

A 3D phase-field based Eulerian variational framework for multiphase fluid-structure interaction with contact dynamics

Xiaoyu Mao^a, Rajeev Jaiman^{a,*}

^a*Department of Mechanical Engineering, University of British Columbia, Vancouver, Canada*

Abstract

Using a fixed Eulerian mesh, interface-capturing approaches such as volume-of-fluid, level-set and phase-field methods have been successfully utilized for a broad range of moving boundary problems involving multiphase fluids and single-phase fluid-structure interaction. Nevertheless, multiphase fluids interacting with multiple solids are rarely explored, especially for large-scale finite element simulations with contact dynamics. In this work, we introduce a novel parallelized three-dimensional fully Eulerian variational framework for simulating multiphase fluids interacting with multiple deformable solids subjected to solid-to-solid contact. In the framework, each solid or fluid phase is identified by a standalone phase indicator. Moreover the phase indicators are initialized by the grid cell method, which restricts the calculation to several grid cells instead of the entire domain and provides efficiency for modeling evolving interfaces on a fixed mesh. A diffuse interface description is employed for a smooth interpolation of the physical properties across the phases, yielding unified mass and momentum conservation equations for the coupled dynamical interactions. For each solid object, temporal integration is carried out to track the strain evolution in an Eulerian frame of reference. The coupled differential equations are solved in a partitioned manner and integrated via nonlinear iterations. We first verify the framework against reference numerical data in a two-dimensional case of a rotational disk in a lid-driven cavity flow. The case is generalized to a rotational sphere in a lid-driven cavity flow to showcase large deformation and rotational motion of solids and examine the convergence in three dimensions. We then simulate the falling of an immersed solid sphere on an elastic block under gravitational force to demonstrate the translational motion and the solid-to-solid contact in a fluid environment. Finally, we demonstrate the capability of our proposed framework for a ship-ice interaction problem involving multiphase fluids with an air-water interface and contact between a floating ship and ice floes.

Keywords: Fully Eulerian FSI, Phase-field method, Multiphase fluids, Solid-to-solid contact, Partitioned iterative coupling

1. Introduction

Multiphase fluid-structure interaction is a bi-directional coupling of multiphase fluids with moving/deforming structures, where the structures can contact each other. It is frequently encountered in many natural phenomena and engineering applications. Practical applications

*Corresponding author

Email address: r.jaiman@mech.ubc.ca (Rajeev Jaiman)

include the formation and clearance of blood clots [1, 2], soft robotics [3] in multiphase flow, floating offshore structures, subsea pipelines and of particular interest of the current work, the ship-ice interaction in the Arctic region [4]. These problems usually involve large translational and rotational motion of solids, the resulting fluid-structure interaction (FSI), evolving interfaces of multiphase fluids, and contact between solids. In such scenarios, robust and accurate modeling of fluid-solid and fluid-fluid interfaces with solid-to-solid contact interactions is very challenging for the state-of-the-art numerical methods. The simulation of such cases requires a general multiphase and multiphysics numerical framework to treat evolving interfaces consistently and overcome intrinsic differences between the kinematics and dynamics of fluids and solids.

Various techniques treating the fluid-solid interface have been developed in the past several decades. As one of the most accurate approaches, the arbitrary Lagrangian-Eulerian method [5] has been widely used and demonstrated its success in solving FSI problems with an oscillatory motion of solids. However, it may fail for solids subjected to large translational and rotational motion, which can cause significant mesh distortion during the evolution of the fluid-solid interface. Topological changes in the contact between solids are also quite challenging to handle. To resolve the mesh distortion, one can use independent meshes for fluids and solids and couple the solutions with interpolation or Lagrangian multiplier, as conducted in the immersed boundary methods [6], overset grid methods [7, 8] and fictitious domain methods [9, 10]. In these methods, the fluid-solid coupling in non-matching grids can violate conservation laws and boundary conditions and thus must be judiciously constructed. Another approach to resolve the inconsistency between fluids and solids is to unify the kinematics and dynamics in an Eulerian frame of reference [11, 12]. Through an interface-capturing approach on a fixed mesh, the fully Eulerian approach allows for arbitrary motion and topological changes. The velocity and stress continuity at fluid-fluid and fluid-solid interfaces are naturally satisfied via unified velocity field and momentum equations. These traits bring convenience and robustness for numerical simulations of multiphase FSI problems with reasonable accuracy, thus being the focus of the current work.

While alleviating the issues in interface motion and dynamics, two major challenges arise from the fully Eulerian description namely (i) accuracy of interface representation and evolution, and (ii) evaluation of displacement and strain of solids in an Eulerian frame of reference. For the first challenge, we employ our recently proposed interface-preserving phase-field method [13]. The phase-field method involves a diffuse interface description, where the interface is modeled as a transitional region with finite thickness, across which the physical properties vary smoothly. The transition pattern is adaptively regularized to maintain a consistent and smooth interface transition in the diffuse interface description. For the first time, we apply this description to all the interfaces involved in multiphase FSI problems, namely the fluid-fluid interface Γ^f , the fluid-solid interface Γ^{fs} and the solid-solid interface Γ^{ss} between the fluid domains Ω^f and the solid domains Ω^s as illustrated in Fig. 1. With this approach, the requirement of resolving the interface to satisfy the velocity and traction continuity conditions with the standard finite element method can be circumvented. For the second challenge, one can consider the initial point set method [11], the reference map technique [14], and the evolution of left Cauchy-Green tensor [15, 16]. To maintain accuracy during the evaluation from the solid displacement to strain, we directly evolve the left Cauchy-Green tensor in the current work.

The present work is an extension of the work carried out in our earlier work [17], wherein

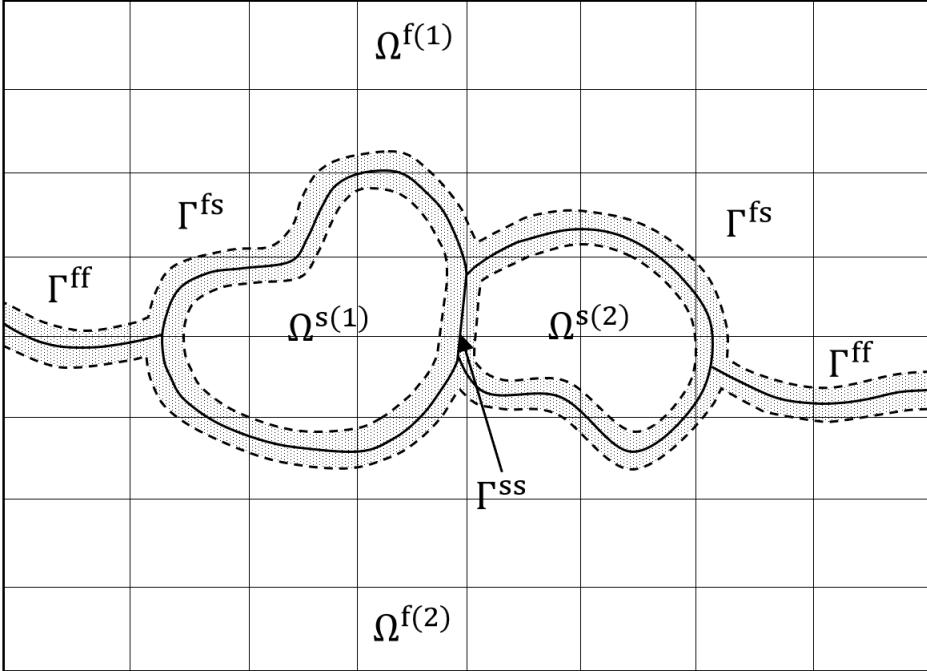


Figure 1: An abstract representation of fluid-fluid and fluid-solid interfaces in multiphase fluid-solid interaction problems. The diffuse interface in the Eulerian mesh is shown as the shaded region.

a two-dimensional (2D) variational framework for a fully Eulerian single-phase FSI is established. In the current work, the framework is extended to three-dimensional (3D) multiphase fluids interacting with multiple solid objects with solid-to-solid contact dynamics. The framework employs a combined formulation for multiphase flow and fluid-structure interaction using our interface-preserving method for unstructured meshes. We consider a 3D finite element discretization in a general computational domain with given boundary conditions. The interfaces are identified by multiple order parameters indicating the local composition while evolving by the interface-preserving phase-field equations. The resulting order parameters are used to unify the mass and momentum conservation equations via phase-dependent interpolation. In the regions indicated as solid domain, the left Cauchy-Green tensor is tracked to complement the solid strain, hence completing the Eulerian multiphase FSI framework. The equations are solved via the Newton-Raphson method and integrated in a partitioned iterative manner. Solving these partial differential equations for the coupled multiphase FSI system in a 3D domain can be computationally very expensive. To make the simulations practical and robust, we employ a stabilized Petrov-Galerkin discretization and a fully implicit time marching scheme to enable a coarser level of spatial and temporal resolutions. The framework is realized through multi-processor parallelized implementation to further accelerate the simulations.

Through the aforementioned computational efforts, we have successfully built a fully integrated, stable and parallelized 3D fully Eulerian finite element solver for general-purpose multiphase FSI problems. We showcase the capability and validity of the framework for the test cases of increasing complexity, namely a rotating disk and a sphere in a lid-driven cavity flow, falling and bouncing back of a solid sphere on an elastic block. We highlight the accuracy and convergence of the framework to capture the fluid-solid interface with solid-to-solid contact interactions. In practical multiphase FSI problems, diffuse interfaces for solid objects can have complex geometries that cannot be generated through simple analytical expressions.

In these cases, the initialization of the order parameter field needs to be achieved in a general and robust manner during the pre-processing stage. To accomplish this goal, we suggest a phase-field interface processing algorithm by integrating the grid cell method and ray-casting algorithm. A novel non-duplicated plane partitioning is developed for the discrete implementation of the ray-casting algorithm. Finally, we demonstrate the geometry processing utilities with our proposed fully Eulerian finite element framework for a simplified situation of ship-ice interaction.

The organization of this paper is as follows: Section 2 describes the formulations of the unified Eulerian variational framework for multiphase FSI. Section 3 presents the methodologies to initialize the order parameter field for complex geometries. We first examine the initialization of the order parameters in Section 4. After that, the multiphase FSI framework is verified and demonstrated in cases with increasing complexity in Section 5. The ship-ice interaction is discussed and demonstrated in Section 6. Finally, we conclude the work in Section 7.

2. Fully Eulerian formulation for multiphase FSI

In this section, we present the continuous and semi-discrete formulations for the fully Eulerian multiphase FSI framework. We start with the unified mass and momentum conservation equations for the continuum. Then we describe the interface equations and the constitutive relation for the fluid phases. We next introduce the interface equations, the constitutive relation, and the kinematic equations for the solid phases. Specifically, we provide a detailed description of the linearization of the solid stress. Finally, we discuss some implementation details in the framework.

2.1. Conservation laws for the multiphase FSI system

We present the partial differential equations for the unified mass and momentum conservation at the continuous level and then review the semi-discrete variational formulations using a stabilized finite element discretization.

2.1.1. Strong differential form

Consider a physical domain $\Omega \times]0, T[$ with spatial coordinates \mathbf{x} and temporal coordinate t . The domain comprises subdomains Ω_t^i , $i = 1, 2, \dots, n$, where i and n are the index and the total number of the subdomains. Each subdomain is considered to be an incompressible Newtonian fluid or Neo-Hookean solid of particular physical properties. The subdomains are distinguished by a fraction function $\alpha^i(\phi^i(\mathbf{x}, t))$, where $\phi^i(\mathbf{x}, t)$ being the order parameter in the phase-field method. Both $\alpha^i(\phi^i)$ and ϕ^i are used to indicate a binary system that the current location (\mathbf{x}, t) is or is not phase i , which leads to $\alpha^i(\phi^i(\mathbf{x}, t)) = 1$, $\mathbf{x} \in \Omega_t^i$ and $\alpha^i(\phi^i(\mathbf{x}, t)) = 0$, $\mathbf{x} \in \Omega \setminus \Omega_t^i$. In the current work, $\alpha^i(\phi^i) = \frac{1+\phi^i}{2}$ is used. Through phase-dependent interpolation, the density, viscosity, stress, and the body force of the continua can be calculated as: $\rho = \sum_{i=1}^n \alpha^i(\phi^i) \rho^i$, $\nu = \sum_{i=1}^n \alpha^i(\phi^i) \nu^i$, $\mathbf{b} = \sum_{i=1}^n \alpha^i(\phi^i) \mathbf{b}^i$ and $\boldsymbol{\sigma} = \sum_{i=1}^n \alpha^i(\phi^i) \boldsymbol{\sigma}^i$. Along with the incompressible constraint, the momentum and mass conservation can be written as:

$$\rho(\partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v}) = \nabla \cdot \boldsymbol{\sigma} + \mathbf{b}, \quad (1)$$

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

where \mathbf{v} is a unified velocity field for the continua.

2.1.2. Semi-discrete variational form

We employ the generalized- α technique [18] for the temporal discretization which enables a user-controlled high frequency damping desirable for coarse discretizations in space and time. For a first-order system of variable φ , the generalized- α method is given by:

$$\partial_t \varphi^{n+\alpha_m} = f(\varphi^{n+\alpha}), \quad (3)$$

$$\varphi^{n+1} = \varphi^n + \Delta t \partial_t \varphi^n + \Delta t \varsigma (\partial_t \varphi^{n+1} - \partial_t \varphi^n), \quad (4)$$

$$\partial_t \varphi^{n+\alpha_m} = \partial_t \varphi^n + \alpha_m (\partial_t \varphi^{n+1} - \partial_t \varphi^n), \quad (5)$$

$$\varphi^{n+\alpha} = \varphi^n + \alpha (\varphi^{n+1} - \varphi^n), \quad (6)$$

where α , α_m and ς are the generalized- α parameters depending on the user-defined spectral radius ρ_∞ :

$$\alpha = \frac{1}{1 + \rho_\infty}, \quad \alpha_m = \frac{1}{2} \left(\frac{3 - \rho_\infty}{1 + \rho_\infty} \right), \quad \varsigma = \frac{1}{2} + \alpha_m - \alpha. \quad (7)$$

The temporal evolution works in a predictor-multicorrector manner. In every nonlinear iteration, we first predict the solution at $n + 1$, interpolate the solution to $n + \alpha$ and solve the first-order system. After that, we correct the solution at $n + 1$ according to the solution at $n + \alpha$. In the current work, the spectral radius is selected as $\rho_\infty = 0$. This applies to the temporal discretization of all equations hereafter.

Now we introduce the spatial discretization using the stabilized finite element method. Suppose \mathcal{S}_v^h and \mathcal{S}_p^h denote the space of trial solutions such that:

$$\mathcal{S}_v^h = \{ \mathbf{v}_h \mid \mathbf{v}_h \in (H^1(\Omega))^d, \mathbf{v}_h = \mathbf{v}_D \text{ on } \Gamma_D \}, \quad (8)$$

$$\mathcal{S}_p^h = \{ p_h \mid p_h \in L^2(\Omega) \}, \quad (9)$$

where $(H^1(\Omega))^d$ denotes the space of square-integrable \mathbb{R}^d -valued functions with square-integrable derivatives on Ω , $L^2(\Omega)$ is the space of the scalar-valued functions that are square-integrable on Ω and Γ_D represents the Dirichlet boundaries with the value of \mathbf{v}_D . Similarly, we define \mathcal{V}_ψ^h and \mathcal{V}_q^h as the space of test functions such that:

$$\mathcal{V}_\psi^h = \{ \boldsymbol{\psi}_h \mid \boldsymbol{\psi}_h \in (H^1(\Omega))^d, \boldsymbol{\psi}_h = \mathbf{0} \text{ on } \Gamma_D \}, \quad (10)$$

$$\mathcal{V}_q^h = \{ q_h \mid q_h \in L^2(\Omega) \}. \quad (11)$$

The variational statement of the momentum and mass conservation equations can be written as: find $[\mathbf{v}_h(t^{n+\alpha}), p_h(t^{n+1})] \in \mathcal{S}_v^h \times \mathcal{S}_p^h$ such that $\forall [\boldsymbol{\psi}_h, q_h] \in \mathcal{V}_\psi^h \times \mathcal{V}_q^h$,

$$\begin{aligned} & \int_{\Omega} \rho(\boldsymbol{\phi}_h^{n+\alpha}) (\partial_t \mathbf{v}_h^{n+\alpha_m} + \mathbf{v}_h \cdot \nabla \mathbf{v}_h) \cdot \boldsymbol{\psi}_h d\Omega + \int_{\Omega} \boldsymbol{\sigma}_h(\boldsymbol{\phi}_h^{n+\alpha}) : \nabla \boldsymbol{\psi}_h d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \frac{\tau_m}{\rho(\boldsymbol{\phi}_h^{n+\alpha})} (\rho(\boldsymbol{\phi}_h^{n+\alpha}) \mathbf{v}_h \cdot \nabla \boldsymbol{\psi}_h + \nabla q_h) \cdot \mathcal{R}_m d\Omega^e \\ & + \int_{\Omega} q_h (\nabla \cdot \mathbf{v}_h) d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \nabla \cdot \boldsymbol{\psi}_h \tau_c \rho(\boldsymbol{\phi}_h^{n+\alpha}) \mathcal{R}_c d\Omega^e \\ & = \int_{\Omega} \mathbf{b}(\boldsymbol{\phi}_h^{n+\alpha}) \cdot \boldsymbol{\psi}_h d\Omega, \end{aligned} \quad (12)$$

where ϕ is a vector of all the order parameters used for the phase-dependent interpolation, $\mathbf{A} : \mathbf{B} = A_{i,j} B_{i,j}$, \mathcal{R}_m and \mathcal{R}_c denote the element-wise residuals of the momentum and mass conservation equations respectively, and n_{el} is the number of elements. In Eq. (12), the terms in the first line represent the Galerkin projection of the momentum conservation equation in the test function space ψ_h and the second line comprises of the Petrov-Galerkin stabilization term for the momentum conservation equation. The third line denotes the Galerkin projection and stabilization terms for the mass conservation equation. The terms in the last line are the Galerkin projection of the body force. The stabilization parameters τ_m and τ_c in Eq. (12) are given by [19, 20]:

$$\tau_m = \left(\left(\frac{2}{\Delta t} \right)^2 + \mathbf{v}_h \cdot \mathbf{G} \mathbf{v}_h + C_I \left(\frac{\mu(\phi)}{\rho(\phi)} \right)^2 \mathbf{G} : \mathbf{G} \right)^{-1/2}, \quad \tau_c = \frac{1}{\text{tr}(\mathbf{G}) \tau_m}, \quad (13)$$

where C_I is a constant derived from the element-wise inverse estimates [21], \mathbf{G} is the element contravariant metric tensor and $\text{tr}(\mathbf{G})$ is the trace of the contravariant metric tensor. This stabilization in the variational form circumvents the Babuška-Brezzi condition which needs to be satisfied by any standard mixed Galerkin method [22].

2.2. Equations for the fluid phases

The fluids can be solved naturally in an Eulerian frame of reference. Therefore, we only need a phase-field equation to track the interface of each fluid phase. To maintain a consistent transition of physical properties across the diffuse interface, we employ the interface-preserving phase-field method developed in [13]. We first present the Allen-Cahn phase-field equation and the time-dependent mobility model. We next turn our attention to the stabilized semi-implicit form of our formulation.

2.2.1. Strong differential form

For each phase of a Newtonian fluid, we employ the interface-preserving phase-field method[13], which is given as:

$$\frac{\partial \phi^i}{\partial t} + \mathbf{v} \cdot \nabla \phi^i = -\gamma^i(t) \left(F'(\phi^i) - \varepsilon^2 \nabla^2 \phi^i - \beta(t) \sqrt{F(\phi^i)} \right), \quad (14)$$

$$\gamma^i(t) = \frac{1}{\eta} \mathcal{F} \left(\left| \frac{(\nabla \phi^i)^T \cdot \nabla \mathbf{v} \cdot \nabla \phi^i}{|\nabla \phi^i|^2} \right| \right), \quad (15)$$

where ε is the interface thickness parameter, $F(\phi^i) = \left(0.25 \left((\phi^i)^2 - 1 \right)^2 \right)$ is the bulk energy, $\beta(t) = \frac{\int_{\Omega} F'(\phi^i) d\Omega}{\int_{\Omega} \sqrt{F(\phi^i)} d\Omega}$ is the Lagrange multiplier for mass conservation [23]. The RHS of Eq. 14 is the gradient flow of the free energy functional converging to a hyperbolic tangent profile, the magnitude of which is proportional to the mobility coefficient γ . The time-dependent mobility coefficient $\gamma^i(t)$ adaptively regularizes the hyperbolic tangent profile to address the convective distortion, which is the intensity of compression or extension on the diffuse interface due to the convection effect. The convective distortion is quantified by the projection of the velocity gradient in the normal direction of the diffuse interface region. We consider $\mathcal{F}(\varphi(\mathbf{x}, t)) = \sqrt{\frac{\int (\varphi(\mathbf{x}, t))^2 d\Omega}{\int 1 d\Omega}}$, where $-\delta < \phi < \delta$ is the RMS value of the function field $\varphi(\mathbf{x}, t)$ in the interface region, where the boundaries of the region are selected as $\delta = 0.9$. The parameter η is a user-defined value to control the regularization and its selection is discussed in [13].

2.2.2. Discrete form of the time-dependent mobility model

We present the discrete form of the time-dependent mobility model in this subsection. The strong form of the time-dependent mobility model is given by Eq. (15):

$$\gamma(t) = \frac{1}{\eta} \mathcal{F} \left(\left| \frac{(\nabla\phi)^T \cdot \nabla\mathbf{v} \cdot \nabla\phi}{|\nabla\phi|^2} \right| \right), \quad (16)$$

where $\mathcal{F}(\varphi(\mathbf{x}, t)) = \sqrt{\frac{\int (\varphi(\mathbf{x}, t))^2 d\Omega}{\int 1 d\Omega}}$, $\mathbf{x} \in \Gamma_{DI}^\phi(t)$ and $\Gamma_{DI}^\phi(t) = \{(\mathbf{x}, t) | |\phi(\mathbf{x}, t)| \leq 0.9\}$. In Eq. (16), $\mathcal{F} \left(\left| \frac{(\nabla\phi)^T \cdot \nabla\mathbf{v} \cdot \nabla\phi}{|\nabla\phi|^2} \right| \right)$ can be approximated as the RMS of $|(\nabla\phi)^T \cdot \nabla\mathbf{v} \cdot \nabla\phi / |\nabla\phi|^2|$ at all the nodes located inside the diffuse interface region $\Gamma_{DI}^\phi(t)$.

The nodal value of $|(\nabla\phi)^T \cdot \nabla\mathbf{v} \cdot \nabla\phi / |\nabla\phi|^2|$ is calculated as follows. The nodal value of $\mathbf{v}_h(t^{n+\alpha})$, $\phi_h(t^{n+\alpha})$ is used to interpolate the $\nabla\mathbf{v}_h(t^{n+\alpha})$ and $\nabla\phi_h(t^{n+\alpha})$ at the quadrature points, and L^2 -projection is used to project the value on the quadrature points back to the nodes inside the diffuse interface region [24]. If the node lies outside the diffuse interface region, the value is assigned to be zero. For example, for the node p , we have:

$$\left| \left(\frac{(\nabla\phi_h)^T \cdot \nabla\mathbf{v}_h \cdot \nabla\phi_h}{|\nabla\phi_h|^2} \right)_p \right| = \begin{cases} \left| \frac{\sum_e \int_{\Omega^e} N_p((\nabla\phi_h)^T \cdot \nabla\mathbf{v}_h \cdot \nabla\phi_h) / |\nabla\phi_h|^2 d\Omega^e}{\sum_e \int_{\Omega^e} N_p d\Omega^e} \right| & \text{if } |\phi_p| \leq 0.9, \\ 0 & \text{if } |\phi_p| > 0.9, \end{cases} \quad (17)$$

where N_p represents the shape function at node p . The discrete form of the time-dependent mobility model is given by:

$$\gamma(t^{n+\alpha}) = \frac{1}{\eta} \left(\sqrt{\frac{1}{n_{DI}} \sum_{p=1}^{n_{DI}} \left| \left(\frac{(\nabla\phi_h)^T \cdot \nabla\mathbf{v}_h \cdot \nabla\phi_h}{|\nabla\phi_h|^2} \right)_p \right|^2} \right), \quad (18)$$

where n_{DI} denotes the number of the nodes lying inside the diffuse interface region.

The L^2 -projection in Eq. (17) needs extra attention in a multi-processor parallelized environment. In the current implementation, we first calculate the numerator and denominator at the RHS of Eq. 17 on each partitioned mesh block. For boundary nodes between the mesh blocks, message passing is performed to take the contribution from adjacent mesh blocks. After the numerator and the denominator are exchanged synchronously across the mesh, the division is calculated as the last step. A detailed discussion regarding the parallelized implementation of the L^2 -projection can be found in Appendix A.

2.2.3. Semi-discrete variational form

Suppose $\mathcal{S}_{\phi^i}^h$ and \mathcal{V}_ψ^h denote the space of trial solution and test functions such that:

$$\mathcal{S}_{\phi^i}^h = \{ \phi_h^i \mid \phi_h^i \in H^1(\Omega), \phi_h^i = \phi_D^i \text{ on } \Gamma_D^{\phi^i} \}, \quad (19)$$

$$\mathcal{V}_\psi^h = \{ \psi_h \mid \psi_h \in H^1(\Omega), \psi_h = 0 \text{ on } \Gamma_D^{\phi^i} \}, \quad (20)$$

where $\Gamma_D^{\phi^i}$ denotes the Dirichlet boundaries for the order parameter with boundary value of ϕ_D^i . The variational statement of the Allen-Cahn equations can be written as: find $\phi_h^i(t^{n+\alpha}) \in \mathcal{S}_{\phi^i}^h$

such that $\forall \psi_h \in \mathcal{V}_\psi^h$ for the Allen-Cahn equation:

$$\int_{\Omega} \left(\psi_h \partial_t \phi_h^{i,n+\alpha_m} + \psi_h (\mathbf{v}_h^{n+\alpha} \cdot \nabla \phi_h^i) + \gamma^i(t^{n+\alpha}) (\nabla \psi_h \cdot (\hat{k} \nabla \phi_h^i) + \psi_h \hat{s} \phi_h^i - \psi_h \hat{f}) \right) d\Omega \\ + \sum_{e=1}^{n_{el}} \int_{\Omega_e} (\mathbf{v}_h^{n+\alpha} \cdot \nabla \psi_h) \tau_{\phi^i} \mathcal{R}_{\phi^i}^i d\Omega_e = 0, \quad (21)$$

where \mathbf{v} , \hat{k} , \hat{s} and \hat{f} are the flow velocity, the modified diffusion coefficient, modified reaction coefficient and modified source respectively which are defined in [25], $\mathcal{R}_{\phi^i}^i$ denotes the element-wise residuals for the Allen-Cahn equation of the i -th fluid phase and the time-dependent mobility is denoted as $\gamma^i(t^{n+\alpha})$. In Eq. (21), the first line is the Galerkin projection of the transient, convection, diffusion, reaction and source terms, the second line represents the Petrov-Galerkin stabilization terms where the stabilization parameters $\tau_{\phi^i}^i$ are given by [19, 20]:

$$\tau_{\phi^i}^i = \left(\left(\frac{2}{\Delta t} \right)^2 + \mathbf{v}_h \cdot \mathbf{G} \mathbf{v}_h + 9\hat{k}^2 \mathbf{G} : \mathbf{G} + \hat{s}^2 \right)^{-1/2}. \quad (22)$$

For the current FSI model, the stress of the incompressible Newtonian fluid used is given by:

$$\boldsymbol{\sigma}^i = -p \mathbf{I} + \mu^i (\nabla \mathbf{v} + (\nabla \mathbf{v})^T), \quad (23)$$

where p is the hydrostatic pressure and μ^i is the dynamic viscosity for the i -th fluid.

2.3. Equations for the solid phases

For each phase of the Neo-Hookean solid, we employ the interface-preserving phase-field method to capture the interface on a fixed mesh. The equations and discretizations are the same as introduced in subsection 2.2. In addition, the strain of the solid needs to be tracked. In the current work, it is accomplished by evolving the left Cauchy-Green tensor. The governing equations are presented first, following the semi-discrete variational form. While the contact between solids can be solved with the current framework via the phase-field method on a fixed mesh, its mechanism needs to be further clarified. In the collision of two solid objects, the contact solids are subjected to no-penetration and force balance conditions at the contact point or surface. In the current work, we only consider the elastic restoring force from the deformation of New-Hookean models with no-slip and no-penetration conditions at the contact point/surface. We want to demonstrate that by tracking each solid object with an independent order parameter and a left Cauchy-Green tensor, the force balance is naturally handled by the stress continuity, while the no-penetration condition and no-slip condition can be naturally satisfied with velocity continuity in a unified velocity field.

2.3.1. Strong differential form

The solid strain is tracked by the evolution of the left Cauchy-Green tensor. However, the evolved components of the tensor persist in the computational field after the solid left the region. When the solid crosses the region again, the remaining values become unphysical initialization, which may result in numerical instability and wrong solutions. To resolve this issue, we try to

recover $\mathbf{B} = \mathbf{I}$ outside of the solid domain at every time step employing an approach inspired by [16]:

$$\alpha^i(\phi^i) \left(\frac{\partial \mathbf{B}^i}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{B}^i - \nabla \mathbf{v} \mathbf{B}^i - \mathbf{B} (\nabla \mathbf{v})^T \right) + (1 - \alpha^i(\phi^i))(\mathbf{B}^i - \mathbf{I}) = 0. \quad (24)$$

With the left Cauchy-Green tensor readily available, the solid stress is calculated as $\boldsymbol{\sigma}^i = -p\mathbf{I} + \mu_L^i(\mathbf{B}^i - \mathbf{I}) + \mu^i(\nabla \mathbf{v} + (\nabla \mathbf{v})^T)$, where μ_L^i and μ^i are the shear modulus and the viscosity for the i -th solid, respectively.

2.3.2. Semi-discrete variational form

For the left Cauchy-Green tensor, we define the space of trial solution $\mathcal{S}_{\mathbf{B}^i}^h$ and test functions \mathcal{V}_{ψ}^h as:

$$\mathcal{S}_{\mathbf{B}^i}^h = \{ \mathbf{B}_h^i \mid \mathbf{B}_h^i \in (H^1(\Omega))^d \}, \quad (25)$$

$$\mathcal{V}_{\psi}^h = \{ \psi_h \mid \psi_h \in (H^1(\Omega))^d \}. \quad (26)$$

The variational statement of the Cauchy-Green tensor can be written as: find $\mathbf{B}_h^i(t^{n+\alpha}) \in \mathcal{S}_{\mathbf{B}^i}^h$ such that $\forall \psi_h \in \mathcal{V}_{\psi}^h$

$$\begin{aligned} & \alpha(\phi^{n+\alpha}) \int_{\Omega} \left(\partial_t \mathbf{B}_h^{i,n+\alpha} + (\mathbf{v}_h^{n+\alpha} \cdot \nabla) \mathbf{B}_h^i - \nabla \mathbf{v}_h^{n+\alpha} \mathbf{B}_h^i - \mathbf{B}_h^i (\nabla \mathbf{v}_h^{n+\alpha})^T \right) : \psi_h d\Omega \\ & + (1 - \alpha(\phi^{n+\alpha})) \int_{\Omega} (\mathbf{B}_h^i - \mathbf{I}) : \psi_h d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau_{\mathbf{B}^i} ((\mathbf{v}_h^{n+\alpha} \cdot \nabla) \psi_h) : \mathcal{R}_{\mathbf{B}^i} d\Omega^e = 0, \end{aligned} \quad (27)$$

where $\mathcal{R}_{\mathbf{B}^i}$ denotes the element-wise residuals of the evolution equation for the left Cauchy-Green tensor. In Eq. (27), the first line is the Galerkin projection, the second line represents the strain recovery term and the Petrov-Galerkin stabilization term. The stabilization parameter is selected as $\tau_{\mathbf{B}^i} = ((2/\Delta t)^2 + \mathbf{v}_h^{n+\alpha} \cdot \mathbf{G} \mathbf{v}_h^{n+\alpha})$.

2.4. Linearization of the solid stress

We employ Newton-Raphson iterations to find the solution of the momentum conservation equation in our fully implicit implementation. Therefore, the residuals and directional derivatives associated with the solid stress are needed. While the discretizations of the hydrostatic pressure and viscous terms are quite standard, the shear components for the i -th solid $\mu_L^i(\mathbf{B}^{i,n+\alpha} - \mathbf{I})$, as a function of the left Cauchy-Green tensor, need to be handled carefully. In the calculation of the residuals associated with the left Cauchy-Green tensor, $\mathbf{B}_h^{i,n+\alpha}$ is directly used. The directional derivatives are calculated following [15], where $\mathbf{B}_h^{i,n+\alpha}$ is written as a function of the velocity utilizing the evolution equation of the left Cauchy-Green tensor. Hence, the Jacobian terms can be derived naturally. Furthermore, the momentum conservation equation and the evolution equation of the left Cauchy-Green tensor are decoupled with this treatment. Using the generalized- α discretization, \mathbf{B}^i (the superscript i is omitted in the

following derivations) at $t^{n+\alpha}$ can be derived as:

$$\begin{aligned}
\mathbf{B}^{n+\alpha} &= \mathbf{B}^n + \alpha(\mathbf{B}^{n+1} - \mathbf{B}^n), \\
&= \mathbf{B}^n + \alpha\Delta t(\partial_t \mathbf{B}^n + \varsigma(\partial_t \mathbf{B}^{n+1} - \partial_t \mathbf{B}^n)), \\
&= \mathbf{B}^n + \alpha\Delta t(\partial_t \mathbf{B}^n + \frac{\varsigma}{\alpha_m}(\partial_t \mathbf{B}^{n+\alpha_m} - \partial_t \mathbf{B}^n)), \\
&= \mathbf{B}^n + \alpha\Delta t \left(\left(1 - \frac{\varsigma}{\alpha_m}\right) \partial_t \mathbf{B}^n + \frac{\varsigma}{\alpha_m} \partial_t \mathbf{B}^{n+\alpha_m} \right). \tag{28}
\end{aligned}$$

Note that Eq. (28) is merely used for the derivation of the Jacobian matrix rather than the calculation of the residuals. To eliminate $\partial_t \mathbf{B}^{n+\alpha_m}$, we consider the evolution equation of the left Cauchy-Green tensor:

$$\partial_t \mathbf{B}^{n+\alpha_m} = -(\mathbf{v}^{n+\alpha} \cdot \nabla) \mathbf{B}^{n+\alpha} + \nabla \mathbf{v}^{n+\alpha} \mathbf{B}^{n+\alpha} + \mathbf{B}^{n+\alpha} (\nabla \mathbf{v}^{n+\alpha})^T. \tag{29}$$

The Jacobian terms can be calculated as:

$$\frac{\delta \mathbf{B}^{n+\alpha}}{\delta \mathbf{v}^{n+\alpha}} = \frac{\alpha \varsigma \Delta t}{\alpha_m} \left(-(\mathbf{N} \cdot \nabla) \mathbf{B}^{n+\alpha} + \nabla \mathbf{N} \mathbf{B}^{n+\alpha} + \mathbf{B}^{n+\alpha} (\nabla \mathbf{N})^T \right), \tag{30}$$

where \mathbf{N} is a vector composed of the shape functions. More details can be found in Appendix B.

2.5. Implementation details

In this subsection, we present the implementation details of our variational framework. The fully Eulerian FSI framework is decoupled and solved in a partitioned-block iterative manner which leads to flexibility and ease in its implementation to existing variational solvers. The root finding process of each block employs the Newton-Raphson method, which can be expressed in terms of the solution increments of the velocity and the pressure, the order parameters, and the left Cauchy-Green tensors ($\Delta \mathbf{u}$, Δp , $\Delta \phi^i$ and $\Delta \mathbf{B}^i$ respectively). We start with the increment of the velocity and pressure:

$$\begin{bmatrix} \mathbf{K}_\Omega & \mathbf{G}_\Omega \\ -\mathbf{G}_\Omega^T & \mathbf{C}_\Omega \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{v} \\ \Delta p \end{Bmatrix} = - \begin{Bmatrix} \overline{\mathcal{R}}_m \\ \overline{\mathcal{R}}_c \end{Bmatrix}, \tag{31}$$

where \mathbf{K}_Ω is the stiffness matrix of the momentum conservation equation consisting of transient, convection, viscous and Petrov-Galerkin stabilization terms, \mathbf{G}_Ω is the gradient operator, \mathbf{G}_Ω^T is the divergence operator for the mass conservation equation and \mathbf{C}_Ω is the stabilization term for the cross-coupling of pressure terms. $\overline{\mathcal{R}}_m$ and $\overline{\mathcal{R}}_c$ represent the weighted residuals of the variational forms of the momentum and mass conservation equations.

For fluid and solid interfaces, the order parameters are solved as:

$$[\mathbf{K}_{\phi^i}] \{ \Delta \phi^i \} = - \{ \overline{\mathcal{R}}(\phi^i) \}, \tag{32}$$

$$(33)$$

where \mathbf{K}_{ϕ^i} and $\overline{\mathcal{R}}(\phi^i)$ are the stiffness matrix and the weighted residual of the i -th Allen-Cahn equation. Lastly, we evolve of the left Cauchy-Green tensors:

$$[\mathbf{K}_{\mathbf{B}^i}] \{ \Delta \mathbf{B}^i \} = - \{ \overline{\mathcal{R}}(\mathbf{B}^i) \}, \tag{34}$$

where \mathbf{K}_{B^i} and $\overline{\mathcal{R}}(\mathbf{B}^i)$ are the stiffness matrix and the weighted residual of the i -th left Cauchy-Green tensor. This finishes one nonlinear iteration. The nonlinear iteration stops when the ratio between the L^2 norm of the increment and the current solution is less than 5×10^{-4} for all the blocks, or the nonlinear iteration number exceeds the upper limit, which is set as 20 in the current work.

Remark 1. For continua with n_f number of fluid phases and n_s number of solid objects, the current PDE system comprises a unified mass and momentum conservation equation, n_f set of equations in subsection 2.2 and n_s set of equations in subsection 2.3.

Remark 2. The left Cauchy-Green tensor is solved as a symmetry tensor with six degrees of freedom. The equation is linearized via the Newton-Raphson iteration. In forming the block matrices, the solution vector is organized as $[\Delta B_{11}, \Delta B_{22}, \Delta B_{33}, \Delta B_{12}, \Delta B_{23}, \Delta B_{13}]^T$. The subsequent linear system is solved with the GMRES method.

3. Phase-field interface processing

In the phase-field diffuse interface description, the local composition is determined by a phase indicator i.e. order parameter. The phase transition across the interface is dictated by a smooth yet highly localized variation of the order parameter. The variation is identical to give a consistent interface transition. This is achieved by a hyperbolic tangent composition of the signed distance function given by $\phi = \tanh(d/(\sqrt{2}\varepsilon))$, where d denotes the signed distance function to the interface. The distance function provides a uniform variation in the normal direction of the interface; the sign indicates the phase or not; the hyperbolic tangent composition localizes the variation close to the interface, and the scaling factor ε controls the thickness of the diffuse interface region.

When the geometry of a solid object is complex, such a field cannot be generated through simple analytical expressions. In numerical simulations, the domain and the interface are usually given in a discrete form: the computational domain is discretized by fixed Eulerian mesh, while the interface usually takes the form of a 2D interface mesh from computer-aided design software. In other words, the initialization of the order parameter field in a 3D Eulerian mesh according to a 2D interface mesh is required.

To initialize the field, we need the signed distance function d to the target interface. This can be further decomposed into two sub-problems: (i) calculate the distance function to the interface and (ii) decide the sign of the distance according to the relative position of the current location to the interface. We introduce the grid cell method [26] to restrict the calculation from the entire domain to several grid cells. A ray-casting algorithm is utilized to determine the relative position of an Eulerian mesh node to the surface mesh, and inside or outside the surface mesh indicating the phase or not.

3.1. Calculation of the distance function

We first introduce the calculation of the distance function. As discussed previously, the variation of the order parameter is highly localized. For example, for points which are 3ε away from the interface, $\phi(d = 3\varepsilon) = 0.9717$, while $\phi(d = \infty) = 1$. At the same time, the phase-field method is inherently energy stable due to the free energy minimization. Therefore, we can safely truncate the calculation of the distance at points that are 3ε away from the interface

without the concern of inaccuracy or instability, and assign values of $|\phi| = 1$, such that the distance calculation is restricted to positions which are close to the interface.

The algorithm starts by decomposing the domain into a series of grid cells. After that, we identify the grid cells that contain the interface elements and associate these interface elements with the grid cell. Lastly, the calculation of the distance is restricted to the nodes in the grid cell and the associated interface element nodes. In dividing the grid cells, we employ the approach of [26]. Denote the location of the Eulerian mesh node p as (x_p, y_p, z_p) where x, y, z are the coordinates in the X, Y and Z directions, and the subscript $p = 1, 2, \dots, n$ denotes the index of the node with n being the total number of nodes. We can define a bounding box for the Eulerian mesh $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}]$, where $x_{\min} = \min_{p=1,2,\dots,n} \{x_p\}$ and the rest values are defined similarly. If n_x, n_y, n_z grid cells are specified in each of the directions, the size of the grid cells will be $h_x = (x_{\max} - x_{\min})/n_x$. To determine the grid cell index (i, j, k) of a point (x_p, y_p, z_p) where i, j, k are grid cell indices in the X, Y and Z directions, we can use $i = \lfloor x_p/h_x \rfloor, j = \lfloor y_p/h_y \rfloor, k = \lfloor z_p/h_z \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor operator which takes the integer part of a positive real number.

Once the division is established, we need to build up the relevance between the grid cells and the surface elements. For a given surface mesh, we first decompose the surface elements as triangle elements by rewriting the connectivity. This process helps to unify the calculation for all the surface elements. For each triangle element, we create a bounding box around it and assign the element to the grid cells that intersect the bounding box. For example, for a triangular element with vertices $(x_v, y_v, z_v), v = 1, 2, 3$, we define the bounding box as $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}]$, where the minimum and the maximum are picked among the three vertices. We further expand the bounding box by 3ε , which ends up with $[x_{\min} - 3\varepsilon, x_{\max} + 3\varepsilon] \times [y_{\min} - 3\varepsilon, y_{\max} + 3\varepsilon] \times [z_{\min} - 3\varepsilon, z_{\max} + 3\varepsilon]$ so that all the nodes within 3ε distance from the surface element can be involved. The index of the corresponding grid cells can be denoted as $[i_{\min}, i_{\max}] \times [j_{\min}, j_{\max}] \times [k_{\min}, k_{\max}]$, where $i_{\min} = \lfloor (x_{\min} - 3\varepsilon)/h_x \rfloor$ and the rests are defined similarly. The surface elements are then assigned to these grid cells.

Once the grid cells and the associated nodes and surface elements are identified, we can restrict the scope of the calculation to each grid cell. For a given node belonging to the grid cell, the distance to the interface is taken as the shortest L_2 distance to the vertices of the surface elements assigned to the current grid cell. After the distance function is acquired, we compound the distance function by hyperbolic tangent function $|\phi| = \tanh(|d|/\sqrt{2}\varepsilon)$ to calculate the value of the order parameter field. For grid cells that do not contain any surface elements, we assign $|\phi| = 1$ for all Eulerian mesh nodes. All the aforementioned steps are summarized in Algorithm 1.

Algorithm 1 Distance calculation for fully Eulerian finite element framework

Allocate Eulerian mesh nodes and surface elements to the grid cells

Loop over interface grid cells

 Loop over Eulerian mesh nodes in the grid cell

 Loop over vertices of the surface elements

 Find the minimum distance $|d|$

 Assign $|\phi| = \tanh(|d|/\sqrt{2}\varepsilon)$

 Loop over grid cells without surface elements

 Assign $|\phi| = 1$ for all Eulerian mesh nodes

3.2. Determination of the sign on a node

Following the calculation of the magnitude for the order parameter value, we need to determine the sign on each node according to its relative position to the surface mesh. To be more specific, whether the point is inside or outside the given 2D interface mesh. This is a well-known topic in planar geometry problems which can be solved by the ray-casting algorithm [26, 27]. In short, we cast a semi-infinite ray from the node in the positive X direction. Then, we count how many times the ray crosses the surface. Once the ray crosses the surface, the cross number shifts between even and odd. Accordingly, we know that the point is located inside/outside the interface according to the odd/even cross number.

Since the domain has been decomposed as grid cells, we can restrict the calculation to the grid cells where the ray passes through. When the projection in the position X -direction is used, for a node (x_p, y_p, z_p) located at grid cell (i, j, k) , we only need to consider an array of grid cells $[i, n_x] \times j \times k$. In the grid cell (i, j, k) , only the surface elements with at least one vertex located at the positive X -direction of the node can be crossed by the ray. For triangle surface elements, this condition can be expressed as $\forall (x_v, y_v, z_v), v = 1, 2, 3, \exists x_v > x_p$. For grid cells $[i + 1, n_x] \times j \times k$, all the surface elements need to be considered.

After restricting the scope of computation to the node and these triangle elements, we now need to determine whether the ray crosses these triangle elements or not. We first project the node and the triangle elements to the $Y-Z$ plane of $x = x_{\max}$, $x_{\max} = \max_{p=1,2,\dots,n} \{x_p\}$. As shown in Fig. 2, we denote the area of the projected triangle from the surface element as A_0 . The areas of the three triangles, which are formed by the projected node and the projected edges of the triangle element, are denoted as A_1, A_2, A_3 , where A_i denotes the area of the triangle formed by the project node and the opposite edge of vertex $v = i$. If the point lies inside the triangle, we have $A_0 = \sum_{i=1}^3 A_i$, which indicates that the ray crosses the surface element. For triangle elements that the ray does not cross through, $A_0 < \sum_{i=1}^3 A_i$ is satisfied. This is illustrated in Fig. 2.

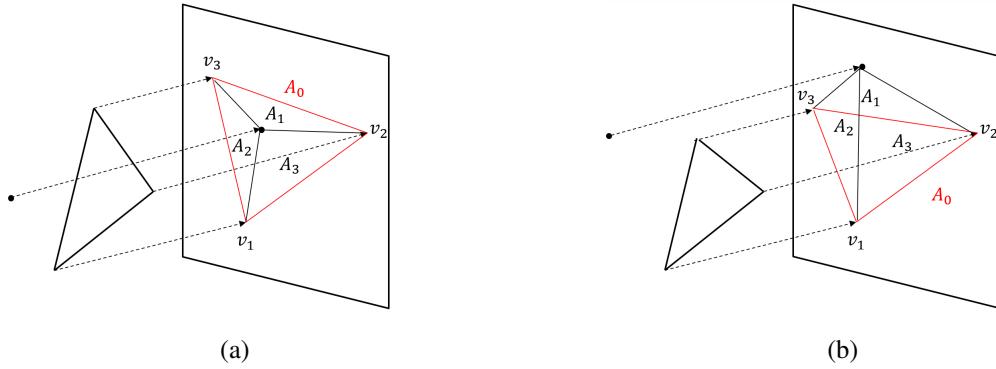


Figure 2: Illustration of ray casting algorithm: conditions for ray (a) crossing and (b) missing the surface element.

However, the criteria of $A_0 = \sum_{i=1}^3 A_i$ can be problematic in implementation. The equality may not be strictly satisfied due to the machine truncation error introduced in the area calculations. Therefore, we relax the condition as:

$$A_1 + A_2 + A_3 - A_0 \leq 2\chi, \quad (35)$$

When the condition is satisfied, we consider that the ray crosses the elements, and the cross number adds one. The relaxation introduces new errors. For example, as illustrated in Fig. 3 (a), where the projected node is close to but located outside the projected triangle element. For all positions between $\vec{v_1v_3}$ and $\vec{v_1v_2}$ where $A_1 \leq \chi$ is satisfied, $A_1 + A_2 + A_3 - A_0 = 2A_1 + A_0 - A_0 \leq 2\chi$ is true. While the ray does not cross the element, the condition is satisfied and falsely adds one to the cross number. A similar situation happens when the ray is close to the vertices.

3.3. Non-duplicated plane partitioning

To resolve this duplicate counting issue, we classify the cross into three types: element cross, edge cross and node cross. They are defined as follows:

- a) If all the areas A_1, A_2, A_3 are larger than χ , the projected node is far away from the edges, and the cross is classified as an element cross.
- b) If there are two areas that are smaller than χ , the cross is classified as a node cross.
- c) If there exists only one area that is smaller than χ , the cross is classified as an edge cross.

This classification partitions the plane into eight regions as illustrated in Fig. 3 (b). Once the cross-type is identified, we store the relevant vertex indices. Thus three, two, and one indices are recorded for element-cross, edge-cross and node-cross, respectively. The cross number adds one only for non-repeated vertex indices combinations, therefore avoiding the duplicate counting problem. Finally, the total cross number equals the summation of these three types of crosses.

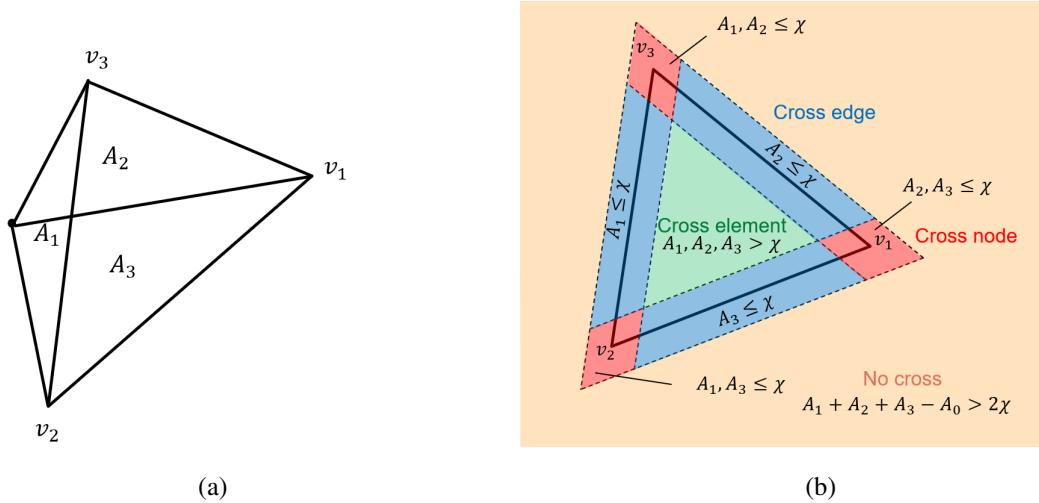


Figure 3: Illustration of the non-duplicated plane partitioning: (a) example where the relaxed condition causes duplicate counting on the cross number, (b) non-duplicated plane partitioning and the corresponding conditions.

The proposed plane partitioning reduces the duplicate counts arising from discrete surface mesh to a non-duplicated count, therefore referred to as non-duplicated plane partitioning. For grid cells that do not contain any surface element, all the nodes inside the grid cell lie on the same side of the interface. In such cases, we only judge the sign for a single node in the grid cell, then assign this sign for all the rest nodes inside the grid cell. The procedures to determine the sign on each Eulerian mesh node are summarized in Algorithm 2.

Algorithm 2 Determination of signs on Eulerian finite element mesh nodes

```
Loop over interface grid cells
  Loop over Eulerian mesh nodes in the grid cell
    Loop over surface elements in grid cells ([i, imax], j, k)
      Count cross number using ray-casting algorithm and NDP partitioning
      Determine the sign according to the cross number
  Loop over grid cells without surface elements
    Pick one Eulerian mesh nodes in the grid cell
    Loop over surface elements in grid cells ([i, imax], j, k)
      Count cross number using ray-casting algorithm and NDP partitioning
    Determine the sign according to the cross number
    Assign the sign to all nodes in the grid cell
```

4. Test cases for phase-field interface processing

In this section, we examine the phase-field interface processing with multiple test cases. We first verify the calculation of the signed distance function in a single grid cell for sphere and cubic interfaces to establish the order of convergence. Then, we test the algorithm for multiple grid cells with the cubic interface case. Finally, we demonstrate the algorithm for the complex geometry of a ship hull.

4.1. Interface meshes for sphere and cube

We start by verifying the calculation of the signed distance function. A domain of $[0, 1] \times [0, 1] \times [0, 1]$ is considered in this problem. A sphere interface with a radius of $r = 0.25$ is centered at $(0.5, 0.5, 0.5)$. The sign is defined as positive inside the sphere interface. A single grid cell is used for the calculation so that the signed distance is calculated throughout the domain. This helps to clarify the convergence of the signed distance function calculation.

The computational domain is discretized with uniform structured cubic elements with an edge length of $h_E = 0.02$ and $h_E = 0.01$, where E denotes the Eulerian background mesh. The sphere interface is discretized with unstructured triangular mesh, whose edge length is denoted as h_S . The interface mesh is shown in Fig. 4 (a). We vary the ratio of the surface mesh size to the Eulerian background mesh size from $h_S/h_E = 2$ to $h_S/h_E = 0.5$. The error is quantified as $e_2 = \|\phi - \phi_{ref}\|/\|\phi_{ref}\|$, where $\|\cdot\|$ denotes the L_2 norm. The results are shown in Fig. 5 (a), where a second-order convergence is observed. We further test the algorithm with a cubic interface mesh, which contains sharp edges and corners. A cubic interface mesh spans through $[0.25, 0.75] \times [0.25, 0.75] \times [0.25, 0.75]$ is used as shown in Fig. 4 (b), the rest of the numerical setup is kept the same. The second-order convergence is confirmed in Fig. 5 (b).

4.2. Calculation with multiple grid cells

We then examine the algorithm with multiple grid cells. The cubic interface mesh case discussed in subsection 4.1 is used. The domain is discretized with a uniform structured mesh of size $h_E = 0.02$, and an unstructured triangle mesh of size $h_S = 0.005$ is used for the cubic interface. The domain is decomposed as 9 segments in the X , Y and Z directions, leading to 729 grid cells. The slice of Y - Z plane at $x = 0.5$ is shown in Fig. 6 (a). As observed, the calculation of the signed distance function is restricted to grid cells that are close to the interface. In the grid cells that are far from the interface, the distance calculation is omitted,

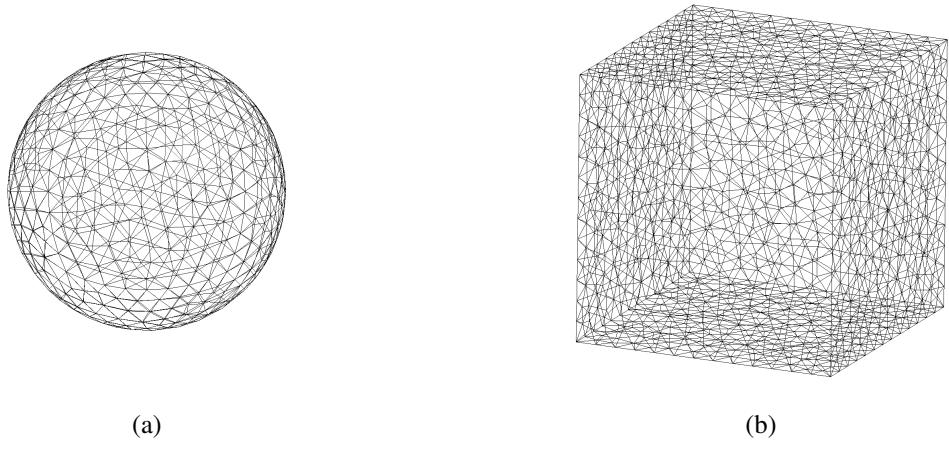


Figure 4: Illustration of the interface meshes for two simple geometries: (a) sphere and (b) cube.

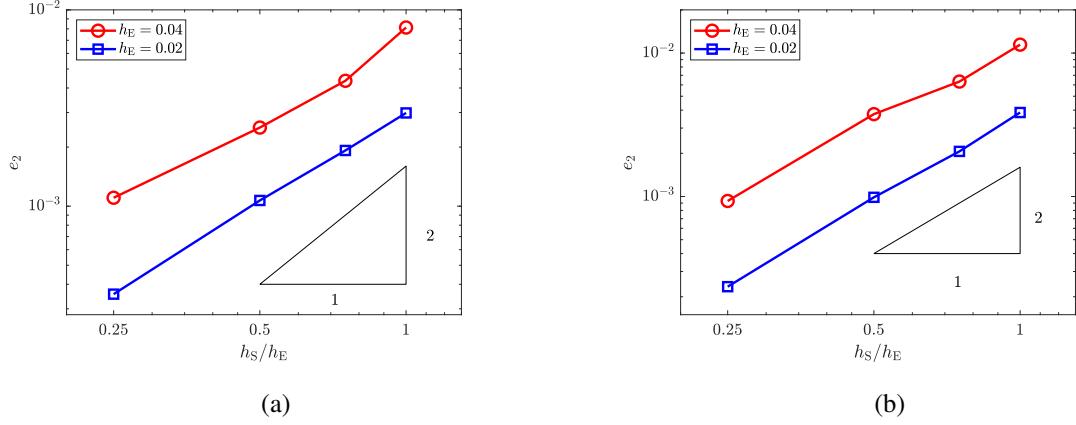


Figure 5: Convergence of the signed distance function for two representative meshes: (a) sphere and (b) cube.

and the values of 1 and -1 are assigned to the order parameter for nodes located inside and outside of the interface, respectively. The last step is to compound the signed distance function with the hyperbolic tangent function. As a result, an order parameter field in which the variation is highly localized to the interface is generated, as shown in Fig. 6 (b).

4.3. Demonstration with a ship hull

Lastly, we demonstrate the algorithm for interfaces with complex geometry. We employ the vessel STAN TUG 1004 for which the vessel geometry is available online¹. The domain and the interface are discretized with unstructured meshes of size $h_E = h_S = 0.02$. The interface mesh is shown in Fig. 7 (a), while the isosurface of $\phi = 0$ of the generated order parameter field is shown in Fig. 7 (b). As observed, the current algorithm can successfully generate the order parameter field for interfaces with complex geometry. The generated field will be further used as the initial condition for demonstrating the ship-ice interaction problem.

¹<https://grabcad.com/library/stan-tug-1004-1>

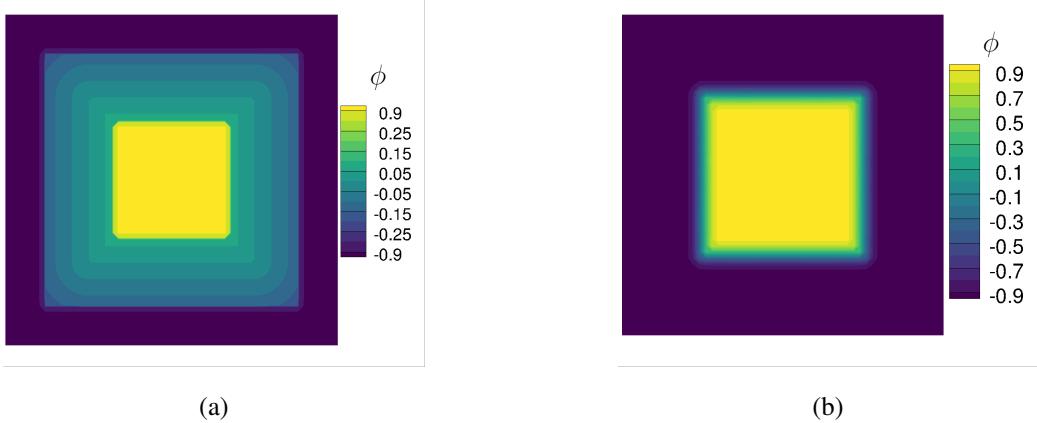


Figure 6: Illustration of contours of (a) signed distance function, and (b) order parameter field calculated with multiple grid cells.

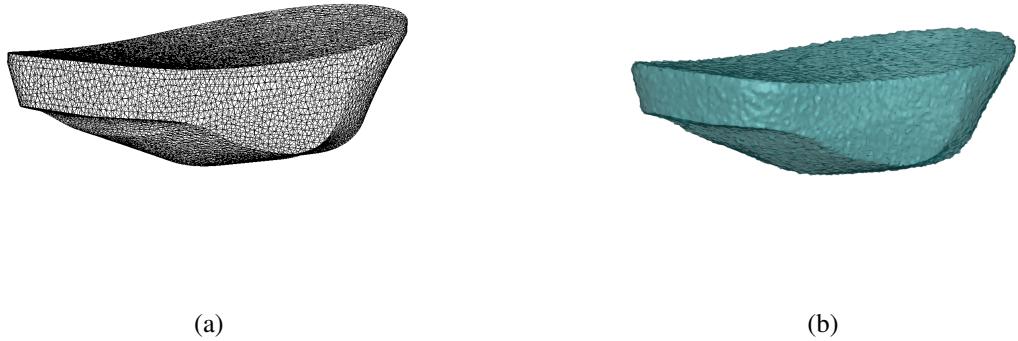


Figure 7: An illustration of a tugboat represented by (a) interface mesh and (b) isosurface of $\phi = 0$.

5. Test cases for multiphase fluid-solid interaction

In this section, we validate and demonstrate the framework with cases of increasing complexity. We start by verifying the framework with a rotational disk in a lid-driven cavity flow. Then, we showcase the solver with a 3D case of a rotating sphere subject to the lid-driven cavity flow. Finally, the contact between solids is demonstrated with an immersed sphere falling on an elastic block under gravitational force.

5.1. Rotating disk in lid-driven cavity flow

In this section, we examine the fluid-solid interaction aspect of our solver in the case of a rotating disk in lid-driven cavity flow. Consider the computational domain of $[0, 1] \times [0, 1]$. A circular disk with radius $r = 0.2$ is centered at $(0.6, 0.5)$. The no-slip boundary condition is applied to the left, right, and bottom boundaries, with a prescribed velocity at the top boundary as $v_x = 1$. The order parameter and the left Cauchy-Green tensor on the boundaries are evolved according to the prescribed velocity. The densities of the solid and fluid are selected as $\rho^s = \rho^f = 1$. The dynamic viscosity is taken as $\mu^s = \mu^f = 0.01$. The shear modulus of the solid is chosen as $\mu_L^s = 0.1$. The problem setup is illustrated in Fig. 8.

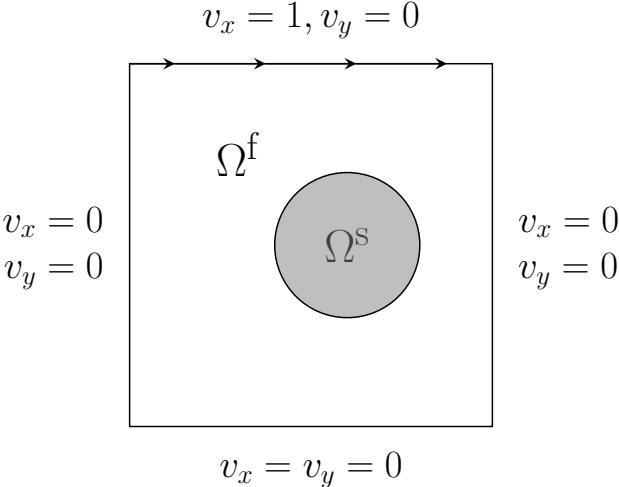


Figure 8: Schematic diagram of a rotating disk in lid-driven cavity flow.

We first perform a time convergence study. The interface parameters are selected as $\eta = 0.2, \varepsilon = 0.02$. A uniform structured triangle mesh with edge length $h = 0.02$ is used to discretize the computational domain. The time steps are bisected from $\Delta t = 0.004$ to $\Delta t = 0.001$, and the case is simulated until $t = 20$. The center of mass of the solid is defined as $\mathbf{x}_c = \int \mathbf{x} d\Omega^s / \int 1 d\Omega^s$, where the solid domain is identified through the corresponding order parameter $d\Omega^s = \alpha^s(\phi^s) d\Omega$. The trajectories of the center of mass with different time steps are plotted in Fig. 9 (a). As observed, the solution has converged at $\Delta t = 0.001$.

After that, we perform an interface convergence study[13]. With the converged time step $\Delta t = 0.001$, we bisect η and ε simultaneously from $\eta = 0.2, \varepsilon = 0.02$ to $\eta = 0.05, \varepsilon = 0.005$. The meshes are refined according to maintain the interface resolution at $h/\varepsilon = 1$. The trajectories of the center of mass are plotted in Fig. 9 (b) and compared with the reference in [28]. As shown by the figure, the solution matches well with the reference solution.

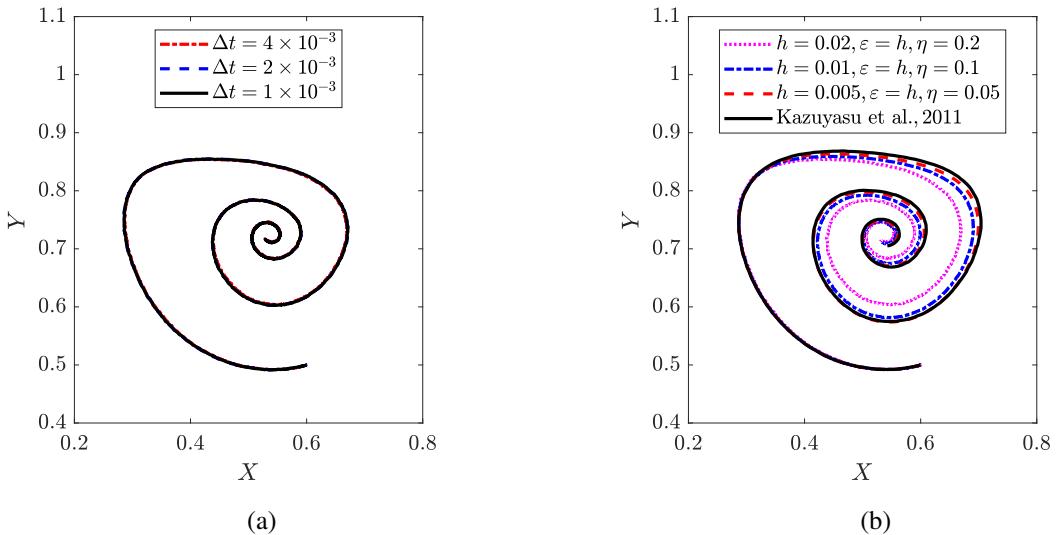


Figure 9: Rotating disk in lid-driven cavity flow: (a) temporal and (b) interface convergence study for the center of mass of the disk.

5.2. Rotating sphere in lid-driven cavity flow

We generalize the rotating disk case as a rotating sphere case to demonstrate the 3D aspect of the solver. A computational domain of $[0, 1] \times [0, 1] \times [0, 1]$ is considered. A solid sphere of radius $r = 0.2$ is initially centered at $(0.6, 0.5, 0.5)$. The no-slip boundary condition is applied to the side and bottom walls, while a constant velocity of $v_x = 1$ is applied at the top boundary. The order parameter and the left Cauchy-Green tensor on the boundaries are evolved according to the prescribed velocity. The densities of the fluid and the solid are chosen as $\rho^s = \rho^f = 1$. The dynamic viscosity is taken as $\mu^s = \mu^f = 0.02$. The shear modulus of the solid is selected as $\mu_L^s = 0.5$. The case setup is illustrated in Fig. 10.

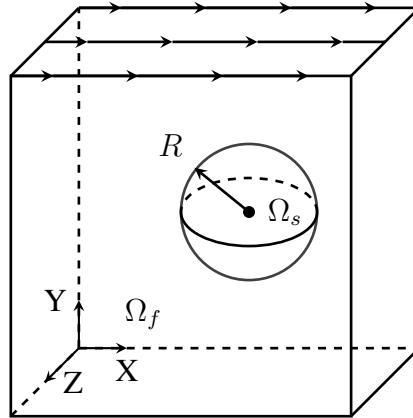


Figure 10: Schematic of a rotating sphere in lid-driven cavity flow.

Similarly, we perform a temporal and interface convergence study. The interface parameters are selected as $\eta = 0.2, \varepsilon = 0.02$. The time steps are bisected from $\Delta t = 0.004$ to $\Delta t = 0.001$, and the case is simulated until $t = 10$. The trajectory of the center of mass on the $X - Y$ plane is plotted in Fig. 11 (a). As shown by the figure, the solution has converged at $\Delta t = 0.001$ with minor variance. Therefore, we use $\Delta t = 0.004$ for the spatial convergence study. With $\Delta t = 0.004$, we perform an interface convergence study. The interface parameters η and ε are scaled with $1/\sqrt{2}$ simultaneously from $\eta = 0.2, \varepsilon = 0.02$ to $\eta = 0.1, \varepsilon = 0.01$. The meshes are refined according to maintain the interface resolution at $h/\varepsilon = 1$. The trajectories of the center of mass are plotted in Fig. 11 (b). As observed, the solution has converged in the most refined case. To better visualize the field, we show the interface of $\phi = 0$ and Z -vorticity at the slice of $z = 0.5$ at $t = 1, 3, 4, 5, 6, 8$ in Fig. 12. As observed, the sphere rotates in the clockwise direction driven by the cavity flow. It touches the wall and experiences large deformation at $t = 4$, and keeps rotating afterward at the upper mid part of the domain, which is reflected by a uniform negative Z -vorticity. As demonstrated in the case, the arbitrary rotational motion of an immersed elastic body can be naturally handled in the current framework.

5.3. Sphere falling on an elastic block

After examining the 3D single-phase FSI, we next look into the contact of immersed solids in a fluid environment. A computational domain of $[0, 1] \times [0, 1] \times [0, 1]$ is considered. A solid sphere centered at $(0.5, 0.75, 0.5)$ at a radius of $r = 0.15$ is initially hanging above a solid block filling the computational domain below $y = 0.25$ and falls under the gravitational acceleration of $\mathbf{g} = (0, -0.98, 0)$. The no-slip boundary conditions are applied to the side and bottom walls, while an outflow boundary condition is applied to the top boundary. The order parameter and

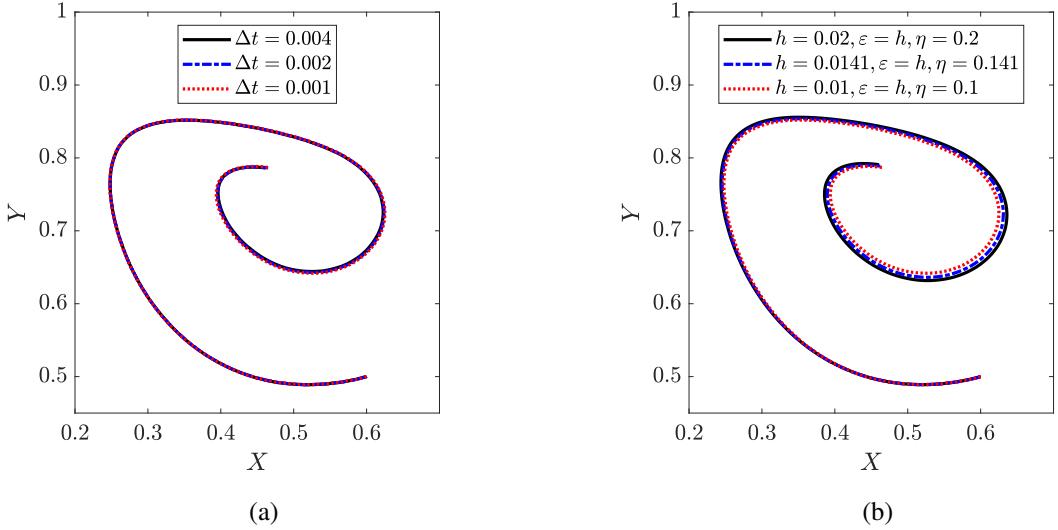


Figure 11: Rotating sphere in lid-driven cavity flow: (a) temporal and (b) interface convergence study for the center of mass of the sphere in $X - Z$ plane.

the left Cauchy-Green tensor on the boundaries are evolved according to the local velocity. The densities of the solid sphere, the solid block and the surrounding fluid are chosen as $\rho^s = 5000$, $\rho^b = 5000$, $\rho^f = 1000$. The viscosities of the solids are set to be zero while the fluid viscosity is selected as $\mu^f = 1$. The shear modulus of the solid sphere and the solid block are taken as $\mu_L^s = 1000$ and $\mu_L^b = 1000$. The problem setup is illustrated in Fig. 13.

In the computational framework, we set one phase for the falling sphere, one for the solid block, and one for the immersed fluid. The computational domain is discretized with tetrahedron elements of size $h = 0.01$. The diffuse interface thickness is chosen as $\varepsilon = h$ and $\eta = 0.1$ is used. The time step is selected as $\Delta t = 0.005$. The case is simulated until $t = 7.5$. The simulation results are visualized at $t = 1, 1.5, 2.5, 3.25, 4, 7.5$ in Fig. 13. As observed, the sphere starts falling due to gravitational force. It reaches the lowest and highest positions at $t = 1, 1.5, 2.5, 3.25, 4, 7.5$ and gradually settles down as depicted at $t = 7.5$. To better monitor the gap during contact between the sphere and the block, we track the trajectory of the bottom of the sphere and the top center of the solid block. To avoid the influence from the diffuse interface modeling, we take a 2ε shift towards the internal of the solids. As a result, the monitor points are initially located at $(0.5, 0.6+2\varepsilon, 0.5)$ and $(0.5, 0.25-2\varepsilon, 0.5)$. The points are evolved by a parallelized particle tracking algorithm, which is presented in Appendix C. The zoomed-in view of the contact interface at $t = 7.5$ and the time history of Y coordinates of the monitor points are plotted in Fig. 15 (a) and (b), respectively. As observed, the no-penetration condition in the process of contact is naturally satisfied. As demonstrated in this case, large translational motion and contact between solids can be naturally handled by the current framework.

6. Application to ship-ice interaction

Of particular interest is the ship-ice interaction in the Arctic region. Located at the north pole and surrounded by lands, the Arctic region encompassing various terrains, enormous biodiversity and rich resources, is now the home to indigenous people of over 40 communities and a total population of four million [29, 30, 31]. Under the effect of global warming, the Arctic Ocean may be seasonally ice-free by 2050 [32]. This creates opportunities for extracting natu-

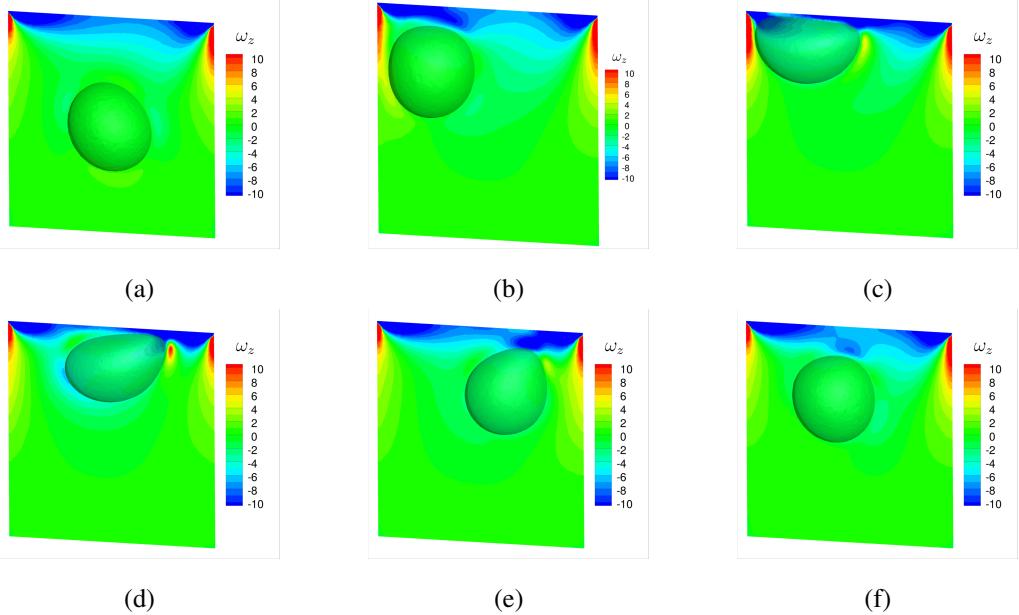


Figure 12: Rotating sphere in lid-driven cavity flow: the evolution of the interface $\phi = 0$ and the Z -vorticity at different time instants $t =$ (a) 1, (b) 3, (c) 4, (d) 5, (e) 6, (f) 8.

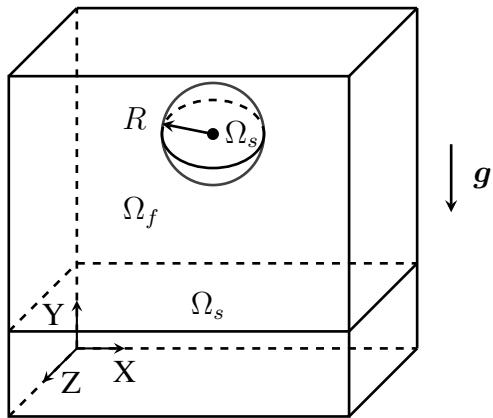


Figure 13: Schematic of an elastic sphere falling on a deformable solid block.

ral resources and finding new routes for transcontinental transportation [33]. With the growing strategic importance of Arctic transportation, it is required to design and manufacture marine vessels that are reliable and safe when sailing across ocean environments with ice sheets and floes. High-fidelity simulation is of great importance to understanding the physical process involved and aiding the design and optimization processes.

The problem of ship-ice interaction involves interactions between ice mechanics, ship structure, hydrodynamics, and wind flow. The ice floating on the sea level can vary in size and geometry: from small-scale ice rubble to large-scale ice ridges and floes [34]. During the sailing of ships, they are in contact with the ship hull, accelerated, pushed aside, or submerged, which creates loads and deformation on both sides. Then, as a source of disturbance, the moving ice can interact with surrounding static ice and cause drifting and rotating motions. The ice and ship hull interaction happens near the water-air interface under the gravitational force. As the ship moves forward, the wave runs up and entrains air bubbles inside. The bubbles then break

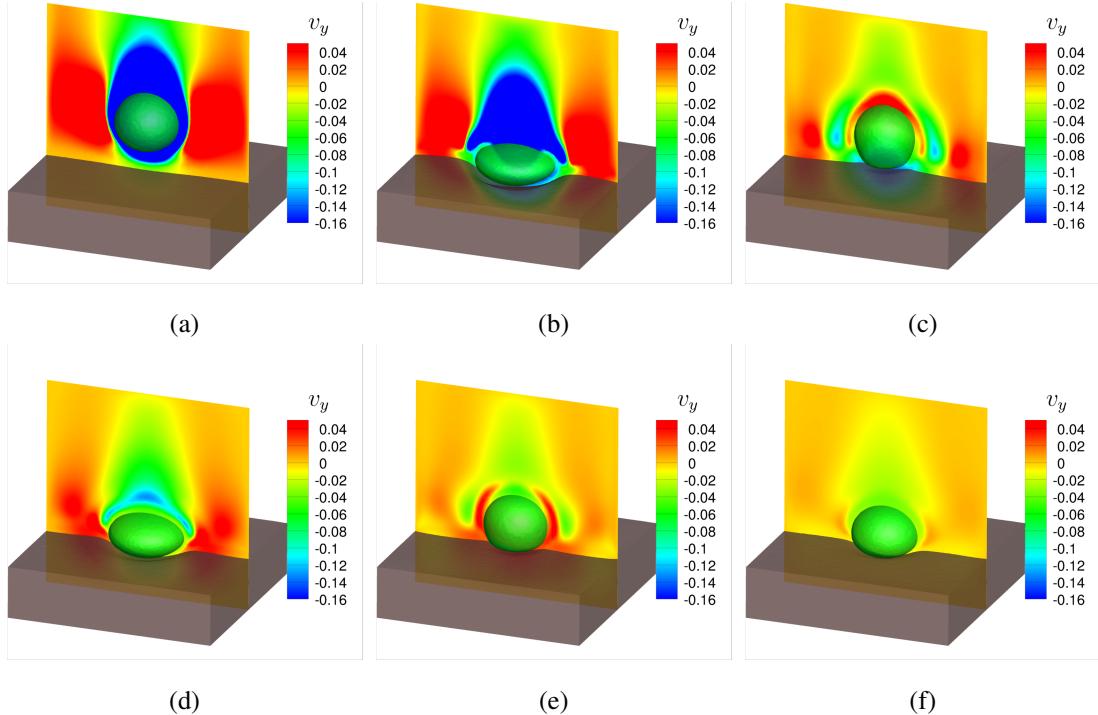


Figure 14: An elastic sphere falling on a deformable solid block: the evolution of the interfaces $\phi = 0$ and v_y contour at $Z = 0.5$ at different time instants $t =$ (a) 1, (b) 1.5, (c) 2.5, (d) 3.25, (e) 4, (f) 7.5.

apart and become sea foam. The sea foam propagates along with waves surrounding the ship and interacts with floating ice and ship hull [35], as illustrated in Fig. 16. They are all coupled and interact through interfaces of different types, forming a highly complex nonlinear dynamic system of multiphase and multi-physics nature, resulting in both the global resistance and local load on the ship. They play important roles in different ways: the global resistance affects the operation of a ship, such as maneuverability and fuel consumption, while the local load is critical for the safety of the ship structure.

While research efforts have been devoted to understanding the ice-ship interaction, the current state-of-the-art methods for calculating coupled hydrodynamics-structure-ice interactions are semi-empirical and highly phenomenological[4]. Using our multiphase FSI framework, for the first time, we will demonstrate ship-ice interaction with nonlinear hydrodynamics effects. With first-principle-based continuum mechanics, the simulation captures relevant physics completely, including the interactions between floating solids and surrounding free surface, and contact between the structures. This is crucial to understand the dynamics involved in the process, and can be utilized to provide better resistance and extreme load prediction.

In the demonstration case, we employ a vessel STAN TUG 1004 for which the vessel geometry is available online². The domain of $[-1, 1] \times [-2, 2] \times [-7.5, 2]$ is considered. The free water level is the $X - Y$ plane of $x = 0$. The leading point of the cross-plane of the ship and the free water level is placed at $(0, 0, -1.1157)$. Six pieces of ice floes in total are posed in front of the boat. They are formed by extruding the cross section from $x = 0$ to $x = -0.3$. The geometries of cross sections of the ice floes are listed in Table 1.

²<https://grabcad.com/library/stan-tug-1004-1>

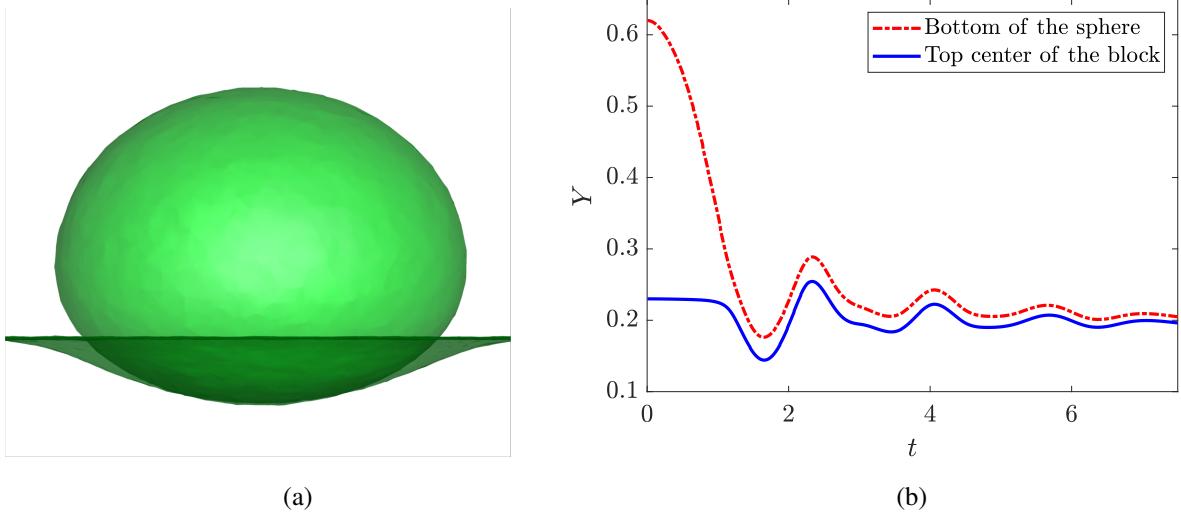


Figure 15: Sphere falling on a solid block: (a) Interfaces of solids $\phi = 0$ at $t = 7.5$, and (b) trajectories of the bottom of the sphere and the top center point of the block

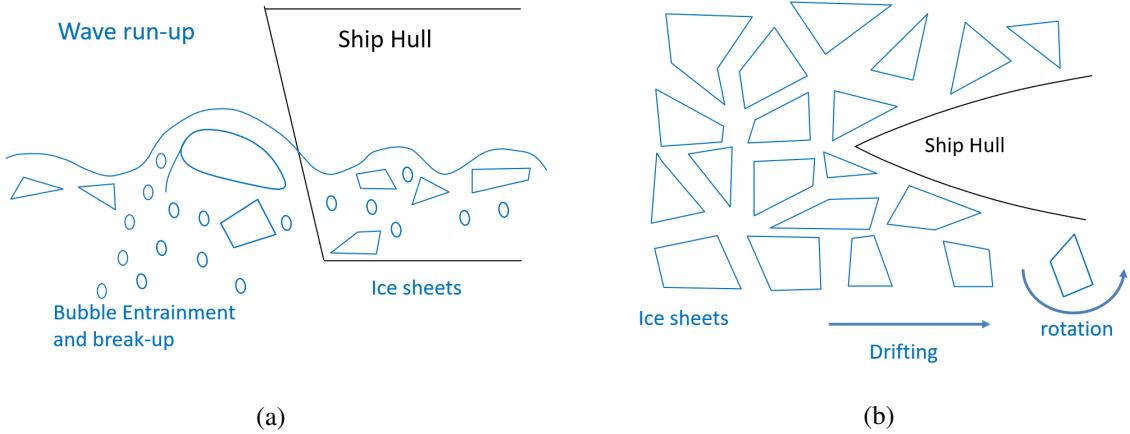


Figure 16: Illustration of ship-ice interaction showing combined effects of hydrodynamics and ice floes interacting with a ship hull: (b) side view and (c) top view.

The boat accelerates from stationary to $v_z = -1$ sinusoidally: $v_z = -\sin(2/5\pi t), t < 1.25, v_z = -1, t \geq 1.25$. This is realized by finding the nodes inside the boat and specifying the velocity on these nodes. The nodes are identified by $-0.2 - d(t) \leq z \leq 0.2 - d(t), |y| \leq 0.2, |x| \leq 0.2$, where $d(t)$ is the moving distance of the ship integrated from the given velocity v_z . The slip boundary condition is applied to side boundaries. The no-slip boundary condition is applied on the bottom boundary, and the outlet boundary condition is used for the top boundary. The zero Neumann boundary condition is applied for all the order parameters and left Cauchy-Green tensors on all the boundaries. The physical properties are chosen such that the overall behavior is captured while the computation is eased for demonstration purposes. Densities of the ship, ice, water, and air phases are selected as $\rho^s = 600, \rho^i = 800, \rho^w = 1000, \rho^a = 100$. The viscosities of the phases are selected as $\mu^s = \mu^i = \mu^w = \mu^a = 1$ to stabilize the numerical simulation. The shear modulus of the ship and the ice floes are selected as $\mu_L^s = \mu_L^i = 1 \times 10^6$ to mimic the nearly rigid contact. The gravitational acceleration is chosen as $\mathbf{g} = (-0.98, 0, 0)$ to amplify the free surface motion. The case is simulated until $t = 4$. The case setup is illustrated

Table 1: Geometry of the ice floes

Index	shape of cross section	coordinate of cross sections in (z, y)
1	triangle	$(-1.1, 0.6), (-1.7, 0.2), (-1.7, 0.9)$
2	quadrilateral	$(-1.6, -0.2), (-2.1, -0.6), (-1.8, -1), (-1, -0.6)$
3	quadrilateral	$(-3, 1.3), (-3.7, 1), (-3.2, 0.3), (-2.7, 0.7)$
4	quadrilateral	$(-4, -0.4), (-4.4, -0.9), (-3.8, -1.7), (-3.2, -0.7)$
5	quadrilateral	$(-4.8, 1), (-5.4, 0.6), (-4.7, 0.4), (-4.1, 0.5)$
6	quadrilateral	$(-5.3, -0.4), (-6, -0.7), (-5.9, -1.2), (-5.1, -1)$

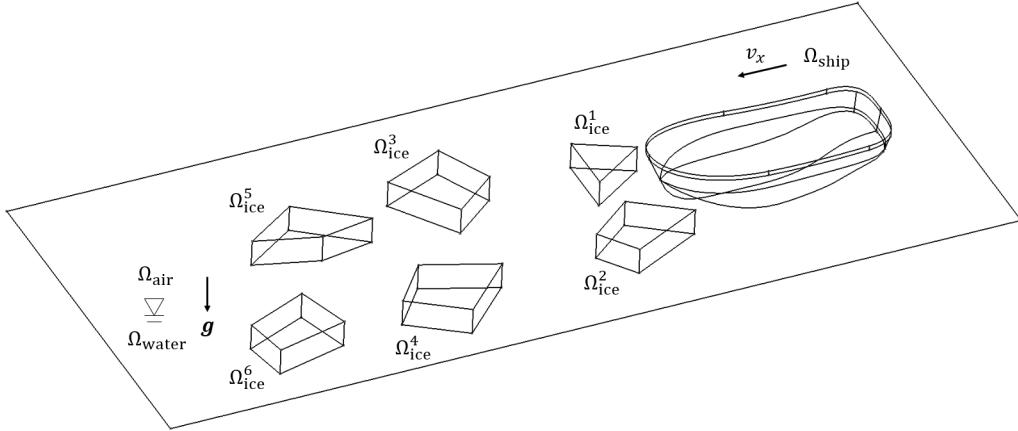


Figure 17: Schematic diagram of an ice-going ship with free surface and ice floes.

in Fig. 17.

In the computational framework, we set one fluid phase for water, one fluid phase for air, one solid phase for the ship, and one solid phase for all the ice floes. The computational domain is discretized with tetrahedron elements of size $h = 0.02$ around the interacting phases. The diffuse interface thickness is chosen as $\varepsilon = h$. With the focus on the geometry of the solids, we set $\eta = 1000$. The time step is selected as $\Delta t = 0.005$. The simulation results are visualized at $t = 0.25, 1, 2, 2.5, 3, 4$ in Fig. 18. As observed, the rich physics of interactions between the ship and free surface, the ship-ice interaction, and the coupling of ice floes and free surface are captured. An interesting dynamics of ship-ice interaction at the bow area can be observed qualitatively. It is worth mentioning that the ice mechanics model is very simplified without considering detailed aspects of constitutive modeling and fracture mechanics. Further parametric investigations and validation should be carried out for more realistic modeling of ice mechanics for the ship-ice interaction with nonlinear hydrodynamics effects.

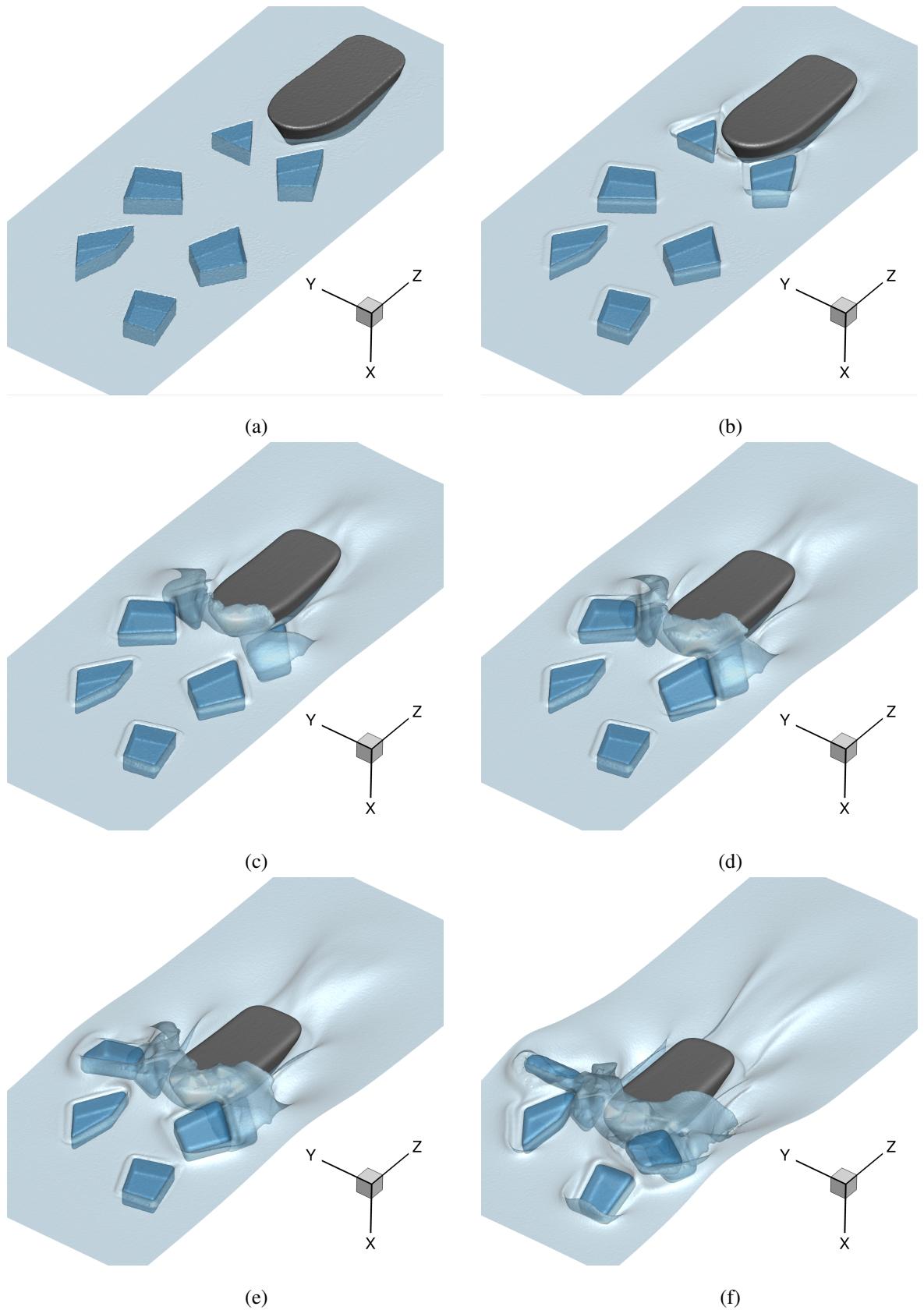


Figure 18: Representative results of ship-ice interaction: superposition of the interfaces $\phi^i = 0$ at different time instants $t =$ (a) 0.25, (b) 1, (c) 2, (d) 2.5, (e) 3, (f) 4.

7. Conclusion

In this paper, we presented a novel unified Eulerian variational framework for multiphase FSI problems based on the phase-field method. By employing the phase-field method on a fixed mesh, the proposed fully Eulerian framework offers numerous attractive characteristics to simulate multiphase fluid-structure interactions with solid-solid contact. Some of the key features of the proposed computational framework are:

- Phase-field interface processing based on grid cell method, ray-casting algorithm and non-duplicated plane partitioning for order parameter initialization of complex interface geometries;
- Unified momentum and mass conservation equations for the dynamics of general multiphase FSI problems;
- Improved accuracy of interface representation with interface-preserving phase-field model;
- High robustness in handling interface motion, including large translation/rotational motion, interface merging/breaking-up and solid-solid contact interactions;
- 3D parallelized implementation for geometric preprocessing of phase-field method and fully-Eulerian finite element simulation.

We demonstrated the versatility and validity of the proposed framework for the increasing complexity of problems. To begin, we verified the framework against reference results with a 2D rotational disk case and generalized the case to the 3D rotational sphere. The cases showed that rotational motion and large deformation can be handled easily in the current framework. Next, we simulated an immersed sphere falling on an elastic block to demonstrate the capability to deal with translational motion and contact between solids. As observed, no-penetration contact can be naturally handled by the framework. Finally, we solved the ship-ice interaction problem involving free surface between air and water and contact between ship and ice floes. The hydraulic effects and contact between solids are faithfully captured in the case. This application demonstrates the capability of the proposed framework to simulate practical multi-physics/multiphase systems with complex structures. The framework allows the incorporation of additional physical fields and any number of solid objects thus enabling a valuable tool for analyzing a broad range of industrial applications.

Acknowledgement

The authors would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) and Seaspan Shipyards for the funding. This research was supported in part through computational resources and services provided by Advanced Research Computing at the University of British Columbia.

Appendix A. Parallel implementation for L^2 -projection

In the implementation of a time-dependent mobility model, we need to calculate the first-order derivatives of velocities and order parameters at nodes. A shape function based weighted

average is used to achieve this, which is referred to as the L^2 -projection. It projects the first-order derivatives from quadrature points to nodes. For node point p and the shape function on this point N_p , the formulation is given as:

$$(\nabla \varphi)_p = \frac{\int N_p \nabla \varphi d\Omega}{\int N_p d\Omega} \quad (\text{A.1})$$

This step takes the information from all directions for nodes inside the domain. However, the information can only come from a single side for nodes located at boundaries. This will cause inconsistency in the order of accuracy.

For example, consider the domain of $x \in [0, 2]$ with three nodes $x = 0, 1, 2$. Denote the domain of $x \in [0, 1]$ as Ω_L and $x \in [1, 2]$ as Ω_R . Piece-wise linear shape functions are used for Galerkin projection. We test the L^2 -projection for the function of $\varphi = x^2$, the analytical solution of the first order derivative is $\frac{\partial \varphi}{\partial x} = 2x$, which is a linear function. When multiplied with linear shape function at the numerator of Eq. A.1, the integrated function is a second-order polynomial, which can be calculated exactly with first-order quadrature. As a result, the first-order derivative values calculated from the L^2 -projection are:

$$\left. \frac{\partial \varphi}{\partial x} \right|_{x=0} = \frac{\int (1-x) 2x d\Omega_L}{\int (1-x) d\Omega_L} \approx \frac{\sum_{qp=1}^2 w_{qp}(1-x_{qp}) 2x_{qp}}{\sum_{qp=1}^2 w_{qp}(1-x_{qp})} \approx 0.667 \quad (\text{A.2})$$

$$\left. \frac{\partial \varphi}{\partial x} \right|_{x=1} = \frac{\int (x) 2x d\Omega_L + \int (2-x) 2x d\Omega_R}{\int (x) d\Omega_L + \int (2-x) d\Omega_R} \approx \frac{\sum_{qp=1}^2 w_{qp}(x_{qp}) 2x_{qp} + \sum_{qp=1}^2 w_{qp}(2-x_{qp}) 2x_{qp}}{\sum_{qp=1}^2 w_{qp}(x_{qp}) + \sum_{qp=1}^2 w_{qp}(2-x_{qp})} \approx 2 \quad (\text{A.3})$$

$$\left. \frac{\partial \varphi}{\partial x} \right|_{x=2} = \frac{\int (x-1) 2x d\Omega_R}{\int (x-1) d\Omega_R} \approx \frac{\sum_{qp=1}^2 w_{qp}(x_{qp}-1) 2x_{qp}}{\sum_{qp=1}^2 w_{qp}(x_{qp}-1)} \approx 3.333 \quad (\text{A.4})$$

As observed, while the result at the interior node $x = 1$ is identical with the exact value, errors are introduced at the boundary points of $x = 0$ and $x = 2$. This can be further clarified by expanding the domain to $x \in [-1, 3]$. Following the same procedures, the first order derivatives can be calculated as: $\left. \frac{\partial \varphi}{\partial x} \right|_{x=-1} = -1.333$, $\left. \frac{\partial \varphi}{\partial x} \right|_{x=0} = 0$, $\left. \frac{\partial \varphi}{\partial x} \right|_{x=1} = 2$, $\left. \frac{\partial \varphi}{\partial x} \right|_{x=2} = 4$, $\left. \frac{\partial \varphi}{\partial x} \right|_{x=3} = 5.333$. As we can see, the error only occurs at the boundaries of the domain.

When the L^2 -projection is implemented in parallel, the computational grid will be decomposed as multiple mesh blocks. At the boundaries of the mesh blocks, the error mentioned in previous paragraphs is introduced. To fix this issue, we first calculate the numerator $\int N_p \nabla \varphi d\Omega$ and the denominator $\int N_p d\Omega$ of Eq. A.1 at each mesh block. Then, we add the values at the mesh block boundary nodes together to take information from all directions. After this information exchange, we perform a node-wise division to complete the L^2 -projection.

Appendix B. Jacobian terms of solid stress in Eulerian frame

In this appendix, we present a 2D component form of the Jacobian terms of the solid stress in the momentum conservation equation with a focus on the shear components. We start from the convection of the left Cauchy-Green tensor:

$$\begin{aligned}\frac{\partial B_{11}}{\partial t} &= \left(-v_x \frac{\partial B_{11}}{\partial x} - v_y \frac{\partial B_{11}}{\partial y} + \left(\frac{\partial v_x}{\partial x} B_{11} + \frac{\partial v_x}{\partial y} B_{21} \right) + \left(B_{11} \frac{\partial v_x}{\partial x} + B_{12} \frac{\partial v_x}{\partial y} \right) \right), \\ \frac{\partial B_{12}}{\partial t} &= \left(-v_x \frac{\partial B_{12}}{\partial x} - v_y \frac{\partial B_{12}}{\partial y} + \left(\frac{\partial v_x}{\partial x} B_{12} + \frac{\partial v_x}{\partial y} B_{22} \right) + \left(B_{11} \frac{\partial v_y}{\partial x} + B_{12} \frac{\partial v_y}{\partial y} \right) \right), \\ \frac{\partial B_{21}}{\partial t} &= \left(-v_x \frac{\partial B_{21}}{\partial x} - v_y \frac{\partial B_{21}}{\partial y} + \left(\frac{\partial v_y}{\partial x} B_{11} + \frac{\partial v_y}{\partial y} B_{21} \right) + \left(B_{21} \frac{\partial v_x}{\partial x} + B_{22} \frac{\partial v_x}{\partial y} \right) \right), \\ \frac{\partial B_{22}}{\partial t} &= \left(-v_x \frac{\partial B_{22}}{\partial x} - v_y \frac{\partial B_{22}}{\partial y} + \left(\frac{\partial v_y}{\partial x} B_{12} + \frac{\partial v_y}{\partial y} B_{22} \right) + \left(B_{21} \frac{\partial v_y}{\partial x} + B_{22} \frac{\partial v_y}{\partial y} \right) \right).\end{aligned}$$

To keep the symmetry of the left Cauchy-Green tensor, we let $B_{21} = B_{12}$ in the practical implementation. Using the generalized- α method, $\mathbf{B}^{n+\alpha}$ can be written as a function of $\mathbf{v}^{n+\alpha}$ as follows:

$$\begin{aligned}B_{11}^{n+\alpha} &= B_{11}^n + \alpha \Delta t \left(1 - \frac{\gamma}{\alpha_m} \right) \partial_t B_{11}^n \\ &\quad + \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-v_x^{n+\alpha} \frac{\partial B_{11}^{n+\alpha}}{\partial x} - v_y^{n+\alpha} \frac{\partial B_{11}^{n+\alpha}}{\partial y} + 2 \frac{\partial v_x^{n+\alpha}}{\partial x} B_{11}^{n+\alpha} + 2 \frac{\partial v_x^{n+\alpha}}{\partial y} B_{12}^{n+\alpha} \right), \\ B_{12}^{n+\alpha} &= B_{12}^n + \alpha \Delta t \left(1 - \frac{\gamma}{\alpha_m} \right) \partial_t B_{12}^n \\ &\quad + \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-v_x^{n+\alpha} \frac{\partial B_{12}^{n+\alpha}}{\partial x} - v_y^{n+\alpha} \frac{\partial B_{12}^{n+\alpha}}{\partial y} \right) \\ &\quad + \frac{\Delta t \alpha \gamma}{\alpha_m} \left(\frac{\partial v_x^{n+\alpha}}{\partial x} B_{12}^{n+\alpha} + \frac{\partial v_x^{n+\alpha}}{\partial y} B_{22}^{n+\alpha} + \frac{\partial v_y^{n+\alpha}}{\partial x} B_{11}^{n+\alpha} + \frac{\partial v_y^{n+\alpha}}{\partial y} B_{12}^{n+\alpha} \right), \\ B_{21}^{n+\alpha} &= B_{12}^{n+\alpha}, \\ B_{22}^{n+\alpha} &= B_{22}^n + \alpha \Delta t \left(1 - \frac{\gamma}{\alpha_m} \right) \partial_t B_{22}^n \\ &\quad + \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-v_x^{n+\alpha} \frac{\partial B_{22}^{n+\alpha}}{\partial x} - v_y^{n+\alpha} \frac{\partial B_{22}^{n+\alpha}}{\partial y} + 2 \frac{\partial v_y^{n+\alpha}}{\partial y} B_{22}^{n+\alpha} + 2 \frac{\partial v_y^{n+\alpha}}{\partial x} B_{12}^{n+\alpha} \right).\end{aligned}$$

Finally, the Jacobian terms of the shear components in the solid stress are given by:

$$\begin{aligned}\mu_L^s \frac{\delta(N_{,x} B_{11}^{n+\alpha} + N_{,y} B_{12}^{n+\alpha})}{\delta v_x^{n+\alpha}} &= \mu_L^s N_{,x} \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-N \frac{\partial B_{11}^{n+\alpha}}{\partial x} + 2 N_{,x} B_{11}^{n+\alpha} + 2 N_{,y} B_{12}^{n+\alpha} \right) \\ &\quad + \mu_L^s N_{,y} \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-N \frac{\partial B_{12}^{n+\alpha}}{\partial x} + N_{,x} B_{12}^{n+\alpha} + N_{,y} B_{22}^{n+\alpha} \right), \\ \mu_L^s \frac{\delta(N_{,x} B_{11}^{n+\alpha} + N_{,y} B_{12}^{n+\alpha})}{\delta v_y^{n+\alpha}} &= \mu_L^s N_{,x} \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-N \frac{\partial B_{11}^{n+\alpha}}{\partial y} \right) \\ &\quad + \mu_L^s N_{,y} \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-N \frac{\partial B_{12}^{n+\alpha}}{\partial y} + N_{,x} B_{11}^{n+\alpha} + N_{,y} B_{12}^{n+\alpha} \right),\end{aligned}$$

$$\begin{aligned}\mu_L^s \frac{\delta(N_{,x}B_{12}^{n+\alpha} + N_{,y}B_{22}^{n+\alpha})}{\delta v_x^{n+\alpha}} &= \mu_L^s N_{,x} \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-N \frac{\partial B_{12}^{n+\alpha}}{\partial x} + N_{,x}B_{12}^{n+\alpha} + N_{,y}B_{22}^{n+\alpha} \right) \\ &\quad + \mu_L^s N_{,y} \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-N \frac{\partial B_{22}^{n+\alpha}}{\partial x} \right),\end{aligned}$$

$$\begin{aligned}\mu_L^s \frac{\delta(N_{,x}B_{12}^{n+\alpha} + N_{,y}B_{22}^{n+\alpha})}{\delta v_y^{n+\alpha}} &= \mu_L^s N_{,x} \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-N \frac{\partial B_{12}^{n+\alpha}}{\partial y} + N_{,x}B_{11}^{n+\alpha} + N_{,y}B_{12}^{n+\alpha} \right) \\ &\quad + \mu_L^s N_{,y} \frac{\Delta t \alpha \gamma}{\alpha_m} \left(-N \frac{\partial B_{22}^{n+\alpha}}{\partial y} + 2N_{,y} \partial B_{22}^{n+\alpha} + 2N_{,x}B_{12}^{n+\alpha} \right).\end{aligned}$$

Appendix C. Parallel implementation of particle tracking method

In an Eulerian frame of reference, tracking a solid particle may be required to know the exact motion of the solid. To achieve this, we need to search for the solid particle in the entire domain according to its coordinates, interpolate the velocity at the particle position, and evolve the particle accordingly. This procedure is repeated at each time step, which can become a large overhead without efficient implementation. To avoid this issue, we adopt a parallelized implementation. We first broadcast the particle position to all the processors. In each processor, we split the elements into tetrahedrons. Then, the particle is searched in all the tetrahedrons. Inside each tetrahedron, assuming linear basis function, we have four basis functions corresponding to the four vertices $N_i(x, y, z) = a_i x + b_i y + c_i z + d_i$, $i = 1, 2, 3, 4$. According to the definition of the shape function, $N_i(x, y, z)$ takes the value of 1 at the node i , and takes the value of 0 at the rest nodes. Due to the linearity, for all the points (x, y, z) inside the tetrahedron, $N_i(x, y, z) \in [0, 1]$, $i = 1, 2, 3, 4$.

With this in mind, we solve for the coefficients a_i, b_i, c_i, d_i , $i = 1, 2, 3, 4$. Considering the first shape function, which takes the value of 1 at the first node, and becomes zero at the rest nodes:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (\text{C.1})$$

Considering all the four nodes, we have:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{C.2})$$

The coefficients can be solved as:

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{bmatrix}^{-1}. \quad (\text{C.3})$$

After getting the coefficients, the value of the shape function on the particle position (x, y, z) can be calculated as:

$$\begin{bmatrix} N_1(x, y, z) \\ N_2(x, y, z) \\ N_3(x, y, z) \\ N_4(x, y, z) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix}^T \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (\text{C.4})$$

if $\forall i = 1, 2, 3, 4$, $N_i(x, y, z) \in [0, 1]$, the particle is inside the tetrahedron. Once the particle is found, its velocity can be interpolated as $v = \sum_{i=1}^4 N_i(x, y, z)v_i$. The solid particle is then evolved by the velocity and passed back to the root processor.

References

- [1] D. Montgomery, F. Municchi, K. Leiderman, clotfoam: An open-source framework to simulate blood clot formation under arterial flow, *SoftwareX* 23 (2023) 101483.
- [2] D. S. Mahdy, Clearing of blood clots using helical robots, Ph.D. thesis, German University in Cairo (2019).
- [3] S. Nojiri, K. Mizushima, Y. Suzuki, T. Tsuji, T. Watanabe, Development of contact area variable surface for manipulation requiring sliding, in: 2019 2nd IEEE International Conference on Soft Robotics (RoboSoft), IEEE, 2019, pp. 131–136.
- [4] Y. Xue, R. Liu, Z. Li, D. Han, A review for numerical simulation methods of ship–ice interaction, *Ocean Engineering* 215 (2020) 107853.
- [5] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian-Eulerian methods, *Encyclopedia of Computational Mechanics Second Edition* (2017) 1–23.
- [6] C. Peskin, *The immersed boundary method*, Vol. 6, Cambridge University Press, 2002.
- [7] J. Benek, J. Steger, F. C. Dougherty, A flexible grid embedding technique with application to the euler equations, in: 6th computational fluid dynamics conference Danvers, 1983, p. 1944.
- [8] H. Tang, S. C. Jones, F. Sotiropoulos, An overset-grid method for 3d unsteady incompressible flows, *J. Comput. Phys.* 191 (2) (2003) 567–600.
- [9] R. Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for Dirichlet problem and applications, *Comp. Meth. Appl. Mech. Engrg.* 111 (3-4) (1994) 283–303.
- [10] F. P. Baaijens, A fictitious domain/mortar element method for fluid–structure interaction, *Int. J. Numer. Meth. Fluid* 35 (7) (2001) 743–761.
- [11] T. Dunne, An Eulerian approach to fluid–structure interaction and goal-oriented mesh adaptation, *Int. J. Numer. Meth. Fluid* 51 (9-10) (2006) 1017–1039.
- [12] T. Wick, Fully Eulerian fluid–structure interaction for time-dependent problems, *Comp. Meth. Appl. Mech. Engrg.* 255 (2013) 14–26.

- [13] X. Mao, V. Joshi, R. Jaiman, A variational interface-preserving and conservative phase-field method for the surface tension effect in two-phase flows, *J. Comput. Phys.* 433 (2021) 110166.
- [14] K. Kamrin, C. H. Rycroft, J.-C. Nave, Reference map technique for finite-strain elasticity and fluid–solid interaction, *Journal of the Mechanics and Physics of Solids* 60 (11) (2012) 1952–1969.
- [15] P. Sun, J. Xu, L. Zhang, Full Eulerian finite element method of a phase field model for fluid–structure interaction problem, *Computers & Fluids* 90 (2014) 1–8.
- [16] D. Mokbel, H. Abels, S. Aland, A phase-field model for fluid–structure interaction, *J. Comput. Phys.* 372 (2018) 823–840.
- [17] X. Mao, R. Jaiman, An interface and geometry preserving phase-field method for fully Eulerian fluid-structure interaction, *J. Comput. Phys.* (2023) 111903.
- [18] K. E. Jansen, C. H. Whiting, G. M. Hulbert, A generalized- α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method, *Comp. Meth. Appl. Mech. Engrg.* 190 (3-4) (2000) 305–319.
- [19] F. Shakib, T. J. Hughes, Z. Johan, A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations, *Comp. Meth. Appl. Mech. Engrg.* 89 (1-3) (1991) 141–219.
- [20] A. N. Brooks, T. J. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, *Comp. Meth. Appl. Mech. Engrg.* 32 (1-3) (1982) 199–259.
- [21] I. Harari, T. J. Hughes, What are c and h?: Inequalities for the analysis and design of finite element methods, *Comp. Meth. Appl. Mech. Engrg.* 97 (2) (1992) 157–192.
- [22] C. Johnson, Numerical solutions of partial differential equations by the finite element method, Cambridge University Press, 1987.
- [23] M. Brassel, E. Bretin, A modified phase field approximation for mean curvature flow with conservation of the volume, *Mathematical Methods in the Applied Sciences* 10 (34) (2011) 1157–1180.
- [24] R. K. Jaiman, M. Guan, T. Miyanawala, Partitioned iterative and dynamic subgrid-scale methods for freely vibrating square-section structures at subcritical Reynolds number, *Computers & Fluids* 133 (2016) 68–89.
- [25] V. Joshi, R. K. Jaiman, A positivity preserving and conservative variational scheme for phase-field modeling of two-phase flows, *J. Comput. Phys.* 360 (2018) 137–166.
- [26] E. Haines, Point in polygon strategies., *Graphics Gems* 4 (1994) 24–46.
- [27] J. O'Rourke, Computational geometry in C, Cambridge university press, 1998.

- [28] K. Sugiyama, S. Ii, S. Takeuchi, S. Takagi, Y. Matsumoto, A full Eulerian finite difference approach for solving fluid–structure coupling problems, *J. Comput. Phys.* 230 (3) (2011) 596–627.
- [29] C. T.-H. Stadtlander, Book review: Sustainable shipping in a changing Arctic (2020).
- [30] T. Schoolmeester, B. Alfthan, E. Baker, H. Gjerdi, J. Skaalvik, J. Fabres, J. Crump, K. Johnsen, L. Puikkonen, T. Kurvits, et al., Global linkages: A graphic look at the changing Arctic (rev. 1).
- [31] J. N. Larsen, G. Fondahl, Arctic human development report: Regional processes and global linkages, Nordic Council of Ministers, 2015.
- [32] M.-L. Timmermans, J. Marshall, Understanding Arctic Ocean circulation: A review of ocean dynamics in a changing climate, *Journal of Geophysical Research: Oceans* 125 (4) (2020) e2018JC014378.
- [33] A. Lavissière, R. Sohier, M. C. Lavissière, Transportation systems in the Arctic: A systematic literature review using textometry, *Transportation Research Part A: Policy and Practice* 141 (2020) 130–146.
- [34] F. Li, L. Huang, A review of computational simulation methods for a ship advancing in broken ice, *Journal of Marine Science and Engineering* 10 (2) (2022) 165.
- [35] L. Huang, J. Tuhkuri, B. Igrec, M. Li, D. Stagonas, A. Toffoli, P. Cardiff, G. Thomas, Ship resistance when operating in floating ice floes: A combined CFD & DEM approach, *Marine Structures* 74 (2020) 102817.