An immersogeometric formulation for free-surface flows with application to marine engineering problems

Qiming Zhu[a], Fei Xu[b,c], Songzhe Xu[a], Ming-Chen Hsu[b], Jinhui Yan[a,*]

[a]*Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign, 205 North Mathews Ave, Urbana, IL 61801, USA*
[b]*Department of Mechanical Engineering, Iowa State University, 2043 Black Engineering, Ames, IA 50011, USA*
[c]*Ansys Inc., 807 Las Cimas Parkway, Austin, TX 78746, USA*

Abstract

An immersogeometric formulation is proposed to simulate free-surface flows around structures with complex geometry. The fluid-fluid interface (air-water interface) is handled by the level set method, while the fluid-structure interface is handled through an immersogeometric approach by immersing structures into non-boundary-fitted meshes and enforcing Dirichlet boundary conditions weakly. Residual-based variational multiscale method (RBVMS) is employed to stabilize the coupled Navier-Stokes equations of incompressible flows and level set convection equation. Other level set techniques, including re-distancing and mass balancing, are also incorporated into the immersed formulation. Adaptive quadrature rule is used to better capture the geometry of the immersed structure boundary by accurately integrating the intersected background elements. Generalized-α method is adopted for time integration, which results in a two-stage predictor multi-corrector algorithm. GMRES solver preconditioned with block Jacobian matrices of individual fluid and level set subproblems is used for solving the coupled linear systems arising from the multi-corrector stage. The capability and accuracy of the proposed method are assessed by simulating three challenging marine engineering problems, which are a solitary wave impacting a stationary platform, dam break with an obstacle, and planing of a DTMB 5415 ship model. A refinement study is performed. The predictions of key quantities of interest by the proposed formulation are in good agreement with experimental results and boundary-fitted simulation results from others. The proposed formulation has great potential for wide applications in marine engineering problems.

Keywords: Immersogeometric method, Level set method, Free-surface flows, Dam break, Shiphydrodynamics

*Corresponding author
Email address: yjh@illinois.edu (Jinhui Yan)

Contents

1. Introduction

Free-surface flow simulations play an essential role in the design and optimization of many marine engineering structures, such as floating offshore wind turbines, tidal turbines, ships, underwater vehicles, etc. In addition to handling high Reynolds number turbulent flows, there are two key challenging problems in free-surface flow simulations. One problem is how to treat the fluid-fluid interface, the associated large property ratio between two fluid phases, pressure discontinuity, and possible violent topological interfacial changes. Another problem is how to treat the fluid-structure interface, which typically has complicated geometry for real engineering structures and the surrounding thin boundary layers.

The methods of treating free-surface can be classified into two categories, interface-tracking and interface capturing [1, 2]. Interface-tracking methods, such as Arbitrary Lagrangian–Eulerian (ALE) methods [3], front-tracking methods [4], boundary-integral methods [5], and Space–Time method [6], explicitly represent the free-surface by using a deformable mesh that moves with the free-surface deformation. Interface-tracking methods possess higher accuracy per degree-of-freedom and have been applied to several challenging problems in offshore engineering and additive manufacturing [7, 8]. However, mesh motion and re-meshing techniques are often required if the free-surface undergoes singular topological changes, which can be very challenging for some scenarios. Interface-capturing methods, including level set [9, 10], front-capturing methods [11], volume of fluid (VOF) [12], phase field [13, 14], and diffuse-interface methods [15, 16], define an auxiliary field in the computational domain and make use of an implicit function to represent the free-surface. Although interface-capturing methods typically need higher mesh resolution around the free-surface to compensate for the lower accuracy, they are more flexible and do not require any mesh motion or re-meshing procedures. The free-surface topological changes can be automatically handled by solving an additional scalar partial differential equation. Interface capturing methods have been widely applied to a wide range of interfacial problems, including bubble dynamics [17–19], jet atomization [20], and free-surface flows [21, 22].

The methods of treating fluid-structure interface can also be classified into two categories, boundary-fitted methods and immersed methods (or non-boundary-fitted methods). Among boundary-fitted methods, Arbitrary Lagrangian–Eulerian (ALE) method [3] and Space–Time method [6] are the two frequently used approaches. Both methods use meshes to represent the fluid-structure interface explicitly. One major difficulty of boundary-fitted methods is that the automatic generation of high-quality volumetric meshes that conform to the complex fluid-structure interface is difficult. It often requires intensive labor-processes, such as de-featuring, geometry cleanup, and mesh manipulation, which are time-consuming in the design loop through analysis. In the context of fluid-structure interaction simulations,

sophisticated mesh motion and re-meshing techniques [23–25] (similar to interface tracking methods) are often required, which makes the problem even more challenging. On the other hand, immersed methods [26] make use of a non-boundary-fitted fluid mesh to approximate the solutions of the fluid equations. Unlike the boundary-fitted methods, the fluid mesh can be independent of the surface representation. This type of method releases the strict mesh conforming constraint, circumvents mesh motion and re-mesh procedures, and simplifies the volumetric mesh generation significantly, especially for the structures with complex boundary geometry. The first immersed boundary method can be found in [27], which deals with computational fluid dynamics (CFD) analysis of heart valves with moving boundaries. Since that, the research on immersed methods has been growing significantly. Some recent developments using immersed approach can be found in [28–34].

Although immersed methods for single-phase fluid flows around complex geometries can be widely found in the literature, immersed methods based on variational principles for free-surface flows are still lacking. In this paper, an immersed free-surface formulation is developed by integrating the immersogeometric methods developed in [35, 36] and the free-surface flow formulation developed in [37–41]. On the one hand, the terminology of immersogeometric methods, inspired by isogeometric analysis [42, 43], denotes the immersed methods that accurately represent the immersed structure boundary to eliminate geometric errors. For example, immersogeometric methods can directly immerse the boundary representation (B-rep) of CAD models into the non-body-fitted background fluid mesh [44]. Some applications to heart valve modeling and compressible flow modeling of rotor-craft can be found in [45–48]. In the present work, immersogeometric concept relies on the Finite Cell Method (FCM), which is introduced by [49, 50] and has been applied to single-phase flow computations in [51, 52]. The FCM captures the structure geometry in intersected elements by adaptive quadrature, which increases the accuracy by adding additional levels of quadrature points. The adaptive quadrature scheme is based on the decomposition of each intersected element into sub-cells, which can be efficiently organized in hierarchical tree data structures. Large number of quadrature points in intersected elements are typically required in this method, but the implementation is extremely flexible and robust. Although unstructured tetrahedral elements are used, FCM can work for almost any geometric model. On the other hand, the techniques previously developed in [37, 38, 40, 41] are adopted to model the free-surface flow. In the formulation, the level set method is chosen to capture the free-surface because of its easiness of implementation and ability to represent complicated free-surface shape by an implicit function. The level set field is convected by the fluid velocity. The free-surface flow motion is governed by unified Navier-Stokes equations of incompressible flows, in which the fluid properties are evaluated with the assistance of the level set field.

Residual-based variational multi-scale formulation (RBVMS) [53], which is widely used for turbulence modeling in CFD simulations [54–56], is adopted to solve the coupled Navier-Stokes and level set equations. The combination of level set and RBVMS has been proved to be an effective technique to model multi-phase flows. The applications include offshore floating wind turbines [38, 39], tidal turbines [37], bubble dynamics [57], and metallic manufacturing [58]. Immersed methods by nature prevent the application of strong enforcement for Dirichlet boundary conditions. For that, a Nitsche-type weak enforcement of essential boundary conditions (weak BC) [59], which can be applied to both boundary-fitted and non-boundary-fitted meshes, is incorporated into the current immersogeometric formulation for free-surface flows.

The paper is structured as follows. Section 2 presents the continuous governing equations of free-surface flows, which include Navier-Stokes equations of incompressible flows and level set convection equation. Section 3 presents semi-discrete formulation, which including RBVMS, re-distancing, mass balancing, and weak BCs. Section 4 presents the tetrahedral finite cell method. Section 5 presents the time integration and linear solver. Section 6 presents the application of the proposed formulation to three challenging problems in marine engineering. The first problem is a solitary wave impacting a stationary platform. The second problem is the dam break with an obstacle. The third problem is the planing of a DTMB 5415 ship model. Simulated results are compared with experimental results and computational results based on boundary-fitted methods from other researchers. Section 7 concludes the paper and specifies the future research work.

2. Governing equations of free-surface flow

2.1. Level set method

In this section, we summarize the governing equations of free-surface flows based on level set method. Let $\Omega \subset \mathbb{R}^3$ denote air-water domain, Γ denote its boundary. In Ω, a scalar function $\phi(\boldsymbol{x}, t)$ is defined at each point. The free-surface is denoted by Γ_l, which is implicitly defined as

$$\Gamma_l = \{\boldsymbol{x} \in \Omega \,|\, \phi(\boldsymbol{x}, t) = 0\} \tag{1}$$

At air subdomain Ω_a and water subdomain Ω_w, $\phi(\boldsymbol{x}, t)$ is a signed distance function with respect to the free-surface. In present work, $\phi(\boldsymbol{x}, t)$ takes negative value in the air phase and positive value in the water phase, namely,

$$\Omega_a = \{\boldsymbol{x} \in \Omega \,|\, \phi(\boldsymbol{x}, t) < 0\} \tag{2}$$

$$\Omega_w = \{\boldsymbol{x} \in \Omega \,|\, \phi(\boldsymbol{x}, t) > 0\} \tag{3}$$

For a given point in Ω, the fluid density $\rho(\phi)$ and viscosity $\mu(\phi)$ can be computed as

$$\rho(\phi) = \rho_a \left(1 - H(\phi)\right) + \rho_w H(\phi) \tag{4}$$

$$\mu(\phi) = \mu_a \left(1 - H(\phi)\right) + \mu_w H(\phi) \tag{5}$$

where ρ_a, μ_a are the density and viscosity of air, ρ_w, μ_w are the density and viscosity of water, respectively, and $H(\phi)$ is the Heaviside function, defined by

$$H(\phi) = \begin{cases} 0 & \phi < 0 \\ \frac{1}{2} & \phi = 0 \\ 1 & \phi > 0 \end{cases} \tag{6}$$

2.2. Navier-Stokes equations of incompressible flow

The free-surface flow motion is governed by the unified Navier-Stokes equations of the incompressible flows, given as

$$\rho(\phi) \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} - \mathbf{f} \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}\left(\mathbf{u}, p\right) = \mathbf{0} \qquad \text{in } \Omega \tag{7}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \qquad \text{in } \Omega \tag{8}$$

$$\mathbf{u} = \mathbf{u}_g \qquad \text{on } \Gamma^{\mathrm{D}} \tag{9}$$

$$\boldsymbol{\sigma}\left(\mathbf{u}, p\right) \cdot \mathbf{n} = \mathbf{h} \qquad \text{on } \Gamma^{\mathrm{N}} \tag{10}$$

where $\mathbf{u}$, p, and $\mathbf{f}$ are the fluid velocity, pressure, and the external force per unit mass, respectively. $\boldsymbol{\sigma}\left(\mathbf{u}, p\right)$ is the Cauchy stress tensor, defined as

$$\boldsymbol{\sigma}\left(\mathbf{u}, p\right) = -p\,\mathbf{I} + 2\mu(\phi)\,\boldsymbol{\varepsilon}(\mathbf{u}) \tag{11}$$

where $\mathbf{I}$ is the identity tensor, and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor, defined as

$$\varepsilon(\mathbf{u}) = \frac{1}{2}\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^{\mathrm{T}}\right) \tag{12}$$

Eq. (9) and Eq. (10) define the boundary conditions, where $\mathbf{u}_g$ is the prescribed velocity on the Dirichlet boundary Γ^{D}, $\mathbf{h}$ is the traction vector on the Neumann boundary Γ^{N}, and $\mathbf{n}$ is the unit normal vector pointing in the wall-outward direction.

The level set field is convected by the fluid velocity, which can be modeled by means of

an additional convection equation, namely,

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \phi = 0 \qquad \text{in } \Omega \tag{13}$$

Eqs. (7)–(10), and Eq. (13) with appropriate initial conditions constitute the strong form governing equations of free-surface flow at the continuous level.

3. Semi-discrete formulation

3.1. RBVMS

In the present work, residual-based variational multi-scale (RBVMS) formulation is utilized to solve the strong form equations presented in the previous section. Consider a collection of disjoint elements $\{\Omega^e\}$, $\cup_e \Omega^e \subset \mathbb{R}^3$, with closure covering the fluid domain: $\Omega \subset \cup_e \overline{\Omega^e}$. Note that Ω^e is not necessarily a subset of Ω if a non-boundary-fitted approach is utilized. Let $\mathcal{V}_u^h$, $\mathcal{V}_p^h$, and $\mathcal{V}_s^h$ denote discrete velocity, pressure, and level set trial function spaces, and $\mathcal{W}_u^h$, $\mathcal{W}_p^h$, and $\mathcal{W}_s^h$ denote the corresponding test function spaces. The RBVMS formulation of free-surface flow is stated as follows. Find $\mathbf{u}^h \in \mathcal{V}_u^h$, $p^h \in \mathcal{V}_p^h$, and $\phi^h \in \mathcal{V}_s^h$ such that for all $\mathbf{w}^h \in \mathcal{W}_u^h$, $q^h \in \mathcal{W}_p^h$, and $\eta^h \in \mathcal{W}_s^h$:

$$B_{\text{VMS}}^{ns}\left(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}\right) + B_{\text{VMS}}^{\text{conv}}\left(\eta^h, \phi^h\right) - F_{\text{VMS}}\left(\{\mathbf{w}^h, q^h\}\right) = 0 \tag{14}$$

where $B_{\text{VMS}}^{ns}\left(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}\right)$, $B_{\text{VMS}}^{\text{conv}}\left(\eta^h, \phi^h\right)$, and $F_{\text{VMS}}\left(\{\mathbf{w}^h, q^h\}\right)$ are given as

$$\begin{aligned}
B_{\text{VMS}}^{ns}\left(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}\right) = & \int_\Omega \mathbf{w}^h \cdot \rho(\phi^h) \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h \right) \, d\Omega + \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right) \, d\Omega \\
& + \int_\Omega q^h \, \boldsymbol{\nabla} \cdot \mathbf{u}^h \, d\Omega \\
& - \sum_e \int_{\Omega^e \cap \Omega} \left(\mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}^h + \frac{\boldsymbol{\nabla} q^h}{\rho(\phi^h)} \right) \cdot \mathbf{u}' \, d\Omega \\
& - \sum_e \int_{\Omega^e \cap \Omega} p' \, \boldsymbol{\nabla} \cdot \mathbf{w}^h \, d\Omega + \sum_e \int_{\Omega^e \cap \Omega} \mathbf{w}^h \cdot (\mathbf{u}' \cdot \boldsymbol{\nabla} \mathbf{u}^h) \, d\Omega \\
& - \sum_e \int_{\Omega^e \cap \Omega} \frac{\boldsymbol{\nabla} \mathbf{w}^h}{\rho(\phi^h)} : (\mathbf{u}' \otimes \mathbf{u}') \, d\Omega \\
& + \sum_e \int_{\Omega^e \cap \Omega} \kappa_{dc}^{ns} \, \boldsymbol{\nabla} \mathbf{w}^h : \boldsymbol{\nabla}^s \mathbf{u}^h \, d\Omega \qquad (15) \\
B_{\text{VMS}}^{\text{conv}}\left(\eta^h, \phi^h\right) = & \int_\Omega \eta^h \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \phi^h \right) \, d\Omega
\end{aligned}$$

$$
\begin{aligned}
&- \sum_e \int_{\Omega^e \cap \Omega} \left(\mathbf{u}^h \cdot \boldsymbol{\nabla} \eta^h \right) \phi' \, d\Omega \\
&+ \sum_e \int_{\Omega^e \cap \Omega} \kappa^c_{dc} \, \nabla \eta \cdot \nabla \phi^h \, d\Omega
\end{aligned} \tag{16}
$$

$$
F_{\mathrm{VMS}} \left(\{ \mathbf{w}^h, q^h \} \right) = \int_\Omega \mathbf{w}^h \cdot \rho(\phi^h) \, \mathbf{f} \, d\Omega + \int_{\Gamma^{\mathrm{N}}} \mathbf{w}^h \cdot \mathbf{h} \, d\Gamma \tag{17}
$$

where the fine scale velocity $\mathbf{u}'$, fine scale pressure p', and fine scale level set function ϕ' are given as

$$
\mathbf{u}' = -\tau_{\mathrm{M}} \left(\rho(\phi^h) (\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \right) \tag{18}
$$

$$
p' = -\rho(\phi^h) \tau_{\mathrm{C}} \, \boldsymbol{\nabla} \cdot \mathbf{u}^h \tag{19}
$$

$$
\phi' = -\tau_\phi \left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \phi^h \right) \tag{20}
$$

Eqs. (14)–(20) feature an extension of the RBVMS of single-phase turbulent flows, first introduced in [53], to free-surface flows. The first two lines in Eq. (15) and the first line in Eq. (16) are the Galerkin formulation of Navier–Stokes equations and level set convection equation, respectively. The rest terms can be interpreted as a stabilized method or large eddy simulation (LES) turbulence model [53, 60–65]. The stabilization parameters τ_{M}, τ_{C}, and τ_ϕ are defined by

$$
\tau_{\mathrm{M}} = \left(\frac{C_t}{\Delta t^2} + \mathbf{u}^{\mathbf{h}} \cdot \mathbf{G} \, \mathbf{u}^{\mathbf{h}} + C_I \, \nu(\phi^h)^2 \, \mathbf{G} : \mathbf{G} \right)^{-1/2} \tag{21}
$$

$$
\tau_{\mathrm{C}} = (\tau_{\mathrm{M}} \, \mathrm{tr} \, \mathbf{G})^{-1} \tag{22}
$$

$$
\tau_\phi = \left(\frac{C_t}{\Delta t^2} + \mathbf{u}^{\mathbf{h}} \cdot \mathbf{G} \, \mathbf{u}^{\mathbf{h}} \right)^{-1/2} \tag{23}
$$

where Δt is the time-step size, C_I is a positive constant [66], $\nu(\phi^h) = \mu(\phi^h)/\rho(\phi^h)$ is the fluid kinematic viscosity, $\mathbf{G}$ is the element metric tensor calculated by the mapping from the iso-parametric element to its physical counterpart. It is defined as the following index notation (Einstein summation notation is used).

$$
G_{ij} = \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j} \tag{24}
$$

where $\boldsymbol{\xi}$ is the parametric coordinates, $\mathrm{tr} \, \mathbf{G}$ is the trace of $\mathbf{G}$.

In order to further improve the stability for high Reynolds number turbulent free-surface

flow simulations, discontinuity capturing is used in our formulation. For that, two terms, $\sum_e \int_{\Omega^e \cap \Omega} \kappa_{dc}^{ns} \ \boldsymbol{\nabla} \mathbf{w}^h : \boldsymbol{\nabla}^s \mathbf{u}^h \ d\Omega$ and $\sum_e \int_{\Omega^e \cap \Omega} \kappa_{dc}^{c} \ \nabla \eta^h \cdot \nabla \phi^h \ d\Omega$, are added to Eq. (15) and Eq. (16), where κ_{dc}^{ns} and κ_{dc}^{c} are residual-based discontinuity capturing parameters. The definitions can be found in [67].

3.2. Re-distancing and mass balancing

In numerical setting, the density and viscosity are calculated as

$$\rho(\phi^h) = \rho_a \left(1 - H_\epsilon(\phi^h)\right) + \rho_w H_\epsilon(\phi^h) \tag{25}$$

$$\mu(\phi^h) = \mu_a \left(1 - H_\epsilon(\phi^h)\right) + \mu_w H_\epsilon(\phi^h) \tag{26}$$

where $H_\epsilon(\phi)$ is the regularized Heaviside function, defined by

$$H_\epsilon(\phi^h) = \begin{cases} 0 & \phi^h \leq -\ \epsilon \\ \dfrac{1}{2}\left(1 + \dfrac{\phi^h}{\epsilon} + \dfrac{1}{\pi} sin(\dfrac{\phi^h \pi}{\epsilon})\right) & |\phi^h| < \quad \epsilon \\ 1 & \phi^h \geq +\ \epsilon \end{cases} \tag{27}$$

where ϵ is the free-surface thickness, which scales with the element length around the free-surface, given as

$$\epsilon = \alpha \left(\frac{\nabla \phi^h}{||\nabla \phi^h||} \cdot \mathbf{G} \frac{\nabla \phi^h}{||\nabla \phi^h||} \right)^{-1/2}. \tag{28}$$

Using the regularized Heaviside function requires the level set field to satisfy the signed distance property. However, the level set field may lose its signed distance property as being convected by fluid velocity. To recover that, a re-distancing approach based on Eikonal equation with the constraint on the air-water interface is added to the formulation. The Eikonal equation reads

$$||\nabla \phi_d|| = 1 \ in \ \Omega_a \tag{29}$$

$$||\nabla \phi_d|| = 1 \ in \ \Omega_w \tag{30}$$

$$\phi_d = 0 \ on \ \Gamma_l \tag{31}$$

where ϕ_d is the re-distanced level set field. In the present work, a pseudo-time $\tilde{t}$ that scales with element length around the free-surface, is introduced to make the equation dynamic. Then, the strong form equation of the re-distancing process can be stated as: given ϕ^h, find

the ϕ_d^h satisfy the following equation

$$\frac{\partial \phi_d}{\partial \tilde{t}} + \text{sign}(\phi_d)(||\nabla \phi_d|| - 1) = 0 \qquad \text{in } \Omega \tag{32}$$

$$\phi_d(\boldsymbol{x}, \tilde{t} = 0) = \phi(\boldsymbol{x}, t) \quad \text{in } \Omega \tag{33}$$

VMS is employed to solve the above equation. The weak formulation of the re-distancing problem is stated as followed: given ϕ^h, find ϕ_d^h, such that for all the test functions $\eta_d^h \in \mathcal{W}_s^h$,

$$\begin{aligned}
&\int_\Omega {\eta_d}^h \left(\frac{\partial \phi_d^h}{\partial \tilde{t}} + S_\epsilon(\phi_d^h)(||\nabla \phi_d^h|| - 1) \right) \, d\Omega \\
&+ \sum_e \int_{\Omega^e \cap \Omega} \tau_\phi^d S_\epsilon(\phi_d^h) \frac{\nabla \phi_d^h}{||\nabla \phi_d^h||} \cdot \nabla \eta_d^h \left(\frac{\partial \phi_d^h}{\partial \tilde{t}} + S_\epsilon(\phi_d^h)(||\nabla \phi_d^h|| - 1) \right) \, d\Omega \\
&+ \sum_e \int_{\Omega^e \cap \Omega} \eta_d^h \lambda_{pen} \frac{\partial H_\epsilon}{\partial \phi_d^h} (\phi_d^h - \phi^h) \, d\Omega = 0
\end{aligned} \tag{34}$$

where $S_\epsilon(\phi_d^h) = 2H_\epsilon(\phi_d^h) - 1$ is the regularized sign function, $S_\epsilon(\phi_d^h)\frac{\nabla \phi_d^h}{||\nabla \phi_d^h||}$ is an equivalent "convective velocity", τ_ϕ^d is a Streamline Upwind Petrov-Galerkin (SUPG) stabilization parameter [60]. λ_{pen} is a penalty parameter to enforce the air-water interface, which is solved by Navier-Stokes and level set convection equations. With help of $\frac{\partial H_\epsilon}{\partial \phi_d^h}$, the penalty term is independent of mesh size [37] and only active around the free-surface.

Level set method by nature is not mass conservative. In order to restore the global mass balance, we shift the level set field by a global constant ϕ'. The solution of ϕ' is obtained by recovering the mass conservation equation, which reads

$$\int_\Omega \frac{\partial H_\epsilon(\phi_d^h + \phi')}{\partial t} \, d\Omega = \int_{\partial\Omega} H_\epsilon(\phi_d^h + \phi') \mathbf{u}^h \cdot \mathbf{n} \, d\partial\Omega \tag{35}$$

The above equation is obtained by the global mass conservation law, given as

$$\int_\Omega \frac{\partial \rho(\phi^h)}{\partial t} \, d\Omega = \int_{\partial\Omega} \rho(\phi^h) \mathbf{u}^h \cdot \mathbf{n} \, d\partial\Omega \tag{36}$$

The above mass-balancing procedure is performed after the re-distancing process. This mass balancing scheme is very efficient because only a scalar equation needs to be solved. Since the level set field is shifted by a global constant, it does not change the signed distance property obtained in the re-distancing stage.

3.3. Variational consistent weak enforcement of Dirichlet boundary conditions

Strongly imposing Dirichlet boundary conditions around fluid-structure interface is not feasible in an immersed approach. In the present work, the Dirichlet boundary condition is enforced weakly by using a Nitsche-based method [59, 68]. For that, Eq. (37) is added to the left hand side of Eq. (15) if no-slip boundary condition is used and Eq. (38) is added to the left hand side of Eq. (15) if no-penetration boundary condition is used:

$$
\begin{aligned}
&- \int_{\Gamma^{\mathrm{D}}} \mathbf{w}^h \cdot \left(\sigma(\mathbf{u}^h, p^h)\mathbf{n}\right) \; d\Gamma \\
&- \int_{\Gamma^{\mathrm{D}}} \left(2\mu(\phi)\,\boldsymbol{\varepsilon}(\mathbf{w}^h)\,\mathbf{n} + q^h\,\mathbf{n}\right) \cdot \left(\mathbf{u}^h - \mathbf{u}_g\right) \; d\Gamma \\
&+ \int_{\Gamma^{\mathrm{D}}} \tau^B \mathbf{w}^h \cdot \left(\mathbf{u}^h - \mathbf{u}_g\right) \; d\Gamma
\end{aligned}
\tag{37}
$$

$$
\begin{aligned}
&- \int_{\Gamma^{\mathrm{D}}} \mathbf{w}^h \cdot \mathbf{n} \left(\sigma(\mathbf{u}^h, p^h) : (\mathbf{n} \otimes \mathbf{n})\right) \; d\Gamma \\
&- \int_{\Gamma^{\mathrm{D}}} \left(2\mu(\phi)\,\boldsymbol{\varepsilon}(\mathbf{w}^h)\,\mathbf{n} + q^h\,\mathbf{n}\right) \cdot \mathbf{n} \left(\mathbf{u}^h \cdot \mathbf{n} - \mathbf{u}_g \cdot \mathbf{n}\right) \; d\Gamma \\
&+ \int_{\Gamma^{\mathrm{D}}} \tau^B \mathbf{w}^h \cdot \mathbf{n} \left(\mathbf{u}^h \cdot \mathbf{n} - \mathbf{u}_g \cdot \mathbf{n}\right) \; d\Gamma
\end{aligned}
\tag{38}
$$

The above formulation can be derived by an augmented Lagrangian multiplier approach or a symmetric interior penalty discontinuous Galerkin method. The first term and the second term in Eq. (37) and Eq. (38) are the so-called consistency and adjoint-consistency terms, respectively. The detailed interpretation of these terms can be found in Bazilevs and Hughes [59]. The parameter τ^B in the last term of Eq. (37) and Eq. (38) is a penalty-like stabilization parameter that helps to satisfy the Dirichlet boundary condition and improve the stability of the formulation. τ^B needs to be carefully chosen. If τ^B is too large, the penalty term dominates the formulation, overshadowing the variational consistency and resulting in an ill-conditioned stiffness matrix. If τ^B is too small, the solution is not stable. More discussions for the appropriate choice of τ^B for immersed methods can be found [35]. In our paper, $\tau^B = C\mu(\phi)/\mu_a$ is used, where C is a constant. Considering the structure boundary may intersect the free-surface, τ^B is scaled with $\mu(\phi)$ to provide a bigger penalty in the water phase. In the present work, C is set to 10^3, which is calibrated by the numerical experiments to achieve a good balance between accuracy and stability.

4. Tetrahedral finite cell method

The main challenge of immersed methods is the geometrically accurate evaluation of volume and surface integrals in the variational formulation in intersected elements. The immersogeometric method in the present work largely relies on the tetrahedral finite cell method (FCM), which uses a volume quadrature method based on recursive subdivision of intersected elements and a surface quadrature method based on an independent surface mesh. In this section, we briefly present the key techniques of FCM and an octree based point location query that can quickly determine whether a quadrature point is located inside or outside the fluid domain.

4.1. Recursive quadrature points generation

Fig. 1 (extracted from [52]) shows the basic concept of FCM method for a 2D case. In FCM, the original computational domain Ω_{phy} is extended by a fictitious domain Ω_{fict} to an embedding domain Ω that can be easily meshed. This introduces several elements (cut elements) that are intersected by the immersed structure boundary. Following the idea from [69], we use a volume quadrature based on recursive subdivision of cut elements and a surface quadrature using an independent surface mesh.

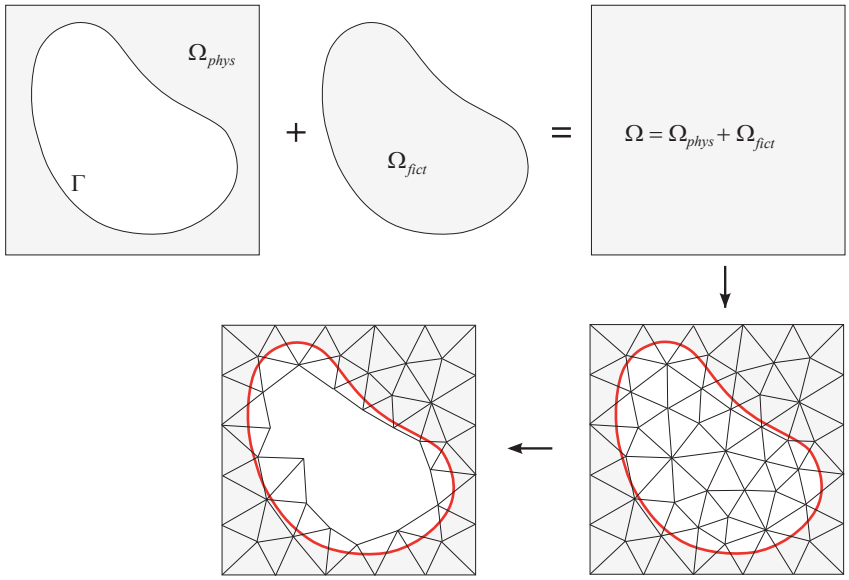


Figure 1: Physical domain (grey region) is extended by fictitious domain (white region) by FCM [52]

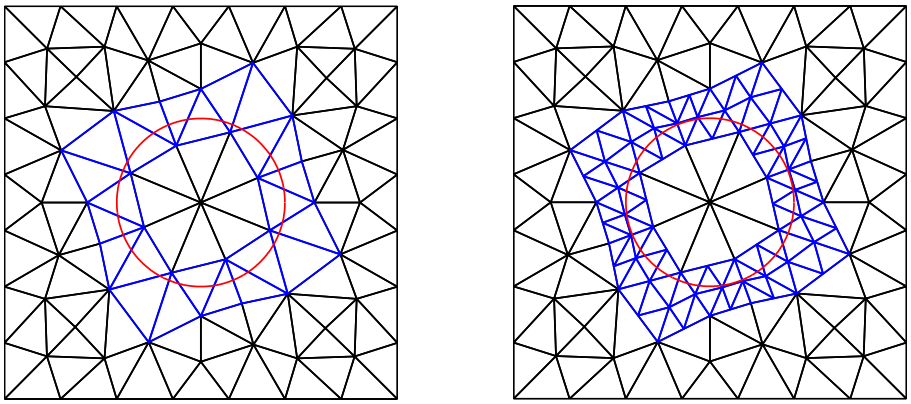
Accuracy of volume quadrature is important for capturing the immersed structure boundary. In present work, recursive subdivision of cut elements is used. For the elements with all nodes inside the computational domain Ω_{phy}, standard 4-point quadrature rule is used

for linear tetrahedron element. For the elements with all nodes outside the computational domain Ω_{phy}, no quadrature point is generated. For cut elements, an element is split into 12 sub-tetrahedral elements recursively. For each sub-tetrahedron, 4-point quadrature is used. The recursive process is performed until the pre-set recursive level is reached. For clarity, Fig. 2 shows a subdivision process of cut elements of a triangular mesh up to level = 3. Enough recursive level is important for accurate geometry representation of the immersed structure boundary. In present work, recursive level = 2 is chosen. Fig. 3 (a) shows the generated quadrature points with level = 1. The quadrature points inside computational domain Ω_{phy} (green points) are used in numerical integration, while quadrature points outside computational domain Ω_{phy} (magenta points) are discarded. Fig. 3 (b) shows the degrees of freedom that will be included or discarded in the computation. Please note that the nodes of cut elements which are outside the computational domain is important for enforcement of Dirichlet boundary condition, so the degrees of freedom of these nodes still need to be solved, even though these nodes also belong to some elements which are totally outside the computational domain Ω_{phy} (magenta circles in Fig. 3 (b)). Only the degrees of freedom of nodes that do not belong to any cut elements will be discarded. For the specific mesh shown in Fig. 3, the only degrees of freedom that will be discarded is the green circle in the center.

Another challenge is the surface integral in the weak BC formulation. The quadrature points of the surface integration locate on an independent surface mesh. To perform the surface integration, the coordinates of these surface quadrature points must then be located in the parameter space of the tetrahedral finite elements in which they fall. This requires us to invert the mapping from the finite element parameter space to physical space. Basis function values of their background volume elements and weak BC terms are then evaluated and integrated at these surface quadrature points and interpolated by the volumetric element containing them. To speed up the process of location query, an octree is constructed. The tetrahedral elements are represented by tight bounding boxes and are sorted into a hierarchical octree. When we query the background element of quadrature points, most elements are eliminated by the octree search. At the deepest level of octree, we need to calculate the parametric coordinate for possible elements and judge whether this quadrature point fall in this element. Please note that it is necessary to have enough surface mesh elements to get accurate surface integration.

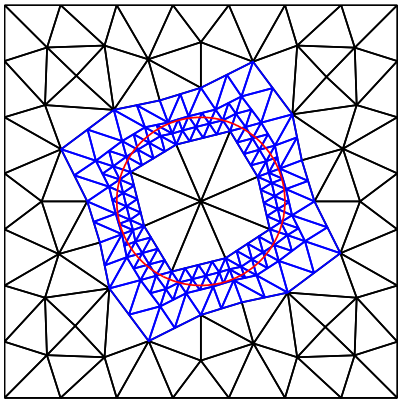
4.2. In-out test by ray-tracing method

Following [52], we briefly present the in-out test for quadrature points. To speed up the in-out test, an octree is constructed. Firstly, all the surface triangle elements are inserted into a hierarchical octree. Secondly, ray-octree intersections are performed recursively to reduce

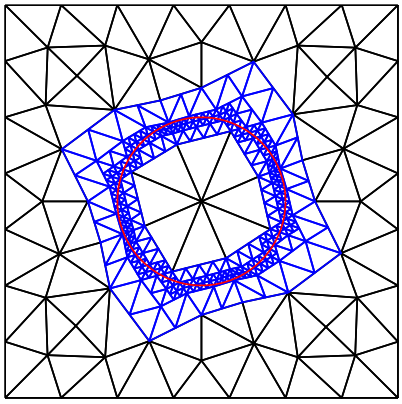


(a) Level = 0

(b) Level = 1

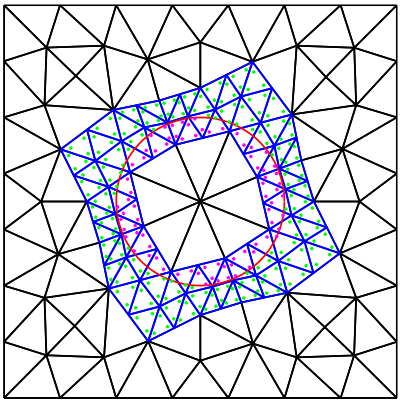


(c) Level = 2

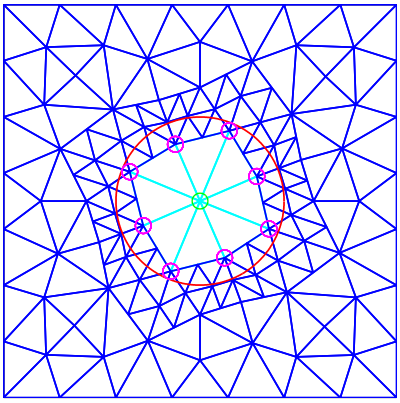


(d) Level = 3

Figure 2: Recursive subdivision of cut elements of a triangular mesh (The red circle denotes the immersed structure boundary, blue elements denote the cut elements in recursive subdivision).



(a)



(b)

Figure 3: Quadrature points in cut elements and degrees of freedom in fictitious domain (recursive level = 1)

the unnecessary ray-triangle intersections. Thirdly, ray-triangle intersections are needed to judge whether ray intersect with possible triangles from the deepest level of the octree. Fourthly, the number of intersection between ray and immersed surface mesh are counted, denoted by N. If the number N is odd, the point is inside the surface. If the number N is even, the point is outside the surface. We assume the immersed surface mesh is closed, so one ray is needed for the in-out test. Fig. 4 provide the explanation of quad-tree with triangle elements.

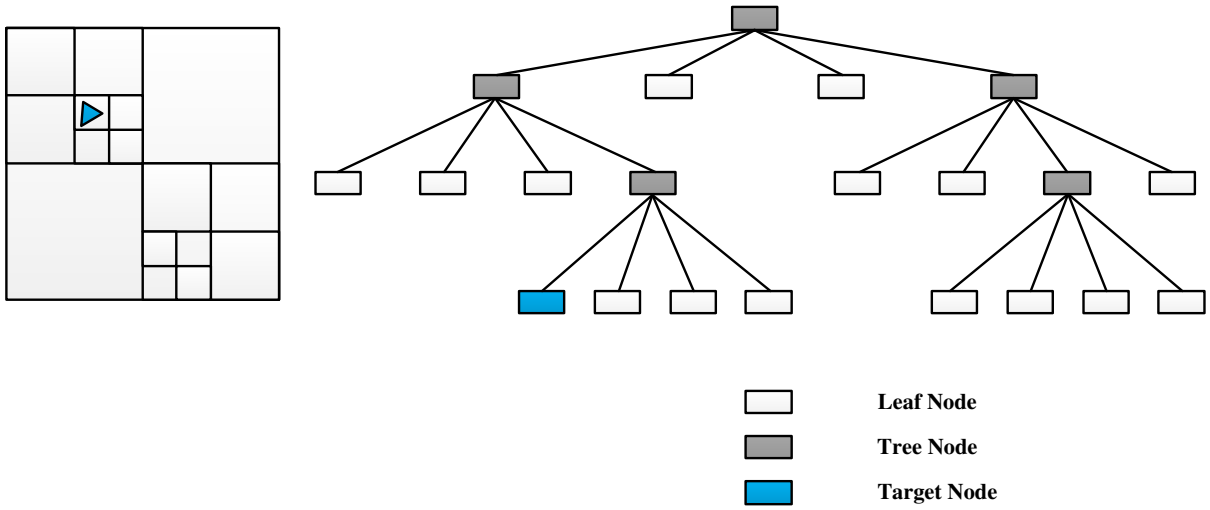


Figure 4: In-out test by octree search (example with quad-tree)

5. Time integration

5.1. Generalized-α method

Generalized-α method [70, 71] is used for time integration. In the generalized-α method, the residuals of free-surface flow equations are evaluated with intermediate-level fluid velocity and level set solutions at each time step, namely,

$$\dot{\mathbf{u}}_{n+\alpha_m} = \alpha_m \dot{\mathbf{u}}_{n+1} + (1-\alpha_m)\,\dot{\mathbf{u}}_n \tag{39}$$

$$\mathbf{u}_{n+\alpha_f} = \alpha_f \mathbf{u}_{n+1} + (1-\alpha_f)\,\mathbf{u}_n \tag{40}$$

$$\dot{\phi}_{n+\alpha_m} = \alpha_m \dot{\phi}_{n+1} + (1-\alpha_m)\,\dot{\phi}_n \tag{41}$$

$$\phi_{n+\alpha_f} = \alpha_f \phi_{n+1} + (1-\alpha_f)\,\phi_n \tag{42}$$

where the quantities with subscript $n+1$ are the unknown solutions at time step $n+1$, and the quantities with superscript n are known solutions are the previous time step n. Besides,

the relationship between nodal degrees of freedom and their time derivatives are given by the following Newmark-β scheme.

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t((1-\gamma)\dot{\mathbf{u}}_n + \gamma\dot{\mathbf{u}}_{n+1}) \tag{43}$$

$$\phi_{n+1} = \phi_n + \Delta t((1-\gamma)\dot{\phi}_n + \gamma\dot{\phi}_{n+1}) \tag{44}$$

In Eqs. (39)–(44), α_m, α_f, and γ are parameters of the Generalized-α and Newmark-β methods chosen based on the unconditional stability and second-order accuracy. With the above definitions, the application of the time integration of the coupled free-surface flow formulation leads to the following nonlinear equations at each time step.

$$\begin{cases} N_M\left(\dot{\mathbf{u}}_{n+\alpha_m}, p_{n+1}, \dot{\phi}_{n+\alpha_m}\right) & = 0 \\ N_C\left(\dot{\mathbf{u}}_{n+\alpha_m}, p_{n+1}, \dot{\phi}_{n+\alpha_m}\right) & = 0 \\ N_L\left(\dot{\mathbf{u}}_{n+\alpha_m}, p_{n+1}, \dot{\phi}_{n+\alpha_m}\right) & = 0 \end{cases} \tag{45}$$

where N_M, N_C, and N_L are the vectors of nodal residuals of fluid momentum, fluid continuity, and level set convection equations, respectively. To solve the above equations, Newton's method is adopted, which results in the following two-stage predictor multi-corrector algorithm.

Predictor stage:

$$\dot{\mathbf{u}}^0_{n+1} = \frac{\gamma-1}{\gamma}\dot{\mathbf{u}}_n \tag{46}$$

$$\mathbf{u}^0_{n+1} = \mathbf{u}_n \tag{47}$$

$$p^0_{n+1} = p_n \tag{48}$$

$$\dot{\phi}^0_{n+1} = \frac{\gamma-1}{\gamma}\dot{\phi}_n \tag{49}$$

$$\phi^0_{n+1} = \phi_n \tag{50}$$

where the quantities with superscript **0** are initial guesses, and **0** denotes the initial value of the Newton-iteration counter.

Multi-corrector stage: Repeat the following procedures until convergence.

1. Evaluate intermediate levels

$$\dot{\mathbf{u}}^l_{n+\alpha_m} = \alpha_m\dot{\mathbf{u}}^l_{n+1} + (1-\alpha_m)\,\dot{\mathbf{u}}_n \tag{51}$$

$$\mathbf{u}^l_{n+\alpha_f} = \alpha_f\mathbf{u}^l_{n+1} + (1-\alpha_f)\,\mathbf{u}_n \tag{52}$$

$$\dot{\phi}^l_{n+\alpha_m} = \alpha_m \dot{\phi}^l_{n+1} + (1-\alpha_m)\,\dot{\phi}_n \tag{53}$$

$$\phi^l_{n+\alpha_f} = \alpha_f \phi^l_{n+1} + (1-\alpha_f)\,\phi_n \tag{54}$$

where l is the Newton-iteration counter.

2. Use the solution of intermediate level to evaluate the right hand side residuals and the corresponding Jacobian matrix.

$$\begin{cases} \left.\frac{\partial R_M}{\partial \dot{\mathbf{u}}_{n+1}}\right|_l \Delta\dot{\mathbf{u}}^l_{n+1} + \left.\frac{\partial R_M}{\partial p_{n+1}}\right|_l \Delta p^l_{n+1} + \left.\frac{\partial R_M}{\partial \dot{\phi}_{n+1}}\right|_l \Delta\dot{\phi}^l_{n+1} &= -R^l_M \\ \left.\frac{\partial R_C}{\partial \dot{\mathbf{u}}_{n+1}}\right|_l \Delta\dot{\mathbf{u}}^l_{n+1} + \left.\frac{\partial R_C}{\partial p_{n+1}}\right|_l \Delta p^l_{n+1} + \left.\frac{\partial R_C}{\partial \dot{\phi}_{n+1}}\right|_l \Delta\dot{\phi}^l_{n+1} &= -R^l_C \\ \left.\frac{\partial R_L}{\partial \dot{\mathbf{u}}_{n+1}}\right|_l \Delta\dot{\mathbf{u}}^l_{n+1} + \left.\frac{\partial R_L}{\partial p_{n+1}}\right|_l \Delta p^l_{n+1} + \left.\frac{\partial R_L}{\partial \dot{\phi}_{n+1}}\right|_l \Delta\dot{\phi}^l_{n+1} &= -R^l_L \end{cases} \tag{55}$$

The above linear equations are solved to get the increment of the velocity, pressure, and level set unknowns.

3. Correct the solutions as follows

$$\dot{\mathbf{u}}^{l+1}_{n+1} = \dot{\mathbf{u}}^l_{n+1} + \Delta\dot{\mathbf{u}}^l_{n+1} \tag{56}$$

$$\mathbf{u}^{l+1}_{n+1} = \mathbf{u}^l_{n+1} + \gamma\Delta t\Delta\dot{\mathbf{u}}^l_{n+1} \tag{57}$$

$$p^{l+1}_{n+1} = p^l_{n+1} + \Delta p^l_{n+1} \tag{58}$$

$$\dot{\phi}^{l+1}_{n+1} = \dot{\phi}^l_{n+1} + \Delta\dot{\phi}^l_{n+1} \tag{59}$$

$$\phi^{l+1}_{n+1} = \phi^l_{n+1} + \gamma\Delta t\Delta\dot{\phi}^l_{n+1} \tag{60}$$

5.2. Fully-coupled linear solver

The multi-corrector stage requires the solution of a large linear system given by Eq. (55), which couples the different components of the free-surface flow formulation. To increase the robustness of the formulation, Eq. (55) is solved by a direct coupling approach [72] based on GMRES solver [73], in which the Jacobian matrix is fully constructed with all terms in RBVMS represented, namely,

$$J = \begin{bmatrix} \frac{\partial R_M}{\partial \dot{\mathbf{u}}_{n+1}} & \frac{\partial R_M}{\partial p_{n+1}} & \frac{\partial R_M}{\partial \dot{\phi}_{n+1}} \\ \frac{\partial R_C}{\partial \dot{\mathbf{u}}_{n+1}} & \frac{\partial R_C}{\partial p_{n+1}} & \frac{\partial R_C}{\partial \dot{\phi}_{n+1}} \\ \frac{\partial R_L}{\partial \dot{\mathbf{u}}_{n+1}} & \frac{\partial R_L}{\partial p_{n+1}} & \frac{\partial R_L}{\partial \dot{\phi}_{n+1}} \end{bmatrix} \tag{61}$$

The condition number of the above matrix is typically very large due to the complexity of the free-surface flow problems. To improve the efficiency and robustness, the following pre-conditioning matrix is used, which represents the inverse of the decoupled Jacobian matrices

for individual Navier-Stokes and level set problems, namely,

$$M = \begin{bmatrix} \frac{\partial R_M}{\partial \dot{\mathbf{u}}_{n+1}} & \frac{\partial R_M}{\partial p_{n+1}} & \mathbf{0} \\ \frac{\partial R_C}{\partial \dot{\mathbf{u}}_{n+1}} & \frac{\partial R_C}{\partial p_{n+1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial R_L}{\partial \dot{\phi}_{n+1}} \end{bmatrix}^{-1} \tag{62}$$

The preconditioning problems are solved by diagonally-preconditioned GMRES solver.

Remark. While the level set convection is solved within the Newton iterations, the re-distancing and mass balancing of level set field processes are performed after the predictor multi-corrector stage for each time step. This is done from the consideration of computational cost.

6. Numerical examples

In this section, the proposed formulation is applied to three marine engineering problems. The first problem is a solitary wave impacting a stationary platform, which is a well-known benchmark problem, widely used for validating free-surface flow simulations. A refinement study is performed on this problem. The second problem is the 3D dam break with an obstacle, which involves violent free-surface evolution. Rich experiment data for pressure is available. The third problem is the free-surface simulation of the planing of a scaled DTMB 5415 ship model. All the simulations in the present paper make use of linear tetrahedral elements. The simulated results are compared against experimental results and computational results from other researchers in the literature to validate the accuracy of the proposed formulation.

6.1. Solitary wave impacting a stationary platform

The computational setup of this problem is defined as follows. As shown in Fig. 5, the simulation domain is a rectangular box with dimensions 10 m × 1 m × 1 m. The platform with dimensions 1.524 m × 0.4 m × 0.3 m is located inside the simulation domain, with its front face 5 m away from the inlet of the simulation domain. A second-order solitary wave profile based on potential flow theory is initialized in the simulation domain. The initial level set function, velocity vectors are defined as

$$\phi = d\left[\zeta \mathrm{sech}(q)^2 - \frac{3}{4}\zeta^2 \mathrm{sech}(q)^2\right] \tag{63}$$

$$u = \sqrt{gd}\left\{\zeta \mathrm{sech}(q)^2 + \zeta^2 \mathrm{sech}(q)^2 \times \left[\frac{1}{4} - \mathrm{sech}(q)^2 - \frac{3}{4}(\frac{s}{d})^2(2 - 3\mathrm{sech}(q)^2)^2\right]\right\} \tag{64}$$

$$v = 0 \tag{65}$$

$$w = \sqrt{gd}\zeta\sqrt{3\zeta}\left(\frac{s}{d}\right)\mathrm{sech}(q)^2\tanh(q) \times \left\{1 - \zeta\left[\frac{3}{8} + 2\mathrm{sech}(q)^2 + \frac{1}{2}\left(\frac{s}{d}\right)^2(1 - 3\mathrm{sech}(q)^2)\right]\right\} \tag{66}$$

where (u, v, w) are the velocity components in stream-wise, span-wise and vertical directions, g is magnitude of gravitational acceleration, d is still water depth, H is the wave height, ζ is the ratio between wave height and still water depth $\zeta = \frac{H}{d}$, $c = \sqrt{gd}(1 + \frac{1}{2}\zeta - \frac{3}{20}\zeta^2)$ is the wave speed, $q = \frac{\sqrt{3\zeta}}{2d}(1 - \frac{5}{8}\zeta)(x - ct)$, $s = z + d$, z is the distance from still water surface. The air-water interface (far from the peak) in the hydrostatic configuration locates at $z = 0$. The parameters in this simulation case are chosen as followed: $d = 0.234696$ m, ζ =0.42, and zero clearance (the distance between the bottom surface of the platform and the still water level) are used. The distance between the wave peak and the front surface of the platform is 2 m.

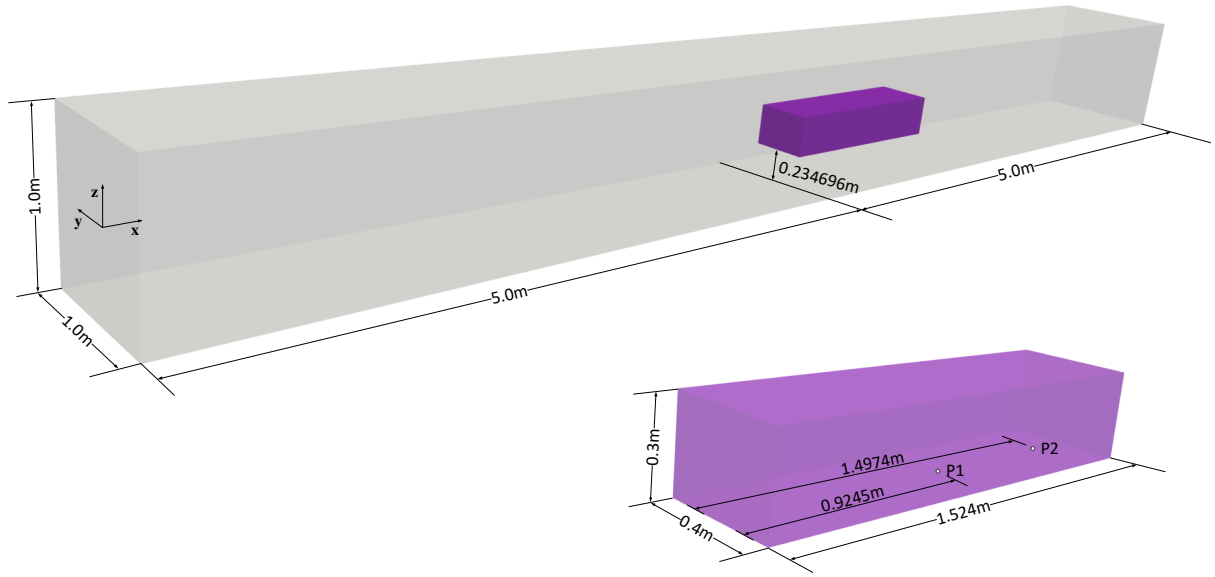


Figure 5: Simulation domain of the solitary wave case

The boundary conditions are defined as follows. Strong no-penetration boundary condition is used for inlet, outlet, side, and bottom boundaries of the simulation domain. Traction-free boundary condition is used for the top surface. Finally, weakly enforced no-slip boundary condition, based on Eq. (37), is used for the fluid-platform interface. The time step is 0.0005 s in the following simulations. To capture the free-surface evolution and hydrodynamic load better, the region around the air-water interface and the platform is refined. A refinement study of the immersogemetric approach is performed using three meshes. A boundary-fitted simulation using the element lengths based on the fine mesh is also performed. The total

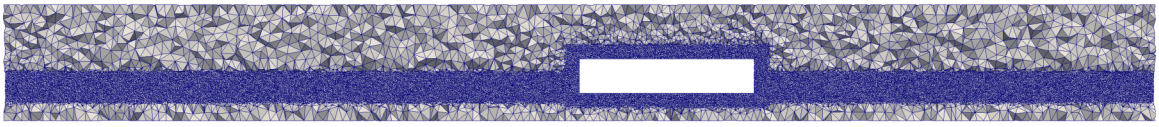
number of nodes, elements, and element lengths of the four meshes are summarized in Table 1 and Table 2. Fig. 6 shows a snapshot of the boundary-fitted mesh and the immersed fine mesh in the central plane ($y = 0$).

Table 1: Element lengths of the meshes in the solitary wave case

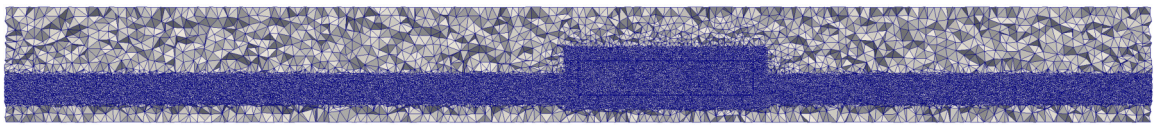
	Near platform	Near outer boundary
Immersed coarse mesh	0.028 m	0.15 m
Immersed medium mesh	0.020 m	0.15 m
Immersed fine mesh	0.014 m	0.15 m
boundary-fitted mesh	0.014 m	0.15 m

Table 2: Number of elements and nodes of the meshes in the solitary wave case

	Number of elements	Number of nodes
Immersed coarse mesh	1,839,451	302,061
Immersed medium mesh	3,842,305	627,472
Immersed fine mesh	8,614,067	1,400,329
boundary-fitted mesh	8,080,835	1,318,197



(a) boundary-fitted mesh



(b) Immersed fine mesh

Figure 6: Meshes of the solitary wave case

Fig. 7 shows the instantaneous free-surface shape colored by velocity magnitude. Both the velocity magnitude and free-surface shape look very natural. The simulation is also able to capture the flow separations near the sharp edge of the platform. Fig. 8 shows the normalized pressure at the two points. The origin of coordinate system is the geometric center of inlet. The coordinate of P1 and P2 are (5.9245 m, 0.0 m, -0.265304 m) and (6.4974m, 0.0m, -0.265304m), respectively. Please note that $t = 0$ is the time when the wave crest arrives

at P1. The location of the two points can be found in Fig. 5. To validate the proposed formulation, the experimental measurements obtained by [74] are also plotted in Fig. 5. All the meshes generate quite accurate results. However, results with coarse mesh over-predict the pressure. One possible reason is that the thickness of air-water interface scales with local mesh size in the formulation. As a result, integrating along the depth direction, the coarse mesh with bigger interface thickness by nature gives higher pure hydrostatic pressure. The refinement study shows that the simulation results of the immersed method gradually converge to the results from boundary-fitted mesh as we refine the mesh.

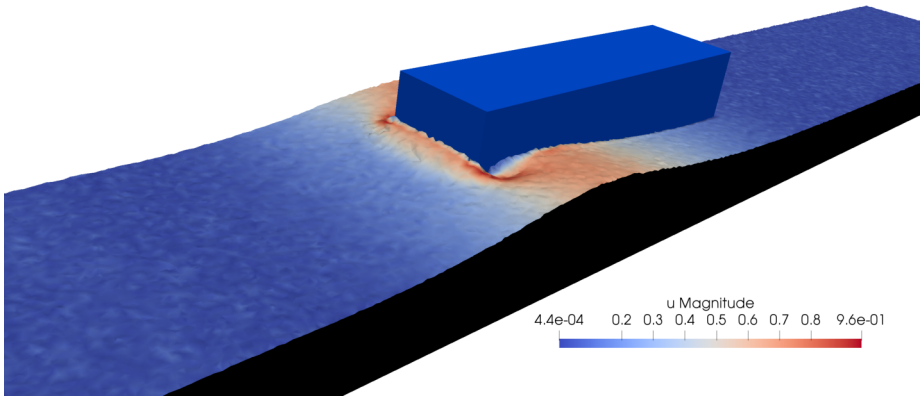
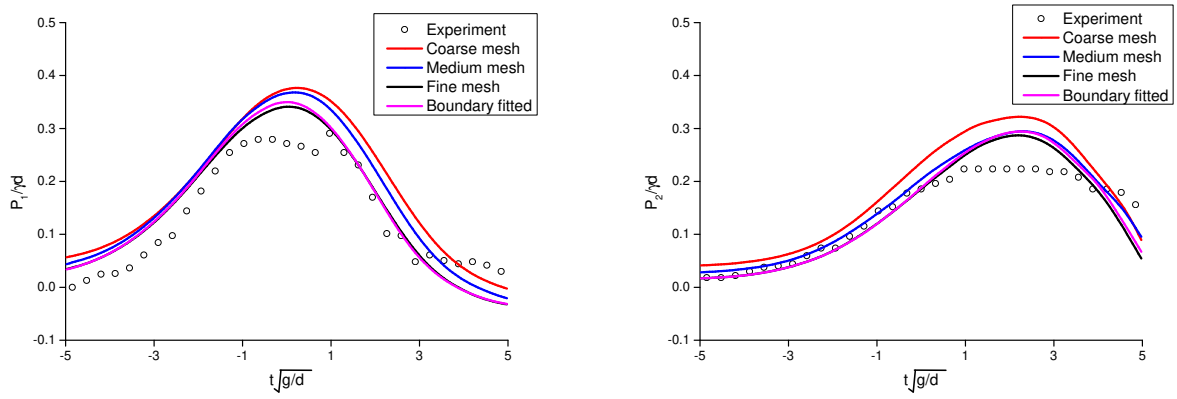


Figure 7: Free-surface deformation colored by velocity magnitude (m/s)



(a) Point 1 (b) Point 2

Figure 8: Time history of normalized pressure ($\gamma = \rho g$)

6.2. Dam break with an obstacle

The dam break case investigates how a column of water, initially at rest, collapses due to gravity and impacts a stationary obstacle. The simulation domain is a rectangular box

with dimensions 3.22 m × 1.0 m × 1.0 m. The water column with dimensions 1.22 m × 1.0 m × 0.5 m initially locates on the left of the domain, with a distance of 2.3955 m from the center of a stationary obstacle with dimensions 0.403 m × 0.161 m × 0.161 m. The computational setup of this problem is shown in Fig. 9. The region around the obstacle is refined to capture the pressure. The element length of the mesh, and the number of nodes and elements are given in Table 3 and Table 4. Fig. 10 shows the mesh in the central plane, Fig. 11 shows a zoom-in view of the mesh. The initial and boundary conditions are set as follows. For initialization, zero velocity is used, and the level set function is defined based on the signed distance with respect to the initial free-surface of the water column. No-penetration boundary condition is set strongly for all the boundaries of the simulation domain, while no-penetration boundary condition is applied weakly, based on Eq. (38), for the fluid-obstacle interface. $\Delta t = 0.0005$ s is used for this case. Simulation is performed until $t = 6.0$ s. Fig. 12 shows the free-surface at $t = 0.50$ s, $t = 1.25$ s, $t = 1.75$ s, and $t = 4.75$

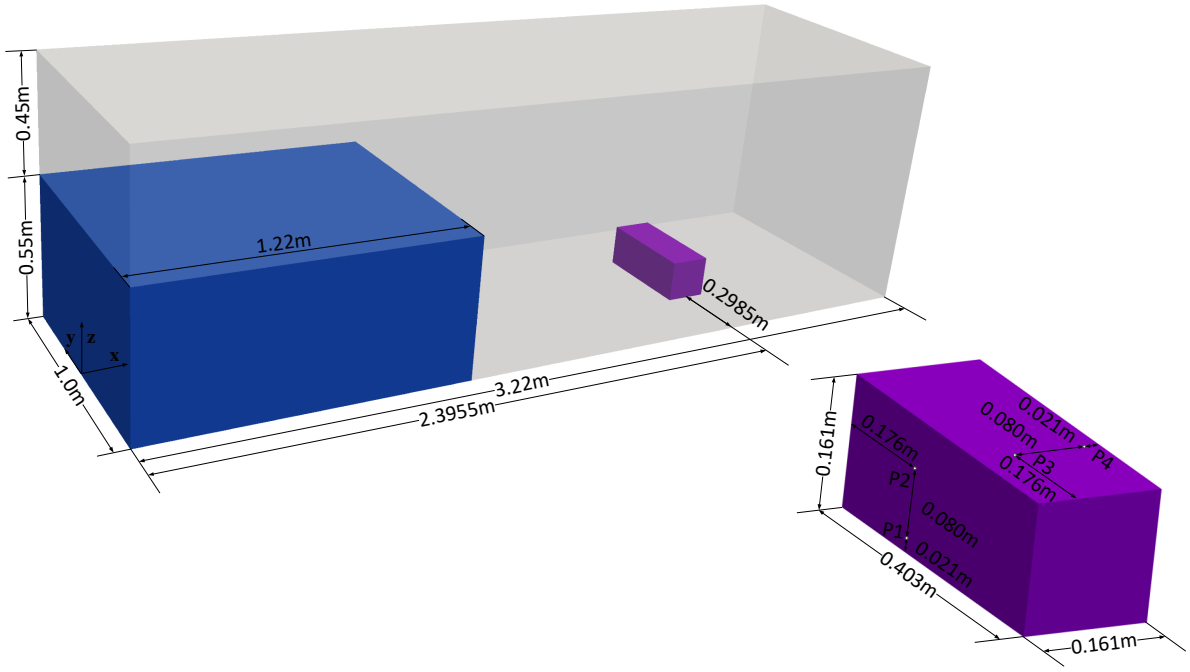


Figure 9: Computational setup of dam break with obstacle

s. When the water hits the obstacle, the free-surface evolution is more violent compared with the previous solitary wave case. After impacting the outlet wall of the tank, the water runs up the back wall quickly and even touches the top of the tank. At the later stages of the simulation, wave breaking occurs. These free-surface features are also observed in the experiments reported in [75].

We report the time history of the pressure at four points on the obstacle in Fig. 13. The

Table 3: Element length of the mesh in the dam break case

Near obstacle box	Near outer boundary
0.0045 m	0.03 m

Table 4: Number of elements and nodes of the mesh in the dam break case

Number of elements	Number of nodes
1,461,086	241,252

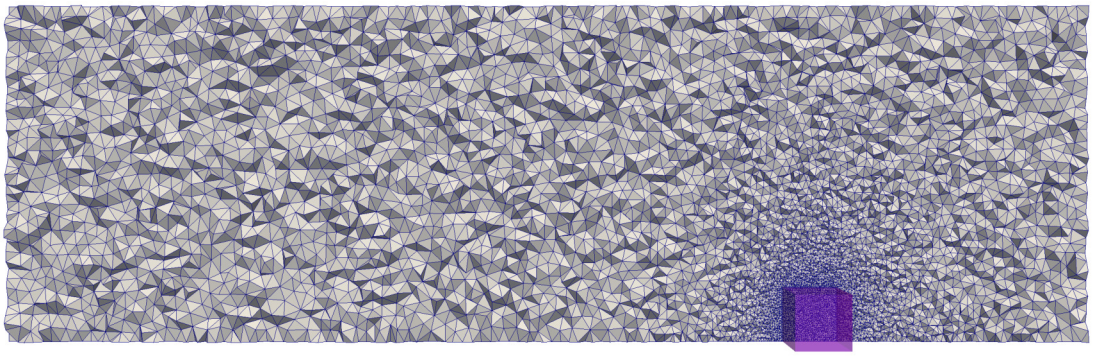


Figure 10: Mesh of the dam break case in central plane

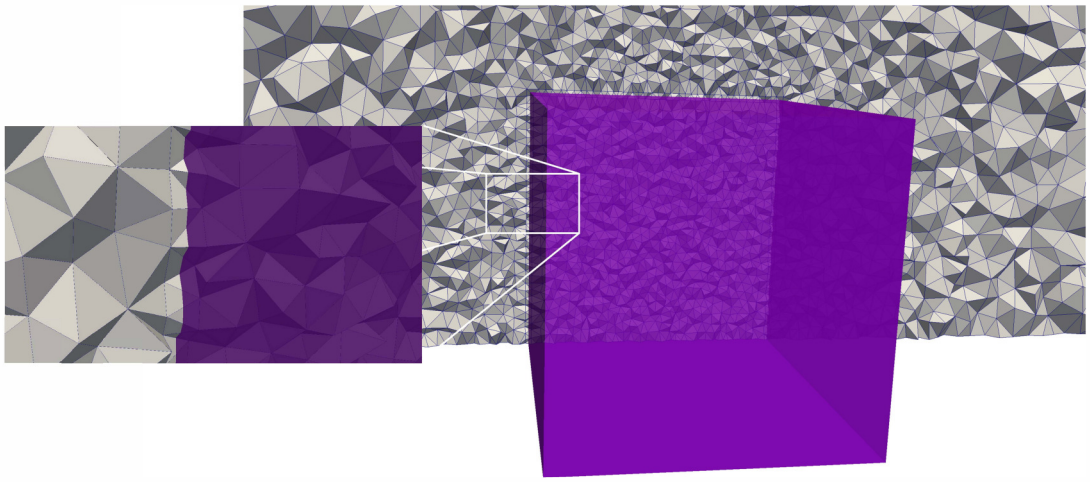


Figure 11: Zoom in of mesh (dam break case) in central plane

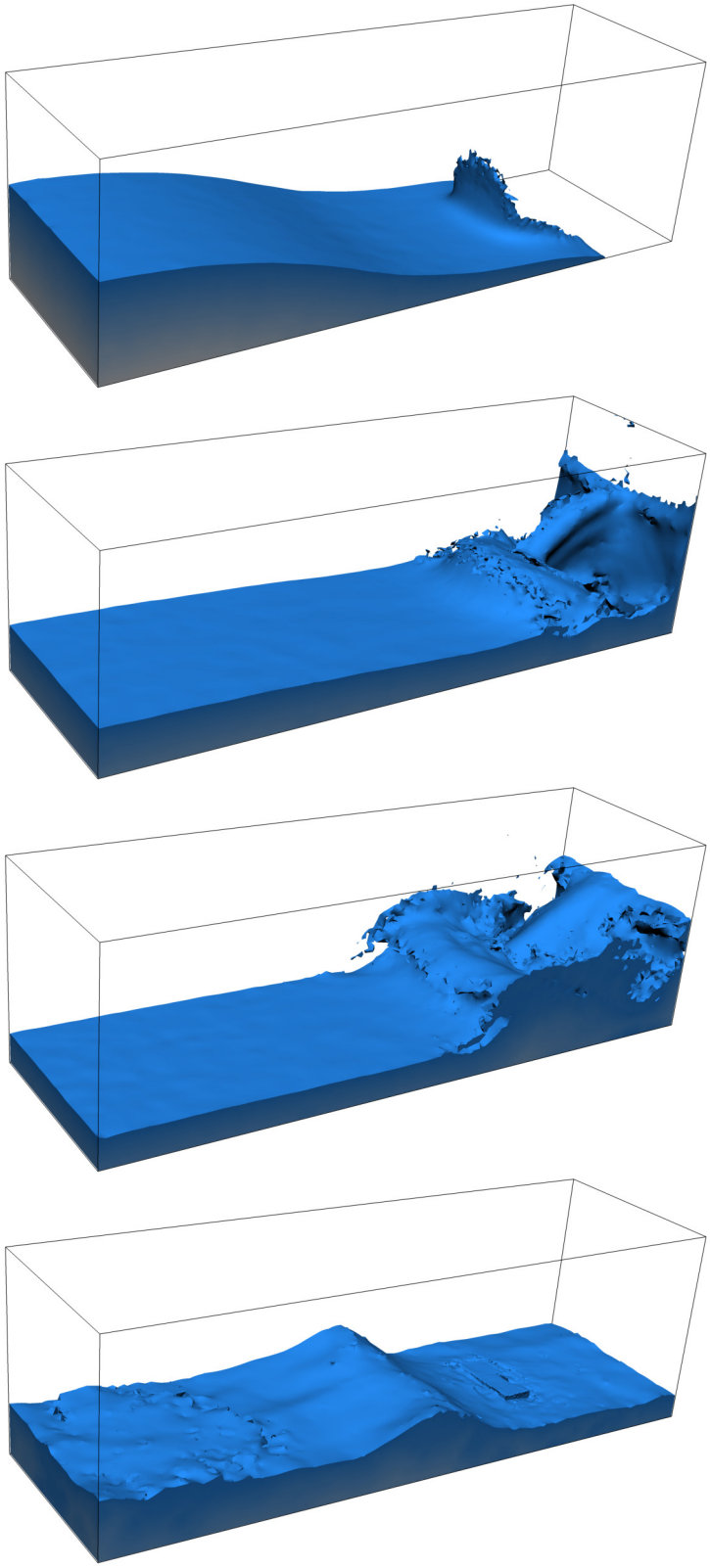
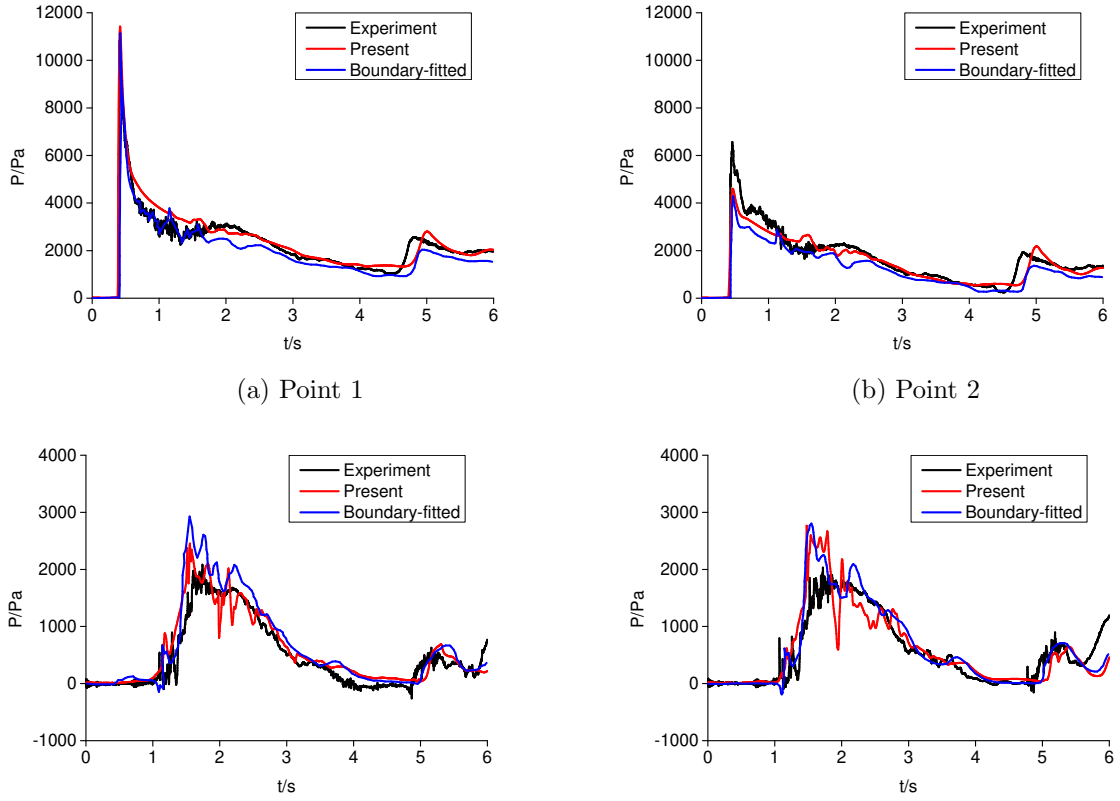


Figure 12: Free-surface deformation of the dam break simulation ($t = 0.50$ s, 1.25 s, 1.75 s, and 4.75 s, from the top to the bottom)

location of the four points is shown in Fig. 9. Experiments data from Maritime Research Institute Netherlands (MARIN) [75] and computational results based on a boundary-fitted approach [40] are also plotted to validate the simulated results in the present work. The coordinate of P1, P2, P3 and P4 are (2.315 m, 0.0255 m, 0.021 m), (2.315 m, 0.0255 m, 0.101 m), (2.375 m, -0.0255 m, 0.161 m) and (2.455 m, -0.0255 m, 0.161 m), respectively. For P1 and P2, excellent agreement is achieved. Although the computational results of both boundary-fitted approach and present immersed approach deviate from the experiment measurement for P3 and P4, this comparison shows that the immersogeometric approach can at least produce the same level of accuracy as boundary-fitted approach for this problem.



(a) Point 1

(b) Point 2

(c) Point 3

(d) Point 4

Figure 13: Pressure time history of the dam break case

6.3. Planing of a DTMB 5415 ship model

In this section, the planing of the David Taylor Model Basin (DTMB) 5415 ship model is simulated by using the proposed immersed formulation.

Fig. 14 shows the CAD model of the DTMB 5415 bare ship. For the geometry details, the readers are referred to [76]. The length of the model L is 5.72 m. The draft T is 0.248 m. The Froude number Fr is 0.28.

Fig. 15 shows the computational setup. The simulation domain is a box with dimensions $3L \times L \times L$. Fig. 16 shows the mesh in the central plane, Fig. 17 shows a zoom-in view of the mesh. The origin of the coordinate system locates at the intersection of the bow with the still free surface. The mesh is refined around the air-water interface and the ship. The element length employed in the mesh, the number of elements and nodes are summarized in Table 5 and Table 6. The boundary conditions are defined as follows. Uniform water speed and zero air speed are applied strongly for inflow, hydrostatic pressure condition is used for outflow, free-slip and no-penetration condition is applied strongly for side boundaries, and no-slip boundary condition is applied weakly for the fluid-ship interface. The time step Δt is set to 0.003 s for this case. The simulation is performed until no noticeable free-surface change is observed (quasi-static stage).

Table 5: Element length employed in the mesh of the DTMB 5415 ship case

	Ship and free-surface	Top boundary	Bottom boundary
Mesh size	0.012 m	0.174 m	0.350 m

Table 6: Number of elements and nodes of the mesh of the DTMB 5415 ship case

Number of elements	Number of nodes
15,272,253	2,788,077

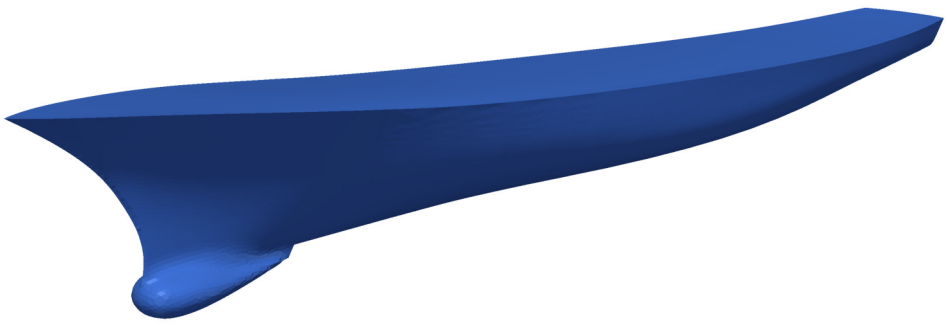


Figure 14: Geometry of DTMB 5415 ship hull

Fig. 18 shows the free-surface colored by the water elevation from two view angles. The wave profile looks symmetric with respect to the centerline at the scale of the figure. Fig. 19 shows the wave heights normalized by L along the center line and along the line of $y/L =$

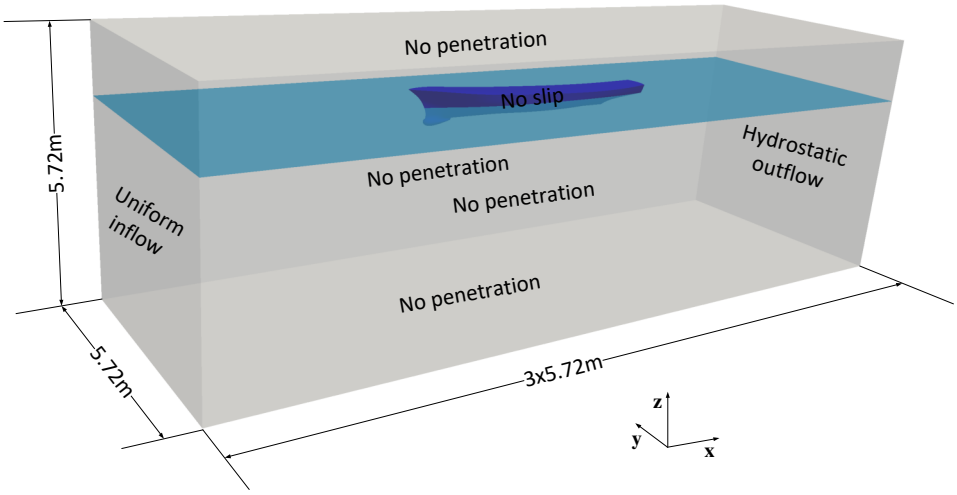


Figure 15: Computational setup of the DTMB 5415 ship case

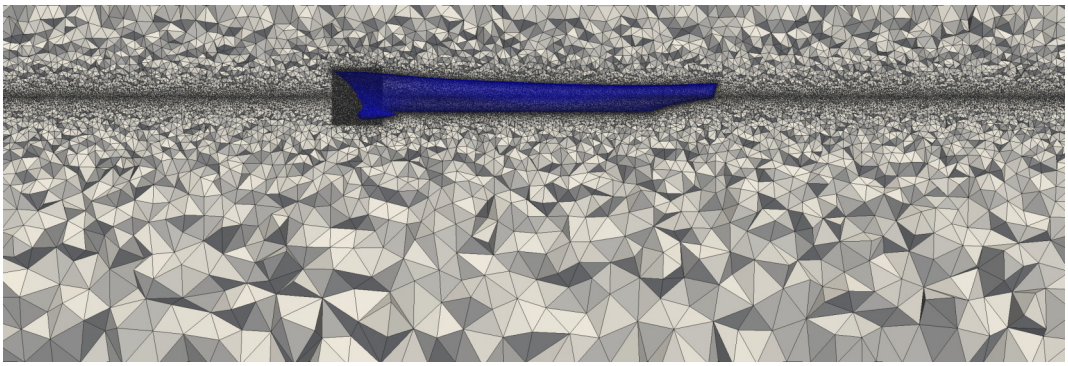


Figure 16: Mesh of of the DTMB 5415 ship case in the central plane

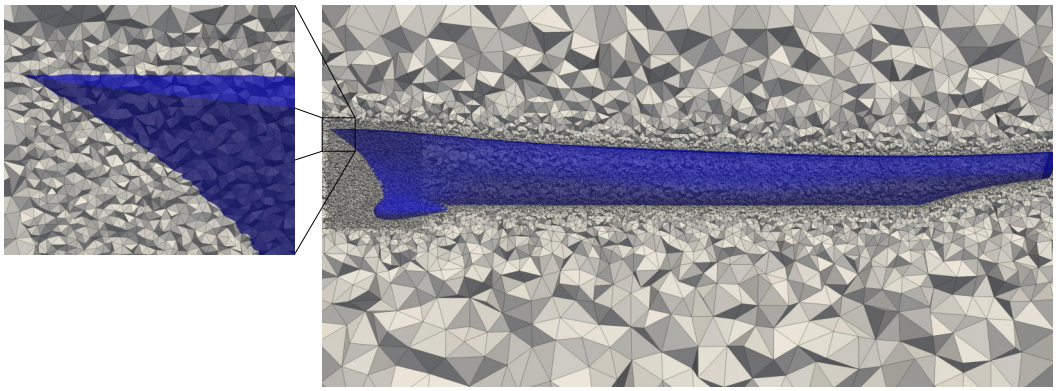


Figure 17: Zoom in of mesh (DTMB 5415 ship case) in the central plane

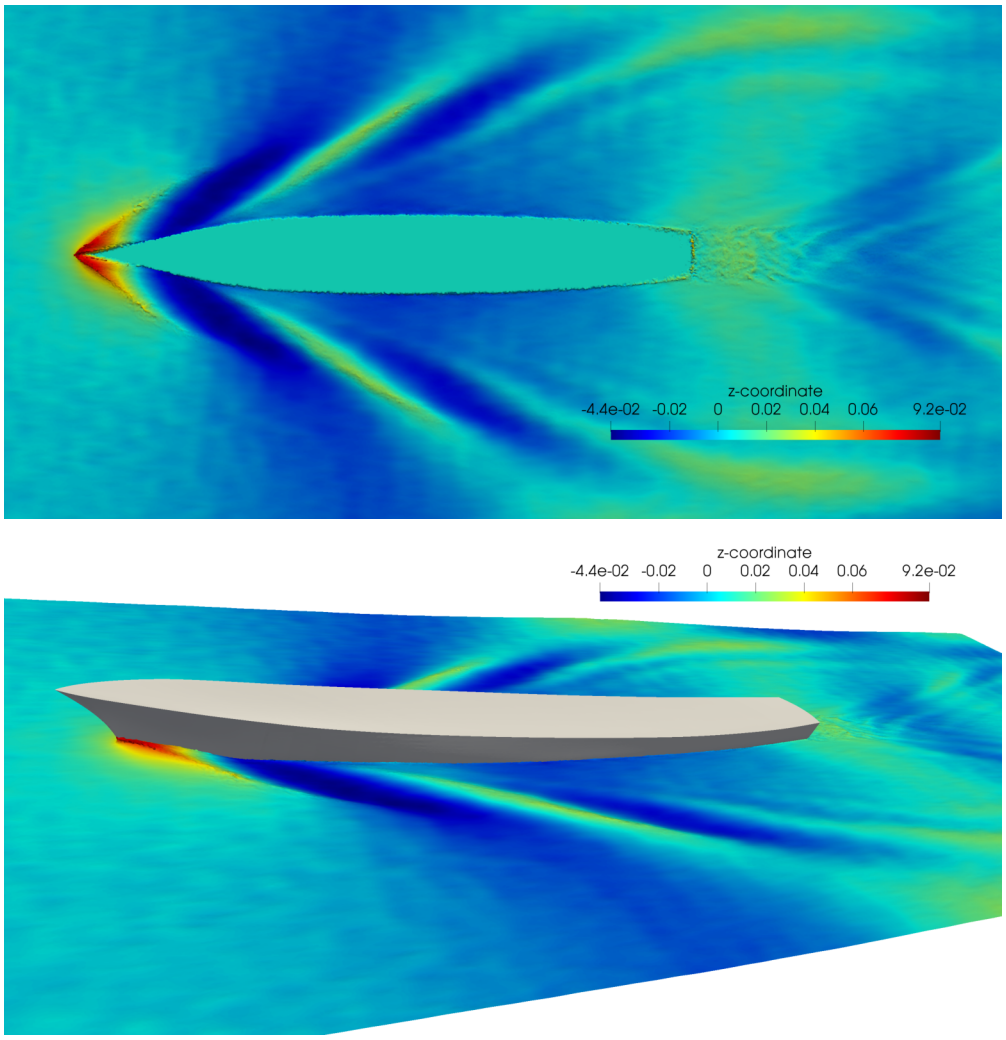
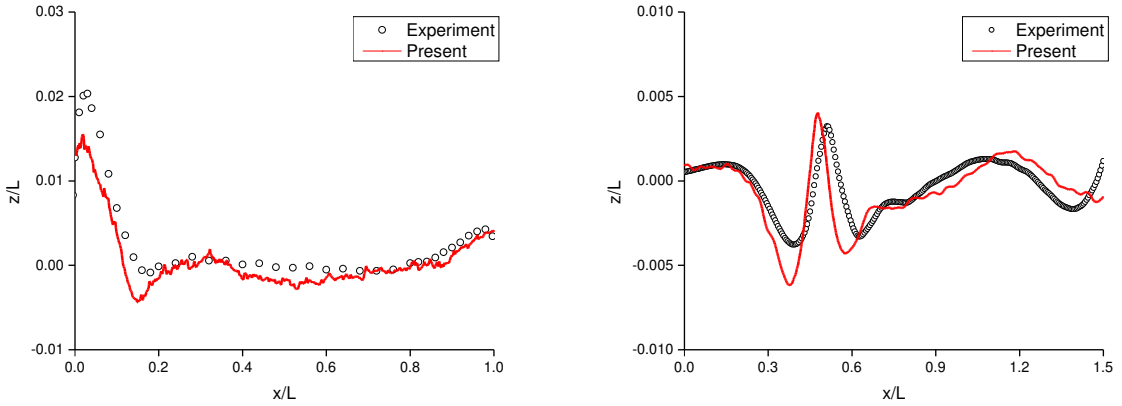


Figure 18: Free-surface colored by wave height (m) at the quasi-static stage from two different view angles



(a) Wave height along the center line

(b) Wave height along line at $y/L = 0.172$

Figure 19: Wave height comparison of the DTMB 5415 ship hull case

0.172, respectively. The experimental data from [77] is plotted for comparison. A close agreement is achieved again that indicates the accuracy of the proposed formulation. The small discrepancy between simulation results and experiment results could be explained as follows. The experiment is conducted in a wide wave tank. To save the computational cost, we adopt a truncated simulation domain with no penetration boundary condition, which may have a slight side wall effect. The mesh resolution near the ship boundary may not be fine enough to capture all the details.

7. Summary and future work

We developed an immersogeometric formulation for free-surface flow simulations around complex geometry by integrating level set method, residual-based variational multi-scale formulation, and finite cell method. The Dirichlet boundary condition on the fluid-structure interface is enforced by a weak formulation. The FCM-based adaptive quadrature is employed to better resolve the immersed structure boundary. The octree-based ray-tracing method is used to perform the in-out test for complex geometry. The accuracy of the proposed formulation is assessed by simulating three challenging marine engineering problems. Computational results agree well with experimental data and computational results from boundary-fitted methods. Together with its high accuracy, the flexibility of the method facilitates high-quality analysis of marine structures with complicated geometry in free-surface flows by circumventing labor-intensive volumetric meshing step. In the future, we plan to include adaptive mesh refinement around free-surface and cavitation model in this immersogeometric formulation.

Acknowledgment

The simulations are performed at the Blue Waters Supercomputer at the University of Illinois at Urbana-Champaign. This support is gratefully acknowledged.

References

[1] A. Prosperetti and G. Tryggvason. *Computational methods for multiphase flow*. Cambridge University Press, 2009.

[2] T. E. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8:83–130, 2001.

[3] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29:329–349, 1981.

[4] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1):25–37, 1992.

[5] J. P. Best. The formation of toroidal bubbles upon the collapse of transient cavities. *Journal of Fluid Mechanics*, 251:79–107, 1993.

[6] T. E. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94(3):339–351, 1992.

[7] H. Braess and P. Wriggers. Arbitrary Lagrangian Eulerian finite element analysis of free surface flow. *Computer Methods in Applied Mechanics and Engineering*, 190(1-2): 95–109, 2000.

[8] Z. Gan, G. Yu, X. He, and S. Li. Numerical simulation of thermal behavior and multicomponent mass transfer in direct laser deposition of co-base alloy on steel. *International Journal of Heat and Mass Transfer*, 104:28–38, 2017.

[9] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1):146–159, 1994.

[10] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79 (1):12–49, 1988.

[11] E. Shirani, N. Ashgriz, and J. Mostaghimi. Interface pressure calculation based on conservation of momentum for front capturing methods. *Journal of Computational Physics*, 203(1):154–175, 2005.

[12] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[13] D. Jacqmin. Calculation of two-phase Navier–Stokes flows using phase-field modeling. *Journal of Computational Physics*, 155(1):96–127, 1999.

[14] J. Liu. *Thermodynamically consistent modeling and simulation of multiphase flows*. PhD thesis, The University of Texas at Austin, 2014.

[15] P. Yue, J. J. Feng, C. Liu, and J. Shen. A diffuse-interface method for simulating two-phase flows of complex fluids. *Journal of Fluid Mechanics*, 515:293–317, 2004.

[16] L. Amaya-Bower and T. Lee. Single bubble rising dynamics for moderate Reynolds number using Lattice Boltzmann Method. *Computers & Fluids*, 39(7):1191–1207, 2010.

[17] S. Nagrath, K. E Jansen, and R. T. Lahey. Computation of incompressible bubble dynamics with a stabilized finite element level set method. *Computer Methods in Applied Mechanics and Engineering*, 194(42):4565–4587, 2005.

[18] M. K. Tripathi, K. C. Sahu, and R. Govindarajan. Dynamics of an initially spherical bubble rising in quiescent liquid. *Nature communications*, 6, 2015.

[19] M. van Sint Annaland, N. G. Deen, and J. A. M. Kuipers. Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method. *Chemical Engineering Science*, 60(11):2999–3011, 2005.

[20] J. M. Gimenez, N. M. Nigro, S. R. Idelsohn, and E. Oñate. Surface tension problems solved with the particle finite element method using large time-steps. *Computers & Fluids*, 2016.

[21] R. Calderer, L. Zhu, R. Gibson, and A. Masud. Residual-based turbulence models and arbitrary Lagrangian–Eulerian framework for free surface flows. *Mathematical Models and Methods in Applied Sciences*, 25(12):2287–2317, 2015.

[22] L. Zhu, S. Goraya, and A. Masud. A stabilized interface capturing method for large amplitude breaking waves. *Journal of Engineering Mechanics*, 2019.

[23] T. E. Tezduyar, S. Sathe, J. Pausewang, M. Schwaab, J. Christopher, and J. Crabtree. Interface projection techniques for fluid–structure interaction modeling with moving-mesh methods. *Computational Mechanics*, 43:39–49, 2008. doi: 10.1007/s00466-008-0261-7.

[24] A. A. Johnson and T. E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119:73–94, 1994. doi: 10.1016/0045-7825(94)00077-8.

[25] M. C. Hsu and Y. Bazilevs. Fluid-structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, DOI: 10.1007/s00466-012-0772-0, 2012.

[26] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

[27] C. S. Peskin. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.

[28] W. K. Liu, Y. Liu, D. Farrell, L. Zhang, X. S. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, J. Lee, et al. Immersed finite element method and its applications to biological systems. *Computer Methods in Applied Mechanics and Engineering*, 195(13-16):1722–1749, 2006.

[29] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part II: Linear advection–diffusion and incompressible Navier–Stokes equations. *Journal of Computational Physics*, 372:996–1026, 2018.

[30] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems. *Journal of Computational Physics*, 372: 972–995, 2018.

[31] Y. Bazilevs, K. Kamran, G. Moutsanidis, D. J. Benson, and E. Oñate. A new formulation for air-blast fluid–structure interaction using an immersed approach. Part I: basic methodology and FEM-based simulations. *Computational Mechanics*, 60(1):83–100, 2017.

[32] Y. Bazilevs, G. Moutsanidis, J. Bueno, K. Kamran, D. Kamensky, M. C. Hillman, H. Gomez, and J. S. Chen. A new formulation for air-blast fluid–structure interaction using an immersed approach: Part II—coupling of IGA and meshfree discretizations. *Computational Mechanics*, 60(1):101–116, 2017.

[33] H. Casquero, Y. Zhang, C. Bona-Casas, L. Dalcin, and H. Gomez. Non-body-fitted fluid–structure interaction: Divergence-conforming b-splines, fully-implicit dynamics, and variational formulation. *Journal of Computational Physics*, 374:625–653, 2018.

[34] Y. Yu, D. Kamensky, M. Hsu, X. Lu, Y. Bazilevs, and Thomas JR T.J.R. Hughes. Error estimates for projection-based dynamic augmented lagrangian boundary condition enforcement, with application to fluid–structure interaction. *Mathematical Models and Methods in Applied Sciences*, 28(12):2457–2509, 2018.

[35] D. Kamensky, M. C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.

[36] D. Kamensky, M. C. Hsu, Y. Yu, J. A. Evans, M. S. Sacks, and T. J. R. Hughes. Immersogeometric cardiovascular fluid–structure interaction analysis with divergence-conforming B-splines. *Computer Methods in Applied Mechanics and Engineering*, 314: 408–472, 2017.

[37] J. Yan, X. Deng, A. Korobenko, and Y. Bazilevs. Free-surface flow modeling and simulation of horizontal-axis tidal-stream turbines. *Computers & Fluids*, 2016.

[38] J. Yan. *Computational free-surface fluid-structure interaction with applications on offshore wind and tidal energy*. PhD thesis, University of California San Diego, 2016.

[39] J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs. Computational free-surface fluid–structure interaction with application to floating offshore wind turbines. *Computers & Fluids*, 141:155–174, 2016.

[40] I. Akkerman, Y. Bazilevs, C. E. Kees, and M. W. Farthing. Isogeometric analysis of free-surface flow. *Journal of Computational Physics*, 230:4137–4152, 2011.

[41] I. Akkerman, Y. Bazilevs, D. J. Benson, M. W. Farthing, and C. E. Kees. Free-surface flow and fluid–object interaction modeling with emphasis on ship hydrodynamics. *Journal of Applied Mechanics*, 79:010905, 2012.

[42] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis. Toward Integration of CAD and FEA*. Wiley, 2009.

[43] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[44] M. C. Hsu, C. Wang, F. Xu, A. J. Herrema, and A. Krishnamurthy. Direct immersogeometric fluid flow analysis using B-rep CAD models. *Computer Aided Geometric Design*, 43:143–158, 2016.

[45] M. C. H. Wu, R. Zakerzadeh, D. Kamensky, J. Kiendl, M. S. Sacks, and M. C. Hsu. An anisotropic constitutive model for immersogeometric fluid–structure interaction analysis of bioprosthetic heart valves. *Journal of Biomechanics*, 2017.

[46] M. C. Hsu, D. Kamensky, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. *Computational Mechanics*, 54:1055–1071, 2014.

[47] F. Xu, S. Morganti, R. Zakerzadeh, D. Kamensky, F. Auricchio, A. Reali, T. J. R. Hughes, M. S. Sacks, and M. C. Hsu. A framework for designing patient-specific bioprosthetic heart valves using immersogeometric fluid–structure interaction analysis. *International Journal for Numerical Methods in Biomedical Engineering*, 34(4):e2938, 2018.

[48] F. Xu, Y. Bazilevs, and M. C. Hsu. Immersogeometric analysis of compressible flows with application to aerodynamic simulation of rotorcraft. *Mathematical Models and Methods in Applied Sciences*, 29(05):905–938, 2019.

[49] J. Parvizian, A. Düster, and E. Rank. Finite cell method: h- and p- extension for embedded domain methods in solid mechanics. *Computational Mechanics*, 41:122–133, 2007.

[50] A. Düster, J. Parvizian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197(45-48):3768–3782, 2008.

[51] S. Xu, F. Xu, A. Kommajosula, M. C. Hsu, and B. Ganapathysubramanian. Immersogeometric analysis of moving objects in incompressible flows. *Computers & Fluids*, 189: 24–33, 2019.

[52] F. Xu, D. Schillinger, D. Kamensky, V. Varduhn, C. Wang, and M. C. Hsu. The tetrahedral finite cell method for fluids: Immersogeometric analysis of turbulent flow around complex geometries. *Computers & Fluids*, 2015.

[53] Y. Bazilevs, V. M. Calo, J. A. Cottrel, T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197: 173–201, 2007.

[54] S. Xu, N. Liu, and J. Yan. Residual-based variational multi-scale modeling for particle-laden gravity currents over flat and triangular wavy terrains. *Computers & Fluids*, 188: 114–124, 2019.

[55] J. Yan, A. Korobenko, A. E. Tejada-Martínez, R. Golshan, and Y. Bazilevs. A new variational multiscale formulation for stratified incompressible turbulent flows. *Computers & Fluids*, 2016.

[56] T. M. van Opstal, J. Yan, C. Coley, J. A Evans, T. Kvamsdal, and Y. Bazilevs. Isogeometric divergence-conforming variational multiscale formulation of incompressible turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 2016.

[57] J. Yan, S. Lin, Y. Bazilevs, and G. J. Wagner. Isogeometric analysis of multi-phase flows with surface tension and with application to dynamics of rising bubbles. *Computers & Fluids*, 179:777–789, 2019.

[58] J. Yan, W. Yan, S. Lin, and G. J. Wagner. A fully coupled finite element formulation for liquid–solid–gas thermo-fluid flow with melting and solidification. *Computer Methods in Applied Mechanics and Engineering*, 336:444–470, 2018.

[59] Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36:12–26, 2007.

[60] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[61] T. E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics*, 28:1–44, 1992.

[62] T. E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190:411–430, 2000.

[63] T. J. R. Hughes, L. Mazzei, A. A. Oberai, and A. Wray. The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. *Physics of Fluids*, 13:505–512, 2001.

[64] T. J. R. Hughes, G. Scovazzi, and L. P. Franca. Multiscale and stabilized methods. In *Encyclopedia of Computational Mechanics*. John Wiley & Sons, 2004.

[65] M. C. Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199:828–840, 2010.

[66] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Sweden, 1987.

[67] T. E. Tezduyar. Finite element methods for fluid dynamics with moving boundaries and interfaces. *Encyclopedia of Computational Mechanics*, 2004.

[68] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36:9–15, 1971.

[69] V. Varduhn, M. C. Hsu, M. Ruess, and D. Schillinger. The tetrahedral finite cell method: Higher-order immersogeometric analysis on adaptive non-boundary-fitted meshes. *International Journal for Numerical Methods in Engineering*, 2015.

[70] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of Applied Mechanics*, 60(2):371–375, 1993.

[71] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190(3-4):305–319, 2000.

[72] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. *Computational fluid-structure interaction: methods and applications*. John Wiley & Sons, 2013.

[73] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7: 856–869, 1986.

[74] J. A. French. *Wave uplift pressures on horizontal platforms*. PhD thesis, California Institute of Technology, 1970.

[75] K. M. T. Kleefsman, G. Fekken, A. E. P. Veldman, B. Iwanowski, and B. Buchner. A volume-of-fluid based simulation method for wave impact problems. *Journal of Computational Physics*, 206(1):363–393, 2005.

[76] Geometry of DTMB 5415 model. http://www.simman2008.dk/5415/5415_geometry.htm, 2008.

[77] J. Longo and F. Stern. Uncertainty assessment for towing tank tests with example for surface combatant DTMB model 5415. *Journal of Ship Research*, 49(1):55–68, 2005.