Dartmouth College

SHINJIRO SUEDA, Texas A&M University

BIN WANG<sup>†</sup>, AICFVE, Beijing Film Academy

BAOQUAN CHEN<sup>†</sup>, CFCS, Peking University & AICFVE, Beijing Film Academy



Fig. 1. Our novel three-way coupling method can simulate the dynamics of surface-tension-dominant contact between solid and liquid, including static contact between a steel paperclip, cherries sitting on top of water, autumn leaves floating and rotating in a brook, and a water strider robot actuated by its joints.

We propose a novel three-way coupling method to model the contact interaction between solid and fluid driven by strong surface tension. At the heart of our physical model is a thin liquid membrane that simultaneously couples to both the liquid volume and the rigid objects, facilitating accurate momentum transfer, collision processing, and surface tension calculation. This model is implemented numerically under a hybrid Eulerian-Lagrangian framework where the membrane is modelled as a simplicial mesh and the liquid volume is simulated on a background Cartesian grid. We devise a monolithic solver to solve the interactions among the three systems of liquid, solid, and membrane. We demonstrate the efficacy of our method through an array of rigid-fluid contact simulations dominated by strong surface tension, which enables the faithful modeling of a host of new surface-tension-dominant phenomena including: objects with higher density than water that remains afloat; ‘Cheerios effect’ where floating objects attract one another; and surface tension weakening effect caused by surface-active constituents.

CCS Concepts: • Computing methodologies → Physical simulation.

Additional Key Words and Phrases: fluid, rigid body, coupling, surface tension

\*Joint first authors

<sup>†</sup>Corresponding authors

Authors’ addresses: Liangwang Ruan, CFCS, Peking University & AICFVE, Beijing Film Academy, ruanliangwang@pku.edu.cn; Jinyuan Liu, Dartmouth College, jinyuan.liu.gr@dartmouth.edu; Bo Zhu, Dartmouth College, bo.zhu@dartmouth.edu; Shinjiro Sueda, Texas A&M University, sueda@tamu.edu; Bin Wang, AICFVE, Beijing Film Academy, binwangbuaa@gmail.com; Baoquan Chen, CFCS, Peking University & AICFVE, Beijing Film Academy, baoquan.chen@gmail.com.

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

© 2021 Association for Computing Machinery.

0730-0301/2021/8-ART120 \$15.00

<https://doi.org/10.1145/3450626.3459862>

## ACM Reference Format:

Liangwang Ruan, Jinyuan Liu, Bo Zhu, Shinjiro Sueda, Bin Wang, and Baoquan Chen. 2021. Solid-Fluid Interaction with Surface-Tension-Dominant Contact. *ACM Trans. Graph.* 40, 4, Article 120 (August 2021), 12 pages. <https://doi.org/10.1145/3450626.3459862>

## 1 INTRODUCTION

Interactions between solids and fluids driven by the strong interfacial capillary forces are seen ubiquitously, especially in a miniature environment. Examples include small insects walking on a pond surface, autumn leaves floating and swirling in a brook stream, breakfast cereal clumping together in a bowl of milk, and many other small-scale interactions between creatures, plants, and their living aquatic environment (see Fig. 1). These phenomena’ most appealing visual aspects are their largely curved and bent liquid surface, the floating and swaying objects, and the intricate and often unstable balance between the object and the liquid surface. Even a small wrinkle on the water surface might cause the floating object to sink. These surface-tension-driven phenomena have been drawing attention from both theoretical and experimental fluid mechanics [Fitzpatrick 2017; Janssens et al. 2017; Navascues 1979; Popinet 2018], motivating new design of various miniature biomimetic robots [Hu et al. 2010; Ozcan et al. 2014; Song and Sitti 2007; Suhr et al. 2005], and inspiring artistic creations of many beautiful photographs and slow-motion videos.

The most salient feature of such a strongly coupled solid-fluid system is the feasibility of holding an object on a liquid surface whose density is significantly higher than the fluid underneath. For example, the density of a steel paperclip can be as high as 8 times as that of water. The underpinning governing physics lies in the additional curvature force that contributes to balancing the body’s gravity in the vertical direction. As illustrated in Figure 2, for a solid object sitting on a surface-tensioned water surface, its force balance

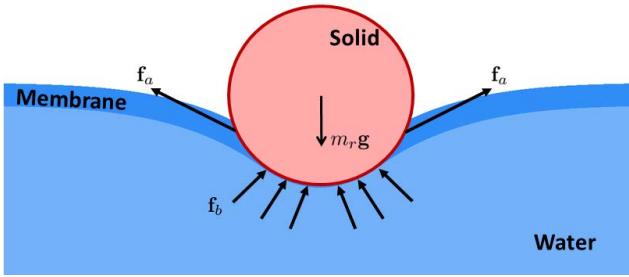


Fig. 2. Solid and fluid interaction. The solid circle stays afloat on the water under the equilibrium among gravity force  $m_r g$ , buoyancy force  $f_b$ , and capillary force  $f_a$ .

can be understood as an equilibrium among three forces:  $m_r g = f_a + f_b$ , with the body's gravity force  $m_r g$ , the water's buoyancy force  $f_b$ , and the water surface's capillary force  $f_a$ . The buoyancy force's effect is deduced by integrating the fluid pressure over the body in contact with water, and the capillary force is calculated by integrating the surface tension along the contact perimeter of the body.

From a computational perspective, accurately modeling the interaction among these three forces requires proper treatment of three subsystems — the liquid body, the solid body, and the strongly tensioned liquid interface between them. However, in both the computational physics and the computer graphics communities, the problem of simulating a strongly coupled surface-tension-dominant contact process remains largely unexplored, due to the lack of effective computational tools to precisely simulate the interactions among the three subsystems. The mainstream numerical paradigms of using an implicit level-set method to model the free-surface flow and its interaction with a Lagrangian solid systems [Batty et al. 2007; Carlson et al. 2004; Robinson-Mosher et al. 2008] suffer from limitations in the following three aspects: First, capturing the liquid-solid contact perimeter is difficult for an implicit geometry representation. Second, using an implicit surface to calculate the curvature as well as the fine-scale capillary wave propagation on the interface is less accurate compared with an explicit mesh representation. Third, this further adds difficulty in building a monolithic system to accurately couple the two systems, especially when the interface dominates the dynamics. In a conventional two-way coupled system, the solid and fluid systems interact with each other via an (implicitly) transferred momentum term between the liquid volume and the solid volume (e.g., see [Robinson-Mosher et al. 2008]). The surface tension force on the interface contributes to the coupling mechanics by enforcing a pressure jump on the liquid's boundary, which (indirectly) affects the solid's dynamics by changing the liquid pressure. There is no direct pathway to bridge the liquid interface and the solid, which makes it infeasible to model the term of  $f_a$  that is essential for a surface-tension-dominant contact process.

To tackle these computational challenges, we propose a novel “three-way” coupling mechanism to model a solid-fluid coupling system driven by strong surface tension. Our key innovation is to treat the surface-tension-dominant interface as a Lagrangian thin

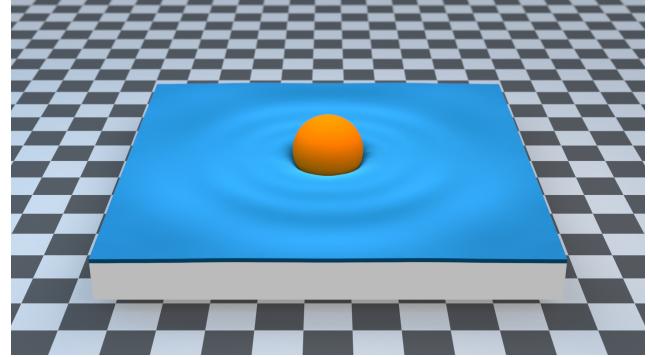


Fig. 3. Sphere Falling into Water. Due to the mesh representation of the thin liquid membrane, we obtain fine-scale wave propagation stimulated by the solid's motion.

membrane that is simultaneously coupled with both the liquid volume and the solid object. Under this new Lagrangian perspective, the interface is no longer an infinitely thin numerical carrier to transfer the surface tension force from the boundary to the interior; instead, it is a “virtual” liquid membrane with a finite, small thickness that can directly impose both buoyancy and surface tension forces onto its contacting objects. In this sense, the two-way coupled system now becomes a three-way coupled one, with an additional thin liquid layer separating the liquid bulk and the solid. Thanks to the Lagrangian nature and its explicit mesh representation of the additional liquid membrane, various physical forces can be effectively discretized and enforced in the coupled evolution of both solid and fluid volume. We develop a full set of numerical infrastructures centered around this “three-way” coupling idea to comprehensively accommodate the treatments of incompressibility, buoyancy, surface tension, rigid articulation, and their various intricate interactions. One significant characteristic of our numerical solution is its ability to handle the coupling between liquid and high-density-ratio solid systems, which was infeasible for all the previous methods. In particular, our simulator can accurately reproduce the surface-tension-dominant floating phenomena of various thin objects, such as paperclip, thin shell boat, pushpin etc., all implemented with their physical densities (with the maximum density ratio up to 8—the density ratio of iron to water).

We summarize our technical contributions as follows:

- A novel Lagrangian thin membrane representation to accurately capture the contacting interactions between solid and fluid driven by strong surface tension.
- A monolithic coupling framework satisfying all velocity and position constraints while conserving momentum.
- A prediction-correction contact handling scheme to accurately process fluid-solid contact with realistic physical parameters.

## 2 RELATED WORKS

*Solid-Fluid Coupling.* State-of-the-art solvers typically use the Eulerian method for fluid and the Lagrangian method for solid, and the coupling between them is usually done by using the solid

velocity as a fluid boundary condition and integrating the pressure on the interface. G'enevaux et al. [2003] use a simple scheme to integrate fluid and solid separately with a coherent-behavior-guaranteed interface force as interaction. Guendelman et al. [2005] follow a similar idea but use the predicted displacement of solid as the velocity boundary constraints in the projection step. These kinds of weak-coupling methods usually lead to limited stability and accuracy, which can be largely solved by monolithic strong coupling methods, i.e., integrating fluid and solid in one system. The first fully implicit stable fluid-rigid two-way coupling solver is presented by Chentanez et al. [2006], who combine the fluid pressure projection equation with velocity update equation of deformable bodies into one asymmetric linear equation. Batty et al. [2007] introduce an SPD (symmetric positive definite) equation to solve the coupling between fluid and rigid body by considering kinetic energy minimization. Robinson-Mosher et al. [2008] further make the coupling system to conserve momentum by using a momentum conservation equation for the fluid-body-mixed cells instead of directly interpolating the pressure on the grid to compute forces and torques on the interface. Robinson-Mosher et al. [2011] further make the equation in [Robinson-Mosher et al. 2008] SPD by using algebraic transformations. Robinson-Mosher et al. [2009] use Lagrangian multipliers method to make the boundary condition be free-slipping, which means only the normal component of the velocity is constrained. Zarifi and Batty [2017] generalize the cut-cell method into the coupling system, managing to achieve free-slipping boundary condition without Lagrangian multipliers by rewriting the incompressible condition in the fluid-body-mixed cells, and also make it SPD. Hyde and Fedkiw [2019] generalize the method in fluid-solid coupling to fluid coupling with sub-grid solids. Aanjaneya [2018] develops an efficient solver to accelerate the fluid-rigid coupling.

*Surface Tension.* Surface tension is important for the simulation of free-surface fluid. It can be represented in different ways depending on the discretization methods. From an Eulerian point of view, surface tension can be described as pressure discontinuity across the interface between different phases. From a Lagrangian point of view, surface tension can be accommodated as forces between surface particles. Surface tension can be treated either explicitly or implicitly. Implicit treatment often provides better numerical stability. Kang et al. [2000] extend the ghost fluid method (GFM) to treat multi-phase incompressible flow including the effects of surface tension. Wang et al. [2005] use surface tension as an explicit boundary condition in the projection step, and use a virtual level set surface penetrating solid to simulate the contact angles of water drops. However, these explicit surface tension treatments will cause instability when surface tension is dominating the scene, such as for the simulation of bubbles. Zheng et al. [2009] propose a semi-implicit surface tension method based on the level set method, by considering the advection of the surface curvature using velocity spatial divergence on the surface, to produce realistic bubble dynamics. Sussman and Ohta [2009] use a notably different scheme by solving a volume-preserving equation based on mean curvature instead of deriving surface tension formulation. Chen et al. [2020] propose an extended cut-cell method for handling liquid structures with surface tension that are smaller than a grid cell. Compared to the grid-based

methods, Lagrangian surface tension exhibits its computational merits for its explicit geometry discretization. There are mainly two categories of Lagrangian methods. Mesh-based methods with connectivity information define the differential operators on a mesh node and its incident triangles. Zhang et al. [2011] simulate droplets on a surface mesh using deformation operators. Batty et al. [2012] model thin viscous sheets with the single-layer mesh with thickness. Da et al. [2015] model complex bubble structures with a Lagrangian non-manifold triangle mesh and integrate the surface tension into vertex-based circulations. Da et al. [2016] develop a surface-only simulation framework that adapts a general 3D fluid solver onto the surface mesh. Ishida et al. [2017] replace the curvature with the gradient of the surface area function, to handle the non-manifold junctions of films and bubbles robustly. Lagrangian representations of surface tension are also easier to be treated implicitly. Schroeder et al. [2012] developed a hybrid framework that formulates implicit surface tension force on a Lagrangian mesh and carries out the pressure solve on background grid. Zheng et al. [2015] introduce an implicit surface tension scheme based on the particle-constructed surface tracking method. Zhu et al. [2015, 2014] propose a unified framework that simulates codimensional phenomena using codimensional simplicial complexes. Surface differential operators can also be approximated by pure particles. The traditional smoothed particle hydrodynamics (SPH) method introduced by [Gingold and Monaghan 1977] uses radial symmetrical smoothing kernels to approximate the differential operators. Other approximation methods include graph Laplacian [Belkin and Niyogi 2008], local triangular mesh [Belkin et al. 2009; Lai et al. 2013], closest point method [Cheung et al. 2015] and moving least squares [Lancaster and Salkauskas 1981; Wang et al. 2020].

### 3 PHYSICS MODEL

In this section, we will first go over the notation convention that we use throughout this paper. Then we will describe the three physical systems: volumetric fluid, surface membrane, and the rigid body.

*Notation convention.* We use plain text to denote scalar quantities (e.g.,  $\rho$ ), and boldface lowercase and uppercase letters to denote vectors (e.g.,  $\mathbf{u}$ ) and matrices (e.g.,  $\mathbf{M}$ ) respectively. Subscripts  $r$  and  $s$  are used to distinguish quantities between the rigid body and the surface membrane, respectively. Quantities in their continuous or discretized form are not explicitly distinguished.

*Volumetric Fluid.* The motion of the fluid is governed by the standard incompressible Navier-Stokes equations:

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}, \quad (1)$$

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

where  $\mathbf{u}$  is fluid's velocity;  $p$  is the pressure;  $\rho$  and  $\mu$  are the fluid's density and dynamic viscosity. We ignore viscosity in our algorithm for simplicity.  $\mathbf{f}$  denotes all body forces applied to the fluid. We enforce the Neumann boundary constraint  $\mathbf{u} \cdot \mathbf{n} = 0$  on wall boundaries and the Dirichlet boundary constraint  $p = 0$  on the free surface. At the regions in contact with rigid bodies, we enforce  $\mathbf{u} = \dot{\mathbf{x}}$ , where  $\dot{\mathbf{x}}$  is the velocity of the contact position on the rigid body.

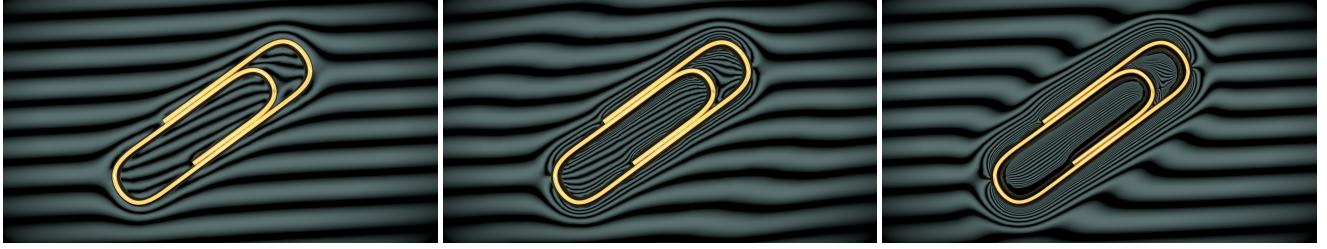


Fig. 4. Paperclips with increasing density stay afloat on the surface of the water. From left to right, the density are 2.0, 5.0, and 7.9 g/cm<sup>3</sup>, respectively. Heavier paperclip deforms the water surface in a larger range with sharper curvature, which can be observed from the degree of distortion of water refraction.

*Surface membrane.* We define the geometry of the surface membrane as the thin region around the fluid's free surface with a constant thickness  $h$ . The governing equations of the fluid membrane are the same as the fluid volume, except that a surface tension force is applied on the membrane due to the unbalanced cohesion force between the fluid molecules near the interface. For a given area surrounded by boundary line  $\Gamma$  on the membrane, its capillary force is calculated as

$$\mathbf{f}_c = \oint_{\Gamma} \sigma \mathbf{dl} \times \mathbf{n}, \quad (3)$$

where  $\sigma$  is the surface tension coefficient, and  $\mathbf{n}$  is the surface normal at  $d\mathbf{l}$  on  $\Gamma$ ; here  $\mathbf{f}_c$  is used to distinguish the capillary force acting on the fluid membrane from the capillary force  $\mathbf{f}_a$  acting on the solid (Eq. 20). From the energy perspective, we can also treat the capillary force as the outcome of surface energy

$$E_c = \sigma A, \quad (4)$$

where  $A$  is the total area of the surface. This energy perspective is what we use in the later analysis. At the same time, the surface membrane exchanges momentum with both the fluid volume underneath and the contacting rigid body.

*Rigid body.* We assume that the (articulated) rigid body is expressed by generalized position,  $\mathbf{q}_r$ , and generalized velocity,  $\mathbf{v}_r$ . For a single rigid body, these generalized coordinates are 6D vectors; for each additional degree of freedom (e.g., joint angles), we add one dimension to these vectors. The equations of motion of an articulated rigid body can be written as:

$$\mathbf{M}_r(\mathbf{q}_r) \frac{d\mathbf{v}_r}{dt} = \mathbf{f}_r(\mathbf{q}_r, \mathbf{v}_r), \quad (5)$$

where  $\mathbf{M}_r$  is the generalized inertia, and  $\mathbf{f}_r$  is the generalized force, which includes gravity, contact forces, Coriolis force, and all other quadratic velocity vectors that may result from the reduced coordinates of the articulated system [Murray et al. 1994; Wang et al. 2019]. We also keep track of the Jacobian matrix,  $\mathbf{J}_r(\mathbf{q}_r)$ , which maps the rigid body's generalized velocity to the 3D world velocity of points that are defined with respect to the rigid body. The transpose of this Jacobian,  $\mathbf{J}_r^T$ , maps the 3D world forces applied at these points to the generalized force of the rigid body. These mappings will be used in later sections to couple the rigid body to the rest of the system.

## 4 DISCRETIZATION

In this section we introduce our discretization for different parts of our system. We use an Eulerian description for the volumetric fluid, a single-layer surface mesh for the surface membrane, and a Lagrangian description for the rigid body.

### 4.1 Fluid

A standard marker-and-cell (MAC) grid is used to discretize the fluid volume by storing the velocity on faces and pressure on cell centers. We take a classical operator-splitting approach to solve the fluid equations [Stam 1999], including semi-Lagrangian advection, applying external forces, and pressure projection. In particular, the projection step calculates the velocity  $\mathbf{u}^{n+1}$  for the next time step based on the intermediate velocity  $\mathbf{u}^*$ :

$$-\mathbf{G}^T \mathbf{u}^{n+1} = 0, \quad (6)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \mathbf{G} \mathbf{p}, \quad (7)$$

where  $\mathbf{G}$  is the discretized gradient operator and  $-\mathbf{G}^T$  is the discretized divergence operator. Substituting  $\mathbf{u}^{n+1}$  from Eq. 7 into Eq. 6 leads to a Poisson equation for pressure:

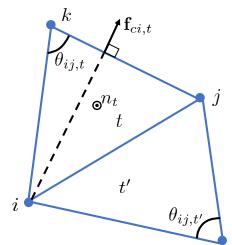
$$\frac{1}{\rho} \mathbf{G}^T \mathbf{G} \hat{\mathbf{p}} = \mathbf{G}^T \mathbf{u}^*, \quad (8)$$

where  $\hat{\mathbf{p}} = \mathbf{p} \cdot \Delta t$  for brevity.

### 4.2 Membrane

We discretize the surface membrane using a triangle mesh  $\mathcal{S}$  in 3D, and a line segment mesh in 2D. We will only focus on the 3D case in this section. The mesh resolution depends on the background grid resolution. In all our experiments, we set the mesh resolution at the initial state to be two times higher than the resolution of the background grid. Other resolutions such as 1.5x and 3x should also work without producing significantly different behaviors.

Each mesh node is interpreted as a particle with mass. The mass of an individual particle is proportional to its surrounding area, and can be calculated as  $m_i = \rho h \sum_{t \in \mathcal{N}_i} A_t / 3$ , where  $\mathcal{N}_i$  is the 1-ring neighborhood of particle  $i$ , and  $A_t$  is the area of its incident triangle  $t$ . With this discretization, we



model the surface tension as capillary forces similarly as [Zheng et al. 2015]:

$$\mathbf{f}_{ci} = \sum_{t \in \mathcal{N}_i} \mathbf{f}_{ci,t} = - \sum_{t \in \mathcal{N}_i} \sigma \frac{\partial \mathbf{A}_t}{\partial \mathbf{x}_i}, \quad (9)$$

where  $\mathbf{f}_{ci,t}$  is the capillary force on particle  $i$  from triangle  $t$ . The exact form of  $\mathbf{f}_{ci,t}$  is

$$\mathbf{f}_{ci,t} = \frac{\sigma}{2} \mathbf{l}_{jk} \times \mathbf{n}_t, \quad (10)$$

where  $\mathbf{l}_{jk} = \mathbf{x}_k - \mathbf{x}_j$ , and  $\mathbf{n}_t$  is the normal of triangle  $t$  pointing out of the surface. To compute  $\mathbf{f}_{ci}$  implicitly, we need the derivative of  $\mathbf{f}_{ci}$  with respect to the particle position. Since this derivative is indefinite, we follow [Zheng et al. 2015] using an all-directional implicit model instead:

$$\mathbf{K}_{cij,t} = - \frac{\partial(\mathbf{f}_{ci,t}, \mathbf{f}_{cj,t})}{\partial(\mathbf{x}_i, \mathbf{x}_j)} = \frac{\sigma \cot \theta_{ij,t}}{2} \begin{pmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{pmatrix}. \quad (11)$$

Assembling  $\mathbf{K}_{cij,t}$  for all pairs of  $i, j$  and triangle  $t$ , we get  $\mathbf{K}_c$ . Since the surface membrane is affected by both the capillary force and fluid's pressure force, we delay the discussion about the membrane's equation of motion to Sec. 5.1.

### 4.3 Rigid Body

To match the time-stepping formulation of the fluid, we discretize the equations of motion of the (articulated) rigid body as follows. At each time step, we first update the rigid body's generalized position as  $\mathbf{q}_r^{n+1} = \mathbf{q}_r^n + \mathbf{v}_r^n \Delta t$ . Then we discretize Eq. 5 semi-implicitly as

$$\mathbf{M}_r \frac{\mathbf{v}_r^{n+1} - \mathbf{v}_r^n}{\Delta t} = \mathbf{f}_r(\mathbf{q}_r^{n+1} + (\mathbf{v}_r^{n+1} - \mathbf{v}_r^n) \Delta t, \mathbf{v}_r^{n+1}). \quad (12)$$

This equation can be further rewritten as

$$(\mathbf{M}_r + \mathbf{D}_r \Delta t + \mathbf{K}_r \Delta t^2) \mathbf{v}_r^{n+1} = \mathbf{M}_r \mathbf{v}_r^n + \mathbf{f}_r(\mathbf{q}_r^{n+1}, \mathbf{v}_r^n) \Delta t + \mathbf{D}_r \mathbf{v}_r^n \Delta t + \mathbf{K}_r \mathbf{v}_r^n \Delta t^2, \quad (13)$$

where  $\mathbf{D}_r = -\frac{\partial \mathbf{f}_r}{\partial \mathbf{v}_r} |_{\mathbf{q}_r^{n+1}, \mathbf{v}_r^n}$ ,  $\mathbf{K}_r = -\frac{\partial \mathbf{f}_r}{\partial \mathbf{q}_r} |_{\mathbf{q}_r^{n+1}, \mathbf{v}_r^n}$ . Finally, we obtain:

$$\hat{\mathbf{M}}_r \mathbf{v}_r^{n+1} = \mathbf{M}_r \mathbf{v}_r^n + \hat{\mathbf{f}}_r(\mathbf{q}_r^{n+1}, \mathbf{v}_r^n), \quad (14)$$

where  $\hat{\mathbf{M}}_r = \mathbf{M}_r + \mathbf{D}_r \Delta t + \mathbf{K}_r \Delta t^2$ , and  $\hat{\mathbf{f}}_r = \mathbf{f}_r + \mathbf{D}_r \mathbf{v}_r^n + \mathbf{K}_r \mathbf{v}_r^n \Delta t$ . This semi-implicit linear system uses the updated generalized positions,  $\mathbf{q}_r^{n+1}$ , and the previous generalized velocities,  $\mathbf{v}_r^n$ , to solve for the next generalized velocities,  $\mathbf{v}_r^{n+1}$ .

## 5 THREE-WAY COUPLING

In this section, we introduce the derivation of our monolithic three-way coupling system for fluid, membrane, and rigid body. Firstly, the membrane-rigid (M-R) and fluid-membrane (F-M) coupling equations are described individually. Then, we explain how we assemble our final three-way coupling equation.

### 5.1 Fluid-Membrane Coupling

We employ the scheme proposed by Robinson-Mosher et al. [2008] to model the F-M coupling, in which fluid uses an Eulerian representation, while surface membrane uses a Lagrangian

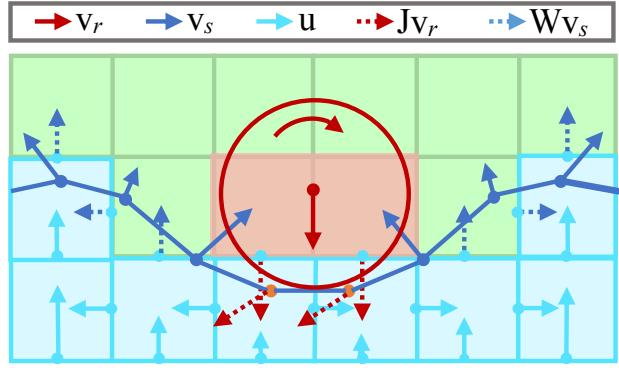


Fig. 5. System DOFs. Solid red arrows show the rigid body's generalized velocities. Solid blue arrows show the surface membrane's particle velocities. Solid cyan arrows show the volumetric fluid's face velocities. Dotted red arrows show the velocities of membrane particles attached to the rigid body and the projected velocities on fluid faces. Dotted blue arrows show the velocities of fluid faces projected from the membrane.

representation. We involve a new type of cell, named "dual cell." As illustrated in the inset figure, the dual cell is centered around the face center, with the same size as the MAC grid. In the entire simulation domain, there are three different kinds of dual cells: fluid-rigid mixed, fluid-air mixed, and all-fluid dual cell. The type of each dual cell is determined by the type of its two incident MAC grid cells without considering the membrane's thickness.

The velocity of the fluid-air dual cell is interpolated from the surface membrane using the interpolation matrix  $\mathbf{W}$ , with row size equal to the number of dual cells, and column size equal to the number of membrane particles. This matrix is used to map the velocity of the membrane particles to the velocity of the fluid's grid. For each dual cell, we traverse each interpenetrated mesh triangle and compute its intersection area. This area is then accumulated to the corresponding entries in  $\mathbf{W}$ . Then,  $\mathbf{W}$  are normalized along row dimension. The transpose,  $\mathbf{W}^T$ , maps the pressure difference at each dual cell to the corresponding particles to provide impulse. It also lumps the mass of fluid at the free surface to the particles. Using  $\mathbf{W}$  we can rewrite the incompressibility equation for fluid as

$$\mathbf{G}^T (\mathbf{u}^* - \frac{1}{\rho} \mathbf{G} \hat{\mathbf{p}} + \mathbf{W} \mathbf{v}_s^{n+1}) = 0. \quad (15)$$

To analyze the dynamics of the membrane, we need to compute the pressure force from the fluid grid. The pressure impulse applied on a dual cell is  $-V \mathbf{G} \hat{\mathbf{p}}$ , where  $V$  is the dual cell's volume. The fluid momentum change is  $\mathbf{M}(\mathbf{W} \mathbf{v}_s^{n+1} - \mathbf{u}^*)$ , where  $\mathbf{M}$  is a diagonal matrix constructed with the fluid mass in each dual cell. Then, the pressure impulse transferred from dual cell to surface membrane is

$$\mathbf{I} = \mathbf{W}^T (-V \mathbf{G} \hat{\mathbf{p}} - \mathbf{M}(\mathbf{W} \mathbf{v}_s^{n+1} - \mathbf{u}^*)). \quad (16)$$

Consequently, the momentum of membrane changes due to the collective effect of pressure impulse and capillary force:

$$\mathbf{M}_s(\mathbf{v}_s^{n+1} - \mathbf{v}_s^n) = \mathbf{f}_c(\mathbf{x}_s^{n+1} + (\mathbf{v}_s^{n+1} - \mathbf{v}_s^n) \Delta t) \Delta t + \mathbf{I}. \quad (17)$$

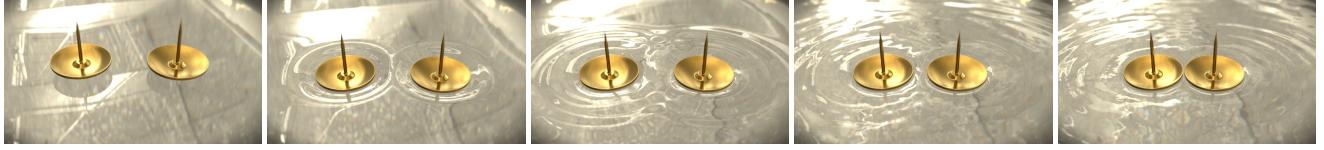


Fig. 6. Attraction between Two Pushpins. Water surface between hydrophobic objects is always slightly lower compared to the rest of the surface. Thus, the capillary force along the perimeter points inward, which forces the pushpins to move slowly toward each other.

For the same reason as we explained in Sec. 4.3, we use the corrected mesh position in  $\mathbf{f}_c(\cdot)$ . By substituting  $\mathbf{I}$  in Eq. 17 into Eq. 16, we obtain the motion equation for membrane:

$$\hat{\mathbf{M}}_s \mathbf{v}_s^{n+1} + \mathbf{W}^T \mathbf{G} \mathbf{V} \hat{\mathbf{p}} = \mathbf{M}_s \mathbf{v}_s^n + \hat{\mathbf{f}}_c(\mathbf{x}_s^{n+1}) \Delta t + \mathbf{W}^T \mathbf{M} \mathbf{u}^*, \quad (18)$$

where  $\hat{\mathbf{M}}_s = \mathbf{M}_s + \mathbf{K}_c \Delta t^2 + \mathbf{W}^T \mathbf{M} \mathbf{W}$ , and  $\hat{\mathbf{f}}_c = \mathbf{f}_c + \mathbf{K}_c \mathbf{v}_s^n \Delta t$ .

Finally, combining Eq. 15 and Eq. 18, the two-way F-M coupling system becomes:

$$\begin{bmatrix} \frac{V}{\rho} \mathbf{G}^T \mathbf{G} & -\mathbf{V} \mathbf{G}^T \mathbf{W} \\ -\mathbf{W}^T \mathbf{G} \mathbf{V} & -\hat{\mathbf{M}}_s \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}} \\ \mathbf{v}_s^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{V} \mathbf{G}^T \mathbf{u}^* \\ -\mathbf{M}_s \mathbf{v}_s^n - \mathbf{W}^T \mathbf{M} \mathbf{u}^* - \hat{\mathbf{f}}_c \Delta t \end{bmatrix}. \quad (19)$$

## 5.2 Membrane-Rigid Coupling

The critical point for M-R coupling resides on how to correctly evaluate adhesion forces between the membrane and the rigid body. We first divide the mesh particles into two sets: free particles  $\mathcal{F}$  and contact particles  $C$ . We set up  $C$  using these three steps:

- (1) If vertex  $i$  and none of its neighboring vertices are in a fluid-air dual cell, put  $i$  in  $C$ .
- (2) If not (1), but the distance from  $\mathbf{x}_i$  to the rigid body's surface is less than a threshold  $\epsilon$  (we use one tenth of grid size), put  $i$  in  $C$ .
- (3) If vertex  $i \in C$  but none of its neighboring vertices are in  $C$ , remove  $i$  from  $C$ .

Since contact particles' velocities are constrained by the rigid body, they are not involved in the coupling system. We use the set  $\mathcal{M}$  to represent the triangle meshes that contain two types of particles, which are usually located around contact boundary regions. We further use  $\mathcal{C}_M$  to indicate contact particles belonging to triangle meshes  $\mathcal{M}$ . The attraction force on particles  $\in \mathcal{C}_M$  can indirectly affect the rigid body through contact. We formulate the capillary force on each of these particles as the summation of the attraction force from all of its incident neighborhood triangles  $\in \mathcal{M}$ . The equivalent adhesion force on the rigid body is  $\sum_{i \in \mathcal{C}_M} \sum_{t \in N_i \cap \mathcal{M}} \mathbf{f}_{ci,t}$ . We can also get the generalized force acting on the rigid body using the Jacobian transpose:

$$\mathbf{f}_a = \sum_{i \in \mathcal{C}_M} \sum_{t \in N_i \cap \mathcal{M}} \mathbf{J}_{ri}^T \mathbf{f}_{ci,t}. \quad (20)$$

To compute  $\mathbf{f}_a$  implicitly, we require the derivative of  $\mathbf{f}_a$  with respect to the free particles' position  $\mathbf{x}_s$  and the rigid body's generalized position  $\mathbf{q}_r$ . The derivative with respect to  $\mathbf{x}_s$  can be obtained using Eq. 11:

$$\mathbf{K}_{a,s,j} = -\frac{\partial \mathbf{f}_a}{\partial \mathbf{x}_j} = \sum_{i \in \mathcal{C}_M} \sum_{t \in N_i \cap \mathcal{M}} \mathbf{J}_{ri}^T \left( -\frac{\sigma \cot \theta_{ij,t}}{2} \right). \quad (21)$$

The derivative with respect to  $\mathbf{q}_r$  can be similarly derived using the chain rule:

$$\mathbf{K}_{a,r} = -\frac{\partial \mathbf{f}_a}{\partial \mathbf{q}_r} = -\sum_{i \in \mathcal{C}_M} \frac{\partial \mathbf{f}_a}{\partial \mathbf{x}_i} \frac{\partial \mathbf{x}_i}{\partial \mathbf{q}_r} = -\sum_{i \in \mathcal{C}_M} \frac{\partial \mathbf{f}_a}{\partial \mathbf{x}_i} \frac{\partial \mathbf{v}_i}{\partial \mathbf{q}_r} = -\sum_{i \in \mathcal{C}_M} \frac{\partial \mathbf{f}_a}{\partial \mathbf{x}_i} \mathbf{J}_{ri}. \quad (22)$$

According to Newton's third law, we can easily get the adhesion force applied to corresponding free particles.  $\mathbf{f}_c$  turns out to be a function of both  $\mathbf{x}_s$  and  $\mathbf{q}_r$ , where the derivative of  $\mathbf{f}_c$  with respect to  $\mathbf{q}_r$  is  $\mathbf{K}_{c,r} = \mathbf{K}_{a,s}^T$ .

## 5.3 Three-way Coupling

The complete description of our system's DOFs can be seen in Fig. 5.

*Fluid.* The fluid velocity on each dual cell can be summarized as:

$$\mathbf{u} = \begin{cases} \mathbf{u}^* - \frac{1}{\rho} \mathbf{G} \hat{\mathbf{p}}, & \text{all-fluid dual cell} \\ \mathbf{W} \mathbf{v}_s, & \text{fluid-air dual cell} \\ \mathbf{J}_r \mathbf{v}_r, & \text{fluid-solid dual cell} \end{cases} \quad (23)$$

where  $\mathbf{J}_r$  is the rigid body Jacobian introduced in Sec. 3. Correspondingly, the fluid incompressibility equation under three-way coupling is:

$$-\mathbf{G}^T (\mathbf{u}^* - \frac{1}{\rho} \mathbf{G} \hat{\mathbf{p}} + \mathbf{W} \mathbf{v}_s + \mathbf{J}_r \mathbf{v}_r) = 0. \quad (24)$$

*Membrane.* By complementing Eq. 18 with adhesion force, we have the membrane's equation of motion under three-way coupling:

$$\tilde{\mathbf{M}}_s \mathbf{v}_s^{n+1} + \mathbf{K}_{c,r} \Delta t^2 \mathbf{v}_r^{n+1} + \mathbf{W}^T \mathbf{G} \mathbf{V} \hat{\mathbf{p}} = \mathbf{M}_s \mathbf{v}_s^n + \tilde{\mathbf{f}}_c \Delta t + \mathbf{W}^T \mathbf{M} \mathbf{u}^*, \quad (25)$$

where  $\tilde{\mathbf{M}}_s = \mathbf{M}_s + \mathbf{K}_c \Delta t^2 + \mathbf{W}^T \mathbf{M} \mathbf{W}$ , and  $\tilde{\mathbf{f}}_c = \mathbf{f}_c(\mathbf{x}_s^{n+1}, \mathbf{q}_r^{n+1}) + \mathbf{K}_c \mathbf{v}_s^n \Delta t + \mathbf{K}_{c,r} \mathbf{v}_r^n \Delta t$ . As we discussed in Sec. 5.2, only free particles are involved in this solve.

*Rigid Body.* The buoyancy force  $\mathbf{f}_b$  on the rigid body is computed using the impulse from the dual cell similar to Eq. 16:

$$\mathbf{f}_b \Delta t = \mathbf{J}_r^T (-\mathbf{G} \mathbf{V} \hat{\mathbf{p}} - \mathbf{M} (\mathbf{J}_r \mathbf{v}_r^{n+1} - \mathbf{u}^*)). \quad (26)$$

By simultaneously applying buoyancy force  $\mathbf{f}_b$ , adhesion force  $\mathbf{f}_a$ , and  $\mathbf{f}_r$ , we get its motion equation under three-way coupling as:

$$\tilde{\mathbf{M}}_r \mathbf{v}_r^{n+1} + \mathbf{K}_{a,s} \Delta t^2 \mathbf{v}_s^{n+1} + \mathbf{J}_r^T \mathbf{G} \mathbf{V} \hat{\mathbf{p}} = \mathbf{M}_r \mathbf{v}_r^n + \tilde{\mathbf{f}}_a \Delta t + \tilde{\mathbf{f}}_r \Delta t + \mathbf{J}_r^T \mathbf{M} \mathbf{u}^*, \quad (27)$$

where  $\tilde{\mathbf{M}}_r = \mathbf{M}_r + \mathbf{D}_r \Delta t + \mathbf{K}_r \Delta t^2 + \mathbf{K}_{a,r} \Delta t^2 + \mathbf{J}_r^T \mathbf{M} \mathbf{J}_r$ ,  $\tilde{\mathbf{f}}_r = \mathbf{f}_r + \mathbf{D}_r \mathbf{v}_r^n + \mathbf{K}_r \mathbf{v}_r^n \Delta t$ , and  $\tilde{\mathbf{f}}_a = \mathbf{f}_a + \mathbf{K}_{a,r} \mathbf{v}_r^n \Delta t + \mathbf{K}_{a,s} \mathbf{v}_s^n \Delta t$ .

Finally we can assemble Eq. 24, Eq. 25, and Eq. 27 into one symmetric three-way coupling system:

$$\begin{bmatrix} \frac{V}{\rho} G^T G & -VG^T W & -VG^T J_r \\ -W^T GV & -\tilde{M}_s & -K_{c,r} \Delta t^2 \\ -J_r^T GV & -K_{a,s} \Delta t^2 & -\tilde{M}_r \end{bmatrix} \begin{bmatrix} \hat{p} \\ v_s^{n+1} \\ v_r^{n+1} \end{bmatrix} = \begin{bmatrix} VG^T u^* \\ -M_s v_s^n - \tilde{f}_c \Delta t - W^T M u^* \\ -M_r v_r^n - \tilde{f}_a \Delta t - \tilde{f}_r \Delta t - J_r^T M u^* \end{bmatrix}. \quad (28)$$

After we compute  $\hat{p}$ , we compute  $u^{n+1}$  using Eq. 7.

## 6 TIME INTEGRATION SCHEME

We first give an overview of our time integration scheme. We devise a prediction-correction scheme to resolve the interaction between the fluid, membrane, and rigid body in each time step. In the prediction step, we update the membrane and the rigid body's positions separately without taking their interaction into consideration. In the correction stage, we first resolve the potential penetration that may have occurred between the membrane and the rigid body by correcting the membrane's contact particles' positions. Then we enforce fluid incompressibility by solving the F-M coupling system. After this two-way coupling solve, we obtain a coherent advection velocity that leads to a collision-free, leakage-free positions without any fluid volume loss. Lastly, the three-way coupling system is solved to guarantee momentum preservation, volume preservation, and interpenetration free velocities.

In each timestep, the algorithm updates the states of the system using the following steps, as depicted in Fig. 7:

- (1) Advance the positions of the rigid body and the membrane particles for one time step.

$$\begin{cases} q_r^{n+1} = q_r^n + v_r^n \Delta t, \\ x_s^* = x_s^n + v_s^n \Delta t. \end{cases} \quad (29)$$

- (2) Perform collision detection and resolve contact by projecting the membrane's contact particles  $C$ 's positions onto the nearest rigid body surface as  $x_c^{**}$ .
- (3) Compute the effective advection velocities  $v_c^\dagger$  of the contact particles  $C$  using

$$v_c^\dagger = \frac{(x_c^{**} - x_c^n)}{\Delta t}. \quad (30)$$

- (4) Revert the membrane's position back to  $x_s^n$ . Then we solve the F-M coupling system Eq. 19, but remove the surface tension part (i.e., remove all the  $f_c$  and  $K_c$ ) and use  $v_c^\dagger$  as the Neumann boundary condition, to get  $\hat{p}^\dagger$  in all fluid cells and  $v_f^\dagger$  for all free particles. By solving this coupling system, we get the fluid's advection velocity

$$u^\dagger = u^n - \frac{1}{\rho} \hat{p}^\dagger, \quad (31)$$

and also the membrane's advection velocity as  $v_s^\dagger = [v_f^\dagger, v_c^\dagger]$ .  $u^\dagger$  and  $v_s^\dagger$  ensures that the fluid preserves volume and that the particles do not collide with the rigid body.

- (5) Advect the membrane's position  $x_s^n$  and the fluid velocity  $u^n$  respectively by the advection velocity  $v_s^\dagger$  and  $u^\dagger$  solved above:

$$\begin{cases} x_s^{n+1} = x_s^n + v_s^\dagger \Delta t, \\ \frac{u^* - u^n}{\Delta t} + u^\dagger \cdot \nabla u^n = 0. \end{cases} \quad (32)$$

Here we follow the standard semi-Lagrangian advection scheme for the fluid.

- (6) Remesh the membrane, and then update the grid cell types and mesh particle types. Our meshing algorithm relies on a set of local operations for topological repair on a triangle mesh. The main idea follows the series of previous works such as EI Topo [Brochu and Bridson 2009]. Then we revert the membrane velocity back to the beginning of the current timestep as  $v_s^n$ , because  $v_s^\dagger$  is only used to update the membrane's position. We also perform velocity interpolation on particles newly created by the mesher.
- (7) Apply external force  $f$  (gravity) to the fluid velocity  $u^*$  and the rigid body's velocity  $v_r^n$ :

$$\begin{cases} u^{**} = u^* + f \Delta t \\ v_r^* = v_r^n + f_r \Delta t. \end{cases} \quad (33)$$

Then we construct and solve the three-way coupling system. Finally, the velocities of contact particles  $C$  are enforced by the corresponding rigid body's velocity.

## 7 RESULTS

In this section, we first describe experiments that help to validate the correctness of our algorithm. Then, we demonstrate the efficacy of our method through an array of rigid-fluid contact simulations dominated by strong surface tension.

### 7.1 Validation

*Membrane Thickness.* In our system, the membrane is not massless but has a virtual thickness  $h$ . In our experiments, we find that  $h$  around  $0.5\Delta x$  is a reasonable choice, where  $\Delta x$  is the grid cell size. As  $h$  increases, the mass and the inertia of the membrane grow accordingly. This will slow down the membrane's local deformation propagation, and consequently, make the surface tension effects less obvious. This can be observed from the comparison result in Fig. 8, in which  $h = 0.5\Delta x$  and  $h = 10\Delta x$  are used respectively.

*Comparison with Level Set.* We compare our membrane's representation of surface tension with the level set's pressure jump method through a bouncing-droplets test shown in Fig. 9. It shows if we choose  $h = 0.5\Delta x$  or  $h = \Delta x$  the droplet simulated using our method can bounce almost synchronously with the one with the level set, but if we set  $h = 10\Delta x$  the oscillation is significantly slower.

*One-time Solve v.s. Two-time Solve.* Our algorithm needs to solve two systems within every single timestep: (1) solving the F-M two-way coupling system to find the effective advection velocity; and (2) solving the three-way coupling system to satisfy all velocity constraints. This is necessary to accommodate our special treatment on contact particles  $C$ . As we enforce both the position and the velocity

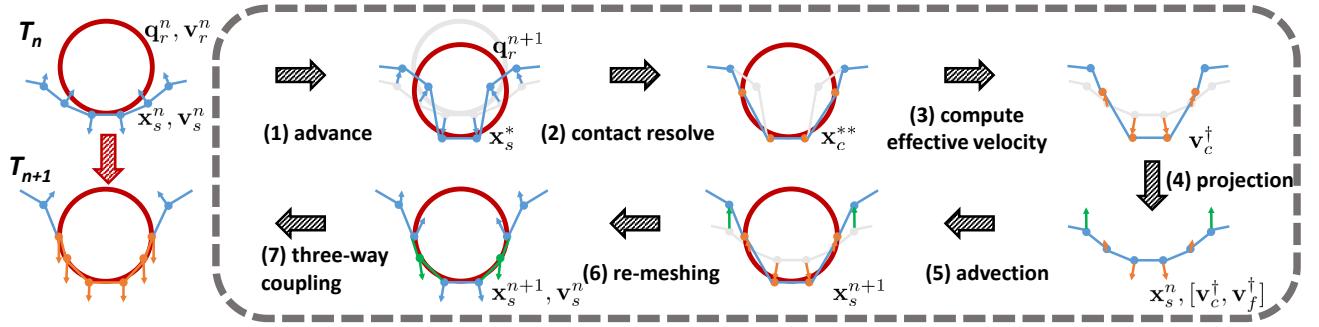


Fig. 7. Our prediction-correction time scheme. In each timestep: (1) Advance the rigid body's and the membrane's position; (2) Resolve contact between the membrane and the rigid body by projecting the contact particles' position; (3) Compute contact particles' effective velocities to resolve interpenetration; (4) Solve the F-M coupling system to get membrane and fluid's advection velocities that guarantee volume preservation of the fluid; (5) Advect the fluid velocity and the membrane's position; (6) Remesh and re-categorize the mesh particles' and grid cells' types; (7) Construct and solve the three-way coupling system. After step (4), we reach a coherent advection velocity that leads to a collision free, leakage free position without any fluid volume loss; after step (7), the three-way coupling system is solved to guarantee momentum preservation, volume preservation, and interpenetration free velocities.

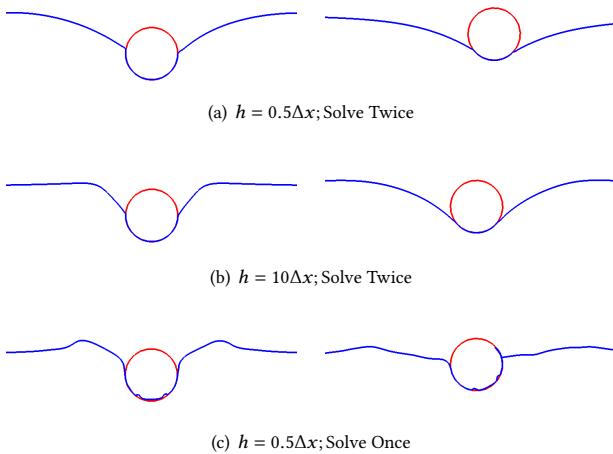


Fig. 8. Validation test. Each row shows two snapshots from our validation tests using different membrane thickness and solving schemes. The configuration of (a) is the normal setup for all our examples; the membrane thickness of (b) is 20 times larger, which makes the membrane surface deformation not propagate far enough; without two-time solve, the membrane surface in (c) has severe sticky artifacts.

constraints of the contact particles, solving only the three-way coupling system will make the contact particles unable to separate from the rigid body in contact by themselves. A comparison between one-time solving (2 only) and two-time solving (1 and 2) is shown in Fig. 8. Without two solves, there are obvious sticking artifacts.

*Surface Tension Coefficient.* We validate the correctness of our surface tension model by setting up a series of simple 2D tests, see Fig. 10. The first row is to verify a classic phenomenon called the “cheerios effect.” The water surface between two hydrophobic objects will be slightly lower compared to its surrounding, causing

an unbalanced force that drives both objects toward each other. The second and third rows aim at comparing the sliding performance of 2D articulated robots between low and high surface tension fluid. Many aquatic creatures rely on surface tension to move and navigate. Our experiments show that high surface tension fluid can help our robot slide more easily. Sliding on a low surface tension fluid is more challenging, making the robot struggle to move forward. A similar phenomenon occurs when the robot tries to jump up from the fluid surface. High surface tension fluid can produce more counter-impulse, thus helping the robot jump higher.

## 7.2 Examples

The physical parameters and performance numbers for all following examples are listed in Tab. 1. We rendered the liquid surface and rigid bodies with triangle meshes in Houdini [2021].

*Sphere Falling into Water.* Fig. 3 shows a sphere falling into a tank of water from the air. The image sequence shows the transition of the sphere from dynamic to static. The surface tension effect is demonstrated by the smoothly curved water surface around the sphere. By modeling the membrane, small-scale wavelets can be naturally generated and diffused even at a low grid resolution.

*Cherries Falling into Water and Milk.* Fig. 13 shows the comparison of a complex geometry falling into two types of fluid with different surface tensions. When falling into a high surface tension fluid (water), the cherries will drain away more fluid during the impact and thus slow down significantly. The cherries can be supported at the interface by buoyancy and capillary forces. For milk, however, the surface tension is not strong enough to counter the impact and support the cherries. Eventually, the cherries will sink to the bottom.

*Paperclips with Different Densities.* Previous demos only show objects with densities similar to the fluid. The power of surface tension, however, is the capability of supporting materials with a much larger density. Our solver can simulate the stable contact between

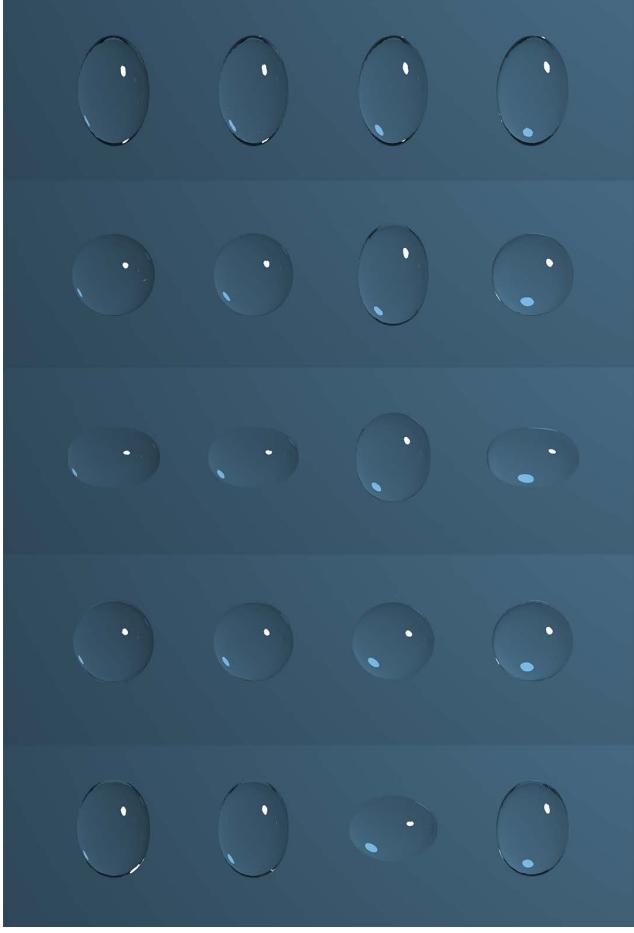


Fig. 9. Bouncing Droplets. Here we show the 1st, 25th, 47th, 67th, and 88th frames of our simulation. The droplets are initialized to be in the same shape and size with  $\sigma = 6 \text{ dyn/cm}$ . From left to right the four droplets are simulated with  $h = 0.5\Delta x$ ,  $h = \Delta x$ ,  $h = 10\Delta x$  and the level set.

water and objects with up to 8 times higher density. Fig. 4 shows the water surface deformed by a set of paperclips with densities ranging from 2 to 7.9. The real density for a paperclip made of steel is about  $8 \text{ g/cm}^3$ . We can see from the refraction in the rendering that heavier paperclips deform the water surface with a longer range and with sharper curvature.

*Breaking Surface and Sinking.* When putting an overweight object on the fluid surface, it will first break the interface then sink. Our solver naturally handles the sinking of such objects and can simulate the progress of water gradually “climbing” onto the object until it completely submerges the object. Fig. 12 shows such progress.

*Boat and Leaves.* Our solver cannot directly solve the contact between thin shells and fluid surface, but with a small modification in the second step of our time scheme—forcing the contact particle to be projected to the bottom side of the thin shell when the particle penetrates it—we can make a thin shell such as a boat and leaves float on top of the water.

*Attraction between Two Pushpins.* Surface tension effect helps generate many interesting natural phenomena. Experiments show that two hydrophobic objects lying on the free surface tend to attract each other. Our simulator can produce this attraction effect, as shown in Fig. 6. The water surface between hydrophobic objects is always slightly lower compared to the rest of the surface. Thus, the capillary force along the perimeter is pointing inward, which forces the objects to move slowly toward each other.

*Water Strider Robot.* Water striders live their lives on the water surface. All of their motions (stand, slide, jump, etc.) rely on surface tension. As shown in Fig. 14, we model a water strider as an articulated rigid body connected by appropriate joints. We kinematically move the joints, causing the insect to move forward. With their middle legs, they push down slightly on the surface of the water, and then push back on the slight ridge resulting from surface tension. When the middle legs finish pushing the water, they are lifted up and brought forward to continue the motion. As can be seen from the accompanying video, the contact points separate properly, allowing the leg to detach from the water without any hindrance.

### 7.3 Performance

We parallelize most of the steps in our method using OpenMP [Dagum and Menon 1998], and implement a GPU version of the Diagonal Preconditioned Conjugate Gradient solver using CUDA [NVIDIA and Vingelmann 2020]. Our 2D examples are performed on a PC with a 16-core 3.40 GHz CPU and an NVIDIA GeForce RTX 2070 graphics card, which also perform the attraction and sinkage of push pin in 3D. Other 3D examples are performed on a workstation with a Xeon W-3175X CPU and an NVIDIA Quadro RTX 8000 graphics card. We summarize our simulation setups, parameters, scale, and time consumption details in Tab. 1. The main bottlenecks are from remeshing and the two solves. Even though our system is indefinite, we use the Diagonal Preconditioned Conjugate Gradient (DPCG) method, with tolerance of  $1e-4$ . We cannot conclude that a DPCG solver can facilitate positive indefinite systems in all cases, but it works well in practice for our examples.

## 8 LIMITATION AND FUTURE WORK

We propose a novel monolithic coupling framework to simulate the surface-tension-dominant contact between fluid and rigid bodies. By introducing a Lagrangian mesh to represent the liquid membrane with a small thickness, our model can handle the interactions between fluid and rigid bodies in a variety of realistic physical settings.

There are several limitations of our current method. First, the three-way coupled system is not symmetric positive definite, which affects the solver’s performance when solving large-scale problems. This problem can be addressed in the future by employing numerical techniques such as [Aanjaneya 2018] and [Bény et al. 2019].

On the other hand, our explicit mesh representation, though exhibiting outstanding performance on handling contact, cannot handle the topological evolution of complex liquid surfaces efficiently. Our mesher supports some simple topological changes such as the ones caused by edge collapse (e.g., see the sinking pushpin example), but its ability is limited when handling more drastic mesh evolutions such as splash, droplet pinch-off, or water bulks merging.

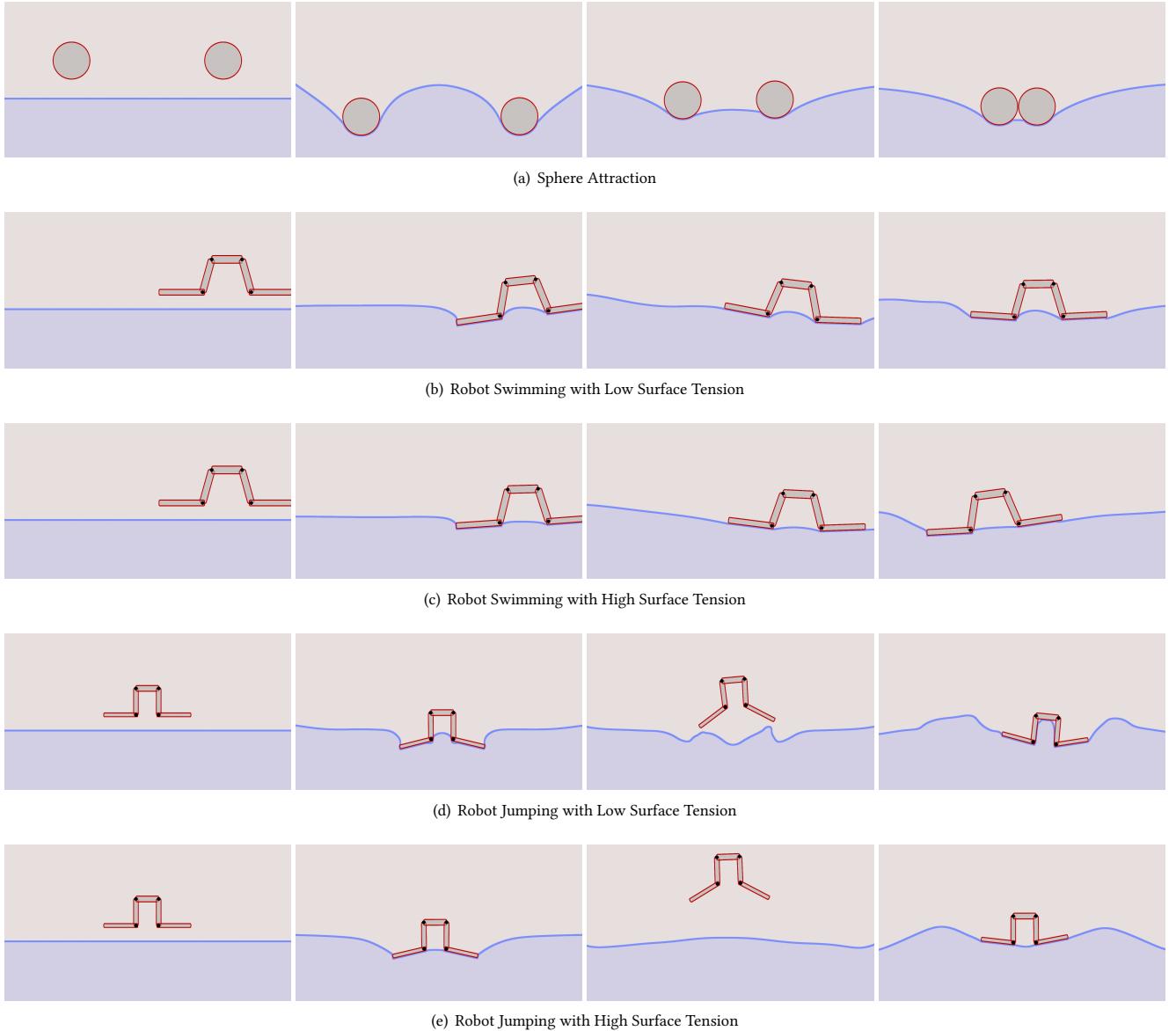


Fig. 10. Surface Tension. (a) "Cheerios effect", two hydrophobic objects lying on free surface tend to attract each other due to unbalance water surface. Comparison between (b) and (c) demonstrates that high surface tension fluid can help the robot slide more easily. Similar phenomenon occurs in (d) and (e), when the robot tries to jump up from the fluid surface. High surface tension fluid can produce more counter-impulse, thus helping the robot jump higher.

Particle-based surface or other connectivity-free Lagrangian methods are worth considering for future work. Regarding the physical model, our current solver does not support hydrophilic materials which limits its scope of applications.

Overall, we believe that the three-way coupling numerical method we proposed in this paper is the first step toward building accurate numerical simulators to model complex solid-fluid interactions with strong surface tension. We hope these computational tools will be useful not only for producing new visual effects for computer

graphics applications but also for facilitating scientific research to better understand these small-scale natural phenomena.

#### ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. This work was supported in part by National Key R&D Program of China (2018YFB1403900). Dartmouth authors acknowledge the funding support from the Burke Research Initiation Award and Toyota TEMA North America Inc. TAMU author acknowledges the

Table 1. Simulation parameters for the examples.

Figure	2D Scenes <sup>†</sup>	$\rho_r$ (g/cm <sup>2</sup> )	Resolution	Cell Size (cm)	First Solve (s)	Second Solve (s)	Time/Step (s)
10(a)	Sphere Attraction	1.1	$160 \times 160$	1.25e-2	0.044	0.047	0.17
10(b,c)	Robot Swimming <sup>‡</sup>	1.1	$320 \times 320$	6.25e-3	0.13	0.19	0.69
10(d,e)	Robot Jumping <sup>‡</sup>	1.1	$320 \times 320$	6.25e-3	0.22	0.21	0.75
Figure	3D Scenes	$\rho_r$ (g/cm <sup>3</sup> )	Resolution	Cell Size (cm)	First Solve (s)	Second Solve (s)	Time/Step (s)
3	Sphere Falling into water	0.4	$80 \times 80 \times 40$	2.5e-1	0.05	0.15	2.1
13	Cherry Water/Milk <sup>§</sup>	1.1	$160 \times 160 \times 80$	1.25e-2	0.32	0.71	10
4	Paperclip	2.0/5.0/7.9	$256 \times 256 \times 64$	7.8125e-2	0.30	6.3	31
12	Push Pin Sinkage	6.8	$256 \times 256 \times 64$	6.25e-2	1.8	4.3	22
6	Push Pin Attraction	3.8	$256 \times 256 \times 64$	6.25e-2	1.9	5.8	25.8
11	Boat and Leaves	-*	$80 \times 80 \times 40$	2.5e-1	0.14	0.32	1.5
14	Water Strider Robot	0.4	$160 \times 200 \times 40$	2.5e-1	0.31	3.5	13

<sup>†</sup> For 2D scenes, the default surface tension coefficient is  $\sigma = 0.9$  dyn, fluid density is  $\rho = 1.0$  g/cm<sup>2</sup>, gravity acceleration is  $g = 9.8$  cm/s<sup>2</sup>. For 3D scenes, default surface tension coefficient is  $\sigma = 72.8$  dyn/cm, fluid density is  $\rho = 1.0$  g/cm<sup>3</sup>, gravity acceleration is  $g = 980$  cm/s<sup>2</sup>.

<sup>‡</sup> 2D robot motions are compared with small surface tension coefficient  $\sigma = 0.3$  dyn.

<sup>§</sup> The surface tension coefficient for milk is  $\sigma = 46$  dyn/cm.

\* The surface density for thin shell is 0.1 g/cm<sup>2</sup>.



Fig. 11. Boat and Leaves. The boat and leaves are modeled by surface meshes with zero thickness. Our algorithm can correctly resolve the collision between the membrane and thin shells.

funding support from the National Science Foundation (CAREER-1846368). We credit the Houdini Education licenses for the video generations.

## REFERENCES

Mridul Aanjaneya. 2018. An efficient solver for two-way coupling rigid bodies with incompressible flow. *Computer Graphics Forum* 37, 8 (2018), 59–68.

Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. on Graphics* 26, 3 (2007), 100–es.

Christopher Batty, Andres Uribe, Basile Audoly, and Eitan Grinspun. 2012. Discrete viscous sheets. *ACM Trans. on Graphics* 31, 4 (2012), 1–7.

Mikhail Belkin and Partha Niyogi. 2008. Towards a theoretical foundation for Laplacian-based manifold methods. *J. of Computer and System Sciences* 74, 8 (2008), 1289–1308.

Mikhail Belkin, Jian Sun, and Yusu Wang. 2009. Constructing Laplace Operator from Point Clouds in Rd. In *Proceedings of the 20'th Annual ACM-SIAM Symposium on Discrete Algorithms*. 1031–1040.

Tyson Brochu and Robert Bridson. 2009. Robust Topological Operations for Dynamic Explicit Surfaces. *SIAM Journal on Scientific Computing* 31, 4 (2009), 2472–2493. <https://doi.org/10.1137/080737617>

J. Bény, C. Kotsalos, and J. Latt. 2019. Toward Full GPU Implementation of Fluid-Structure Interaction. In *2019 18th International Symposium on Parallel and Distributed Computing (ISPD'C)*. 16–22.

Mark Carlson, Peter J. Mucha, and Greg Turk. 2004. Rigid Fluid: Animating the Interplay between Rigid Bodies and Fluid. In *ACM SIGGRAPH 2004 Papers*. Association for Computing Machinery, New York, NY, USA, 1–8.

Yi-Lu Chen, Jonathan Meier, Barbara Solenthaler, and Vinicius C. Azevedo. 2020. An Extended Cut-Cell Method for Sub-Grid Liquids Tracking with Surface Tension. *ACM Trans. Graph.* 39, 6, Article 169 (Nov. 2020), 13 pages.

Nuttapong Chentanez, Tolga G. Goktekin, Bryan E. Feldman, and James F. O'Brien. 2006. Simultaneous Coupling of Fluids and Deformable Bodies. In *ACM SIGGRAPH 2006 Sketches (SIGGRAPH '06)*. 65–es.

Ka Chun Cheung, Leevan Ling, and Steven J Ruuth. 2015. A localized meshless method for diffusion on folded surfaces. *J. of Computational Physics* 297 (2015), 194–206.

Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double bubbles sans toil and trouble: Discrete circulation-based simulation of soap films and foams. *ACM Trans. on Graphics* 34, 4 (2015), 1–9.

Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-only liquids. *ACM Trans. on Graphics* 35, 4 (2016), 1–12.

Leonardo Dagum and Ramesh Menon. 1998. OpenMP: An Industry-Standard API for Shared-Memory Programming. *IEEE Comput. Sci. Eng.* 5, 1 (1998), 46–55.

Richard Fitzpatrick. 2017. Surface tension. In *Theoretical Fluid Mechanics*. IOP Publishing, 1–21.

Oliver G'enevaux, Arash Habibi, and Jean-Michel Dischler. 2003. Simulating Fluid-Solid Interaction. In *Proceedings of the Graphics Interface 2003 Conference, June 11–13, 2003, Halifax, Nova Scotia, Canada*. 31–38.

Robert A Gingold and Joseph J Monaghan. 1977. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 181, 3 (1977), 375–389.

Eran Guendelman, Andrew Selle, Frank Losasso, and Ronald Fedkiw. 2005. Coupling water and smoke to thin deformable and rigid shells. *ACM Trans. on Graphics* 24, 3 (2005), 973–981.

David L. Hu, Manu Prakash, Brian Chan, and John W. M. Bush. 2010. *Water-walking devices*. Springer Berlin Heidelberg, 131–140.

David AB Hyde and Ronald Fedkiw. 2019. A unified approach to monolithic solid-fluid coupling of sub-grid and more resolved solids. *J. of Computational Physics* 390 (2019), 490–526.

Sadashige Ishida, Masafumi Yamamoto, Ryoichi Ando, and Toshiya Hachisuka. 2017. A hyperbolic geometric flow for evolving films and foams. *ACM Trans. on Graphics* 36, 6 (2017), 1–11.

Stoffel D. Janssens, Vikash Chaurasia, and Eliot Fried. 2017. Effect of a surface tension imbalance on a partly submerged cylinder. *Journal of Fluid Mechanics* 830 (2017), 369–386.

Myungjoo Kang, Ronald Fedkiw, and Xu dong Liu. 2000. A Boundary Condition Capturing Method for Multiphase Incompressible Flow. *J. of Scientific Computing* 15 (2000), 323–360.

Rongjie Lai, J. Liang, and H. Zhao. 2013. A local mesh method for solving PDEs on point clouds. *Inverse Problems and Imaging* 7 (2013), 737–755.

P. Lancaster and K. Salkauskas. 1981. Surfaces generated by moving least squares methods. *Math. Comp.* 37 (1981), 141–158.

Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. 1994. *A mathematical introduction to robotic manipulation*. CRC press.

G Navascués. 1979. Liquid surfaces: theory of surface tension. *Reports on Progress in Physics* 42, 7 (jul 1979), 1131–1186.



Fig. 12. Breaking Surface and Sinking. Water gradually “climbing” onto the overweight pushpin until it completely submerges into the water.

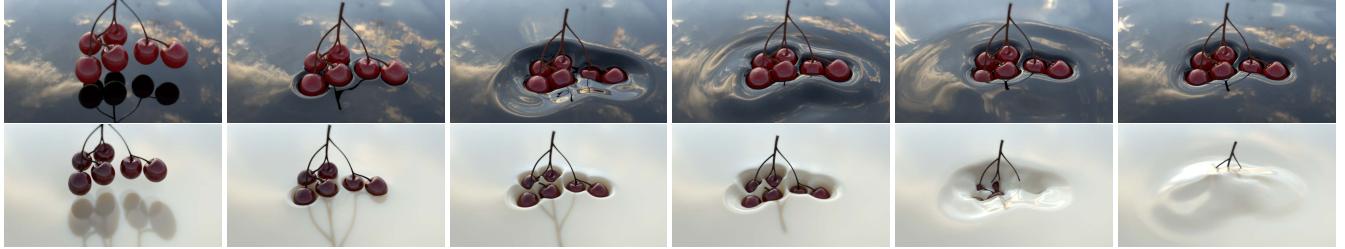


Fig. 13. Cherries Falling into Water and Milk. When cherries fall into a high surface tension fluid (water), they will drain away more fluid during the impact and thus slow down significantly. The cherries can be supported at the interface by buoyancy and capillary forces. Surface-active constituents in milk causes surface tension weakening effect, eventually causing the cherries to sink to the bottom.



Fig. 14. Water Strider Robot. Our water strider robot is modeled using an articulated rigid body. We kinematically control its two middle legs to push down, push back and lift up sequentially. The reaction force cause by surface tension makes our insect robot move forward.

Frank H.P. NVIDIA and Vingelmann, Péter and Fitzek. 2020. CUDA, release: 10.2.89. <https://developer.nvidia.com/cuda-toolkit>

O. Ozcan, H. Wang, J. Taylor, and M. Sitti. 2014. STRIDE II: A Water Strider-inspired Miniature Robot with Circular Footpads. *International Journal of Advanced Robotic Systems* 11 (2014).

Stéphane Popinet. 2018. Numerical Models of Surface Tension. *Annual Review of Fluid Mechanics* 50, 1 (2018), 49–75.

Avi Robinson-Mosher, R Elliot English, and Ronald Fedkiw. 2009. Accurate tangential velocities for solid fluid coupling. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. 227–236.

Avi Robinson-Mosher, Craig Schroeder, and Ronald Fedkiw. 2011. A symmetric positive definite formulation for monolithic fluid structure interaction. *J. of Computational Physics* 230, 4 (2011), 1547–1566.

Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. 2008. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. on Graphics* 27, 3 (2008), 1–9.

Craig Schroeder, Wen Zheng, and Ronald Fedkiw. 2012. Semi-implicit surface tension formulation with a Lagrangian surface mesh on an Eulerian simulation grid. *J. Comput. Phys.* 231, 4 (2012), 2092–2115.

Side Effects Software. 2021. Houdini. <https://www.sidefx.com>.

Y. S. Song and M. Sitti. 2007. Surface-Tension-Driven Biologically Inspired Water Strider Robots: Theory and Experiments. *IEEE Trans. on Robotics* 23 (2007), 578–589.

Jos Stam. 1999. Stable fluids. In *Proc. of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. 121–128.

Steve H. Suhr, Y. S. Song, S. Lee, and M. Sitti. 2005. Biologically Inspired Miniature Water Strider Robot. In *Robotics: Science and Systems*.

Mark Sussman and Mitsuhiro Ohta. 2009. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM J. on Scientific Computing* 31, 4 (2009), 2447–2471.

Hui Wang, Yongxu Jin, Anqi Luo, Xubo Yang, and Bo Zhu. 2020. Codimensional surface tension flow using moving-least-squares particles. *ACM Trans. on Graphics* 39, 4 (2020), 42–1.

Huamin Wang, Peter J Mucha, and Greg Turk. 2005. Water drops on surfaces. *ACM Trans. on Graphics* 24, 3 (2005), 921–929.

Ying Wang, Nicholas J. Weidner, Margaret A. Baxter, Yura Hwang, Danny M. Kaufman, and Shinjiro Sueda. 2019. REDMAX: Efficient & Flexible Approach for Articulated Dynamics. *ACM Trans. on Graphics* 38, 4 (2019), 1–10.

Omar Zarifi and Christopher Batty. 2017. A positive-definite cut-cell method for strong two-way coupling between fluids and deformable bodies. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. 1–11.

Yizhong Zhang, Huamin Wang, Shuai Wang, Yiyi Tong, and Kun Zhou. 2011. A deformable surface model for real-time water drop animation. *IEEE Trans. Visualization & Computer Graphics* 18, 8 (2011), 1281–1289.

Wen Zheng, Jun-Hai Yong, and Jean-Claude Paul. 2009. Simulation of bubbles. *Graphical Models* 71, 6 (2009), 229–239.

Wen Zheng, Bo Zhu, Byungmoon Kim, and Ronald Fedkiw. 2015. A new incompressibility discretization for a hybrid particle MAC grid representation with surface tension. *J. of Computational Physics* 280 (2015), 96–142.

Bo Zhu, Minjae Lee, Ed Quigley, and Ronald Fedkiw. 2015. Codimensional non-Newtonian fluids. *ACM Trans. on Graphics* 34, 4 (2015), 1–9.

B. Zhu, Ed Quigley, Matthew Cong, J. Solomon, and Ronald Fedkiw. 2014. Codimensional surface tension flow on simplicial complexes. *ACM Trans. on Graphics* 33 (2014), 1 – 11.