

## **A Direct Forcing Immersion Boundary Method for Fluid-Structure Interaction under an Adaptive Mesh Framework**

*Yuming Shao<sup>1</sup>, Wentao Wang<sup>1,2</sup>, Jianhua Wang<sup>1</sup>, Decheng Wan<sup>1\*</sup>*

<sup>1</sup> Computational Marine Hydrodynamics Lab (CMHL), School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai, China

<sup>2</sup> China Ship Scientific Research Center, Wuxi, China

\*Corresponding Author

### **ABSTRACT**

Numerical simulations in the ship and ocean engineering field often require consideration of the interactions between structural boundaries and fluid dynamics. In this study, a direct-forcing immersed boundary method (DF-IBM) is implemented within the open-source adaptive mesh solver Basilisk for the first time. By integrating IBM with geometric reconstruction and Level Set tracking, our approach addresses the challenging issue of capturing gas-solid-liquid three-phase interactions. The proposed method is validated through a range of test cases: flow around a cylinder (single-phase flow), a transversely oscillating cylinder (simple motion), water entry and exit of a cylinder (simple motion, simple two-phase flow), and wave-breaking phenomena around a wedge-shaped structure (complex two-phase flow). The results confirm the method's accuracy and reliability across these cases. This work provides a robust foundation for future hydrodynamic simulations involving complex structural motions in marine applications.

**KEY WORDS:** Immersed boundary method, direct forcing, fluid-structure interaction, wedge-shaped bow, wave breaking.

### **INTRODUCTION**

The interaction between two-phase flows and impermeable solid objects is a common phenomenon in both nature and engineering, such as the interaction between ships and water. Accurately simulating the presence and motion of solid objects using numerical methods is highly complex due to the involvement of turbulent boundary layers, surface breaking, and gas-liquid-solid interactions.

Currently, three primary numerical approaches are employed to model fluid-solid interactions: meshless methods, body-fitted mesh methods, and non-body-fitted mesh methods. In meshless methods, interactions between structures and fluids are modeled by assigning discrete particles with distinct physical properties, as seen in methods like Smoothed

Particle Hydrodynamics (SPH) and Moving Particle Semi-implicit (MPS). While these methods have garnered significant attention in recent years, their application to large-scale computations remains challenging due to the high computational cost and algorithmic complexity. In traditional body-fitted mesh methods, the mesh within the object is removed, and the mesh near the interface is reconstructed to conform closely to the object's surface. The quality of the mesh is crucial to the accuracy of the flow field. When the object is in motion, dynamic mesh or overset mesh techniques are often used, requiring mesh reconstruction at each time step. For objects with large motion amplitudes, this approach becomes computationally expensive and prone to divergence. Non-body-fitted mesh methods typically employ the Immersed Boundary Method (IBM). The core of this approach is to model the effects of complex boundaries as body forces in the Navier-Stokes (N-S) equations, thereby circumventing the challenges associated with generating body-fitted meshes. IBM imposes no restrictions on object motion or shape changes, making it particularly suitable for cases involving large motions or deformations of complex objects (Mittal and Seo, 2023).

The IBM is initially proposed by Peskin to simulate the interaction between blood flow and heart valves (Peskin, 1972). Over decades of development, it has been widely applied to fluid-structure interaction and interface modeling. Based on how boundary conditions are imposed, IBM can be categorized into two main types: diffuse-interface methods and sharp-interface methods. In diffuse-interface methods, the influence of the interface is spread into the surrounding flow field in the Eulerian grid using a delta function, leading to approximate satisfaction of the boundary conditions. These methods include the feedback force method (Goldstein et al., 1993), penalty function method (Kim and Peskin, 1993), and virtual spring force method (Huang and Sung, 2010), and are better suited for deformable structure motion.

Sharp-interface methods, on the other hand, impose boundary conditions through forcing points, preserving the sharpness of the interface. These methods are more appropriate for rigid body motion under high

Reynolds numbers. Typical examples include the cut-cell method, ghost cell method, and hybrid Cartesian/immersed boundary method. The cut-cell method, also known as the embedded boundary method (Ghigo et al., 2021; Limare et al., 2023), is ideologically analogous to the body-fitted mesh approach. It involves cutting the computational cells containing the interface and retaining the fluid portion. This method achieves higher computational efficiency and accuracy compared to body-fitted meshes by controlling fluxes at the interface. However, it has certain drawbacks: (1) the interface distribution and cutting within the mesh can vary, complicating its extension to three-dimensional problems; (2) cells with a small fluid fraction can significantly degrade computational efficiency and stability. The ghost cell method and hybrid Cartesian/immersed boundary method share similar mechanisms, differing primarily in how forcing points are constructed. In the former, forcing points are created in the solid region, whereas in the latter, they are placed in the fluid. The hybrid Cartesian/immersed boundary method (Gilmanov and Sotiropoulos, 2005), also known as the direct forcing method, applies boundary conditions as body forces directly to the right-hand side of the discretized N-S equations. Its simplicity and ease of implementation have made it widely adopted by researchers (Fadlun et al., 2000; Vanella et al., 2010).

Yang and Stern (2009), building on the method introduced by Balaras (2004) developed a direct forcing framework for the Immersed Boundary Method (DF-IBM) to simulate strongly coupled fluid-structure interactions. By incorporating the direct forcing approach into a fractional step method, they validated its accuracy through benchmark cases, including flow around a cylinder, vortex-induced vibrations of a square cylinder, and the rolling motion of a rectangular plate. Furthermore, Yang and Stern (2012) integrated this framework with the Level-Set/Ghost-Fluid Method for two-phase flow simulations. They successfully simulated the free surface flows of Wigley and DTMB 5512 hulls at low Froude numbers using coarse grids and demonstrated consistency with results obtained from body-fitted solvers, thus confirming the method's reliability. Kan et al. (2021, 2023) employed a sharp-interface, level-set-based IBM to capture complex pump geometries and simulate the near-water rotational behavior of propellers, investigating the turbulent wake structures. Liu and Hu (2014, 2017) combined a high-order conservative IBM with block-based adaptive mesh refinement, extending its applicability to both compressible and incompressible flows. Later, Hu et al. (2021) applied this method to simulate free surface breaking induced by inclined plates, accurately capturing bow wave breaking and air entrainment phenomena under various inclination angles. Li et al. (2021) utilized this approach to simulate flow around hydrofoils penetrating the free surface, successfully reproducing the hydraulic jump phenomenon at the hydrofoil's wake.

The Basilisk flow solver (Popinet 2003, 2009), an open-source framework designed for incompressible flow with adaptive mesh refinement, is used in this study. The solver already integrates the embedded boundary method (Ghigo et al., 2021). Shao et al. (2024) employed it to simulate free surface breaking induced by hydrofoils at different submergence depths, analyzing phenomena such as bubble entrainment. Similarly, Guo et al. (2023) modeled flow around rotating cylinders using this solver. However, the method faces certain limitations, including a lack of support for OpenMPI parallelization and its inability to handle simultaneous solid-liquid-gas interactions within a single grid cell. While Tavares et al (2024) extended the solver's application to two-phase flows by incorporating contact angle simulations, large-scale, multi-scale two-phase simulations remain challenging. Sharaborin et al (2021) improved the solver's capability to simulate solid interfaces by integrating the penalty function method. Additionally, Huet and Wachs (2023) coupled the front-tracking method

with a Peskin-type IBM within the solver framework to study elastic capsules. Cheng et al. (2022) combined the lattice Boltzmann method with the feedback forcing method to investigate particle-laden flows. Despite these advances, existing IBMs still lack sharpness in handling interfaces. The ultimate goal of this research is to enable hydrodynamic simulations of ship-water interactions that account for structural effects. To this end, we propose a direct forcing method for fluid-structure interactions within the Basilisk solver framework.

This study aims to implement the direct forcing Immersed Boundary Method within an adaptive mesh framework. The structure of this paper is organized as follows: the numerical methods, including governing equations, interface capturing techniques, and the direct forcing model, are introduced first. Subsequently, the method is validated through three benchmark cases: flow around a stationary cylinder at various Reynolds numbers, motion of a moving cylinder, and water entry of a cylinder. Simulations and analyses of wave-breaking phenomena around wedge-shaped structures are then conducted. Finally, the study's findings are summarized.

## NUMERICAL METHOD

### Governing Equations

For incompressible flows, the governing equation can be described as:

$$\rho(\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u}) = -\nabla p + \nabla \cdot (2\mu \mathbf{D}) + \mathbf{a} + \mathbf{f}_B \quad (1)$$

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

Where  $\mathbf{u}$  is the velocity of the fluid,  $\rho$  is the density of the fluid,  $p$  represents the pressure,  $\mu$  is the dynamic viscosity coefficient,  $\mathbf{D}$  is the deformation coefficient, Defined as  $D_{ij} \equiv (\partial_i u_j + \partial_j u_i)/2$ .  $\mathbf{a} = \rho \mathbf{g} + \mathbf{f}_\sigma$ ,  $\mathbf{g}$  represents the acceleration of gravity,  $\mathbf{f}_\sigma$  represents the action of surface tension, and  $\mathbf{f}_B$  represents the boundary conditions under which the DF-IBM method is implemented.

If the  $\mathbf{f}_B$  source term is not considered, the discrete governing equation can be expressed as:

$$\rho^{n+\frac{1}{2}} \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \nabla \cdot (\mathbf{u}^{n+\frac{1}{2}} \otimes \mathbf{u}^{n+\frac{1}{2}}) = \nabla \cdot [2\mu^{n+\frac{1}{2}} \mathbf{D}^*] + \mathbf{a}^n - \nabla p^n \quad (3)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho^{n+\frac{1}{2}}} (\nabla p^n) \quad (4)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (5)$$

The pressure-velocity Poisson equation is obtained by combining Eq. 4 and Eq. 5:

$$\nabla \cdot \left( \frac{\Delta t}{\rho^{n+\frac{1}{2}}} \nabla p^{n+1} \right) = \nabla \cdot \mathbf{u}^* \quad (6)$$

### Interface Capturing Method

The free liquid surface is captured by geometric reconstruction VOF-PLIC method (Fuster et al., 2018) and the normal direction of the interface is determined by the Mixed-Youngs-Centered (MYC) method (Youngs, 1982).

$$\partial_t \varphi + \nabla \cdot (\varphi \mathbf{u}) = 0 \quad (7)$$

where  $\varphi$  is the volume fraction of the VOF method:

$$\begin{cases} \varphi = 0 & \text{Air} \\ 0 < \varphi < 1 & \text{Free surface} \\ \varphi = 1 & \text{Water} \end{cases} \quad (8)$$

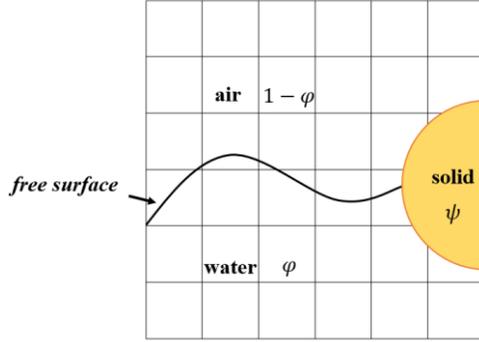


Fig. 1 Interface indicator function diagram

To determine the location of the solid, the Level Set method is used to identify and track the location of the solid, as shown in Fig. 1.  $\psi$  is the symbolic distance function of Level Set, whose value represents the distance from the solid:

$$\partial_t \psi + \nabla \cdot (\psi \mathbf{u}) = 0 \quad (9)$$

$$\begin{cases} \psi < 0 & \text{Liquid} \\ \psi = 0 & \text{Solid boundary} \\ \psi > 0 & \text{Solid} \end{cases} \quad (10)$$

### Immersed boundary formulation

For Eq. (3), the second-order Bell–Collela–Glaz scheme is used to solve the convection term:

$$\mathbf{u}^{**} = \mathbf{u}^n - \Delta t \nabla \cdot (\mathbf{u}^{n+\frac{1}{2}} \otimes \mathbf{u}^{n+\frac{1}{2}}) \quad (11)$$

After solving for  $\mathbf{u}^{**}$ , it is substituted into the diffusion term for further computation:

$$\rho^{n+\frac{1}{2}} \frac{\mathbf{u}^* - \mathbf{u}^{**}}{\Delta t} = \nabla \cdot [2\mu^{n+\frac{1}{2}} \mathbf{D}^*] + \mathbf{a}^n - \nabla p^n \quad (12)$$

here,  $\mathbf{u}^*$  represents the intermediate velocity to be solved, and an DF-IBM source term needs to be added to the right-hand side of the equation:

$$\mathbf{f}_B^n = \frac{\mathbf{u}_f^n - \mathbf{u}^*}{\Delta t} \quad (13)$$

The source term  $\mathbf{f}_B^n$ , accounting for boundary effects, is then incorporated into the discretized equation to solve for the updated intermediate velocity  $\mathbf{u}^*$ :

$$\rho^{n+\frac{1}{2}} \frac{\mathbf{u}^* - \mathbf{u}^{**}}{\Delta t} = \nabla \cdot [2\mu^{n+\frac{1}{2}} \mathbf{D}^*] + \mathbf{a}^n - \nabla p^n + \mathbf{f}_B^n \quad (14)$$

The key challenge lies in determining  $\mathbf{u}_f$ . As shown in Fig. 2, all grid points are classified into three categories: 1) Solid points: Grid points located inside the solid or on its surface. 2) Forcing points: Grid points within the fluid domain that have at least one neighboring grid point inside the solid. 3) Fluid points: Grid points located in the fluid domain

but not classified as forcing points.

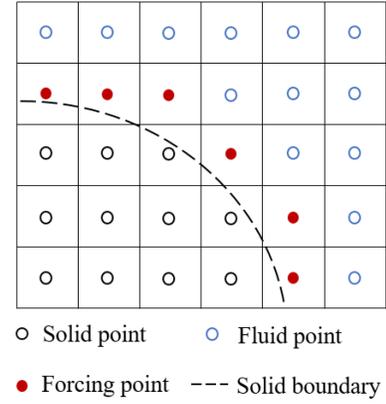


Fig. 2 Grid point classification

For solid points,  $\mathbf{u}_f = \mathbf{u}_s$ , where  $\mathbf{u}_s$  represents the boundary condition on the solid surface. For fluid points,  $\mathbf{u}_f = \mathbf{u}^*$ , where  $\mathbf{u}^*$  is the velocity obtained directly from the flow field. For forcing points, the velocity is calculated through interpolation. The interpolation method used in this study is similar to the approach proposed by Yang and Stern (2012) and Ghigo et al. (2021).

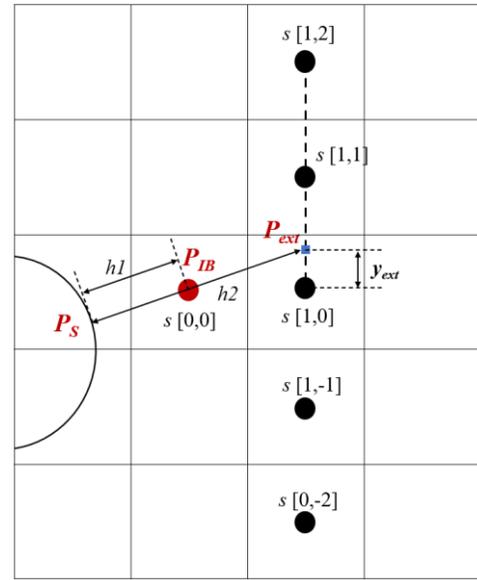


Fig. 3 Interpolation algorithm diagram

As shown in Fig. 3, for forcing point  $P_{IB}$ ,  $P_S$  represents the closest point to the surface, while  $P_{ext}$  is the intersection point where the line extending from  $P_S$  to  $P_{IB}$  meets the centerline of the grid cell adjacent to  $P_{IB}$  on the right.

$$\mathbf{u}_{IB} = \frac{h_2 - h_1}{h_2} \mathbf{u}_S + \frac{h_1}{h_2} \mathbf{u}_{ext} \quad (15)$$

where,  $h_1 = \psi_{IB}$  and  $h_2 = \psi_{ext}$ ,  $\mathbf{u}_{ext}$  need to be interpolated.

The normal vector at  $P_{IB}$  is  $\mathbf{n} = \nabla \psi / |\nabla \psi|$ . If  $n_x = n_y$ , the vertical interpolation template is used as shown in Fig. 3. For the variable  $S$ , if

$y_{ext} > 0.5\Delta$ , then

$$\frac{s_{ext}}{\Delta} = (s[1,0](y_{ext} - 1) + s[1,2](y_{ext} + 1)) \frac{y_{ext}}{2} - s[1,1](y_{ext} - 1)(y_{ext} + 1) \quad (16)$$

where  $\Delta$  is the mesh size. If  $y_{ext} < 0.5\Delta$ , then  $s[1,0]$ ,  $s[1,-1]$ ,  $s[1,-2]$  three grid points interpolation is used. If  $y_{ext} = 0.5\Delta$ , then  $s[1,-1]$ ,  $s[1,0]$ ,  $s[1,1]$  three grid points interpolation is used.

The interpolation method presented here applies to one specific case; other cases follow a similar interpolation approach. Based on this,  $\mathbf{u}_f$  can be updated using Eq. (15), allowing the calculation of  $\mathbf{f}_B$ . The entire computational flow field is shown in Fig. 4.

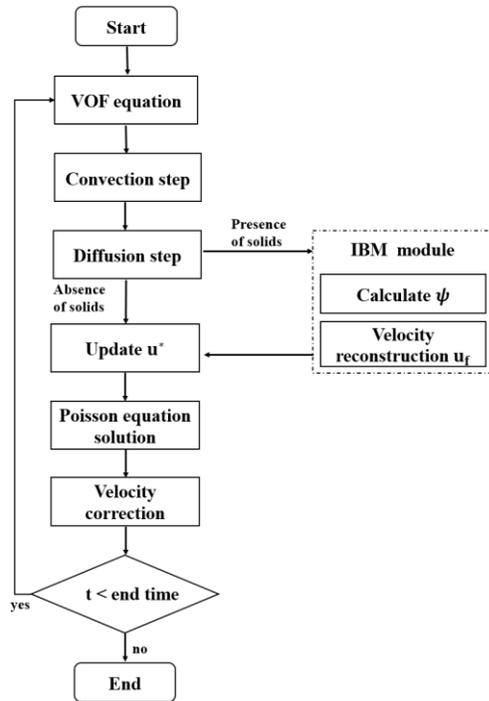


Fig. 4 Calculation flow chart

## RESULTS

### Flow around a circular cylinder

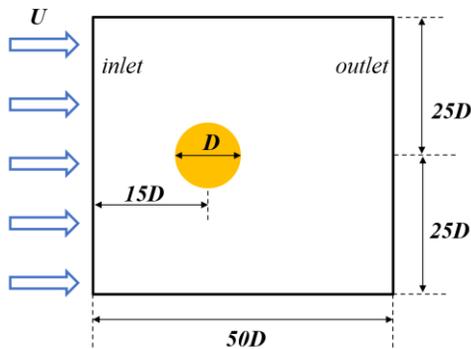


Fig. 5 Computing domain setting for flow around a circular cylinder

The flow around a stationary cylinder is selected to validate the

Immersed Boundary Method, as shown in Fig. 5. The computational domain has dimensions of  $50D \times 50D$ , with the cylinder center located  $15D$  from the inlet. The cylinder diameter is  $D = 1\text{ m}$ , and the inflow velocity is  $U = 1\text{ m/s}$ . Simulations are performed at  $Re = UD/\mu = 40, 100$ , with a minimum grid size of  $0.006D$ . Fig. 6 illustrates the grid configuration for this problem, where the red represents the forcing points that require correction.

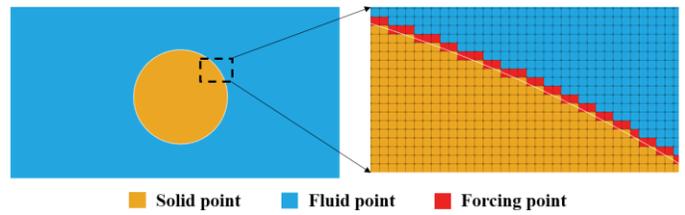


Fig. 6 Grid classification diagram

Fig. 7 and 8 depict the instantaneous vorticity fields for  $Re = 40$  and  $Re = 100$ , respectively. For  $Re = 40$ , no flow separation is observed in the cylinder's wake, and a pair of symmetric vortices forms behind the cylinder. However, for  $Re = 100$ , alternating vortex shedding is evident in the cylinder's wake, forming a Kármán vortex street.

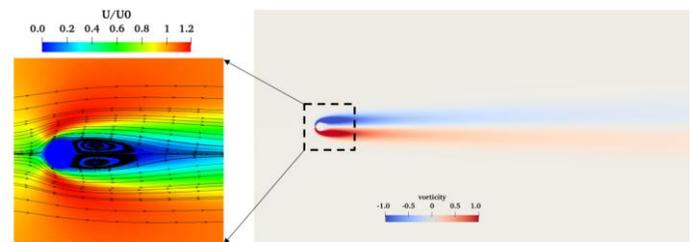


Fig. 7 The velocity and vorticity contour for  $Re=40$

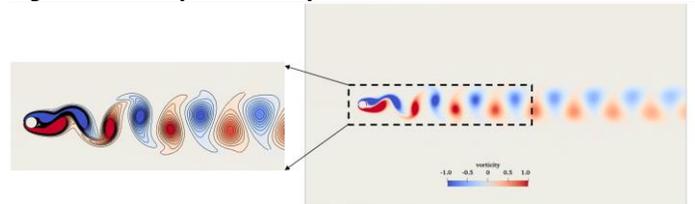


Fig. 8 The vorticity contour for  $Re=100$

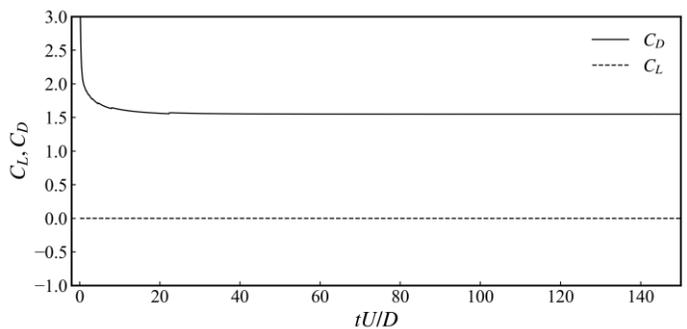


Fig. 9 The time curve of lift coefficient and drag coefficient for  $Re = 40$

Fig. 9 and 10 show the time histories of the lift and drag coefficients for  $Re = 40$  and  $Re = 100$ , respectively. Due to the Kármán vortex street phenomenon, the lift coefficient for  $Re = 100$  exhibits significant oscillations. Furthermore, Table 1 compares the results of this study with those from other authors. For  $Re = 40$ , the comparison includes the drag coefficient  $C_D$  and the symmetric vortex length behind the cylinder  $L_w$ .

For  $Re = 100$ , the comparison involves the mean drag coefficient  $C_{D,mean}$ , the maximum lift coefficient  $C_{L,max}$ , and the Strouhal number  $S_t$ , which characterizes the vortex shedding frequency. The results of the present study are found to be in good agreement with those reported in the papers, demonstrating the accuracy of the Immersed Boundary Method used in this work.

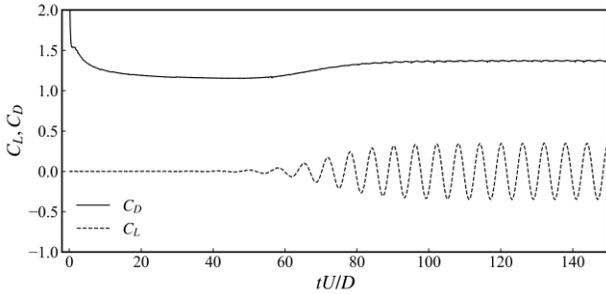


Fig. 10 The time curve of lift coefficient and drag coefficient for  $Re = 100$

Table 1. comparison of results

|                           | $Re = 40$ |       | $Re = 100$   |             |       |
|---------------------------|-----------|-------|--------------|-------------|-------|
|                           | $C_D$     | $L_w$ | $C_{D,mean}$ | $C_{L,max}$ | $S_t$ |
| Present work              | 1.55      | 2.27  | 1.37         | 0.35        | 0.166 |
| Li et al., (2016)         | 1.55      | 2.34  | 1.36         | 0.33        | 0.165 |
| Linnick and Fasel (2005)  | 1.54      | 2.28  | 1.34         | 0.34        | 0.166 |
| Tseng and Ferziger (2003) | 1.53      | 2.21  | 1.42         | -           | 0.164 |

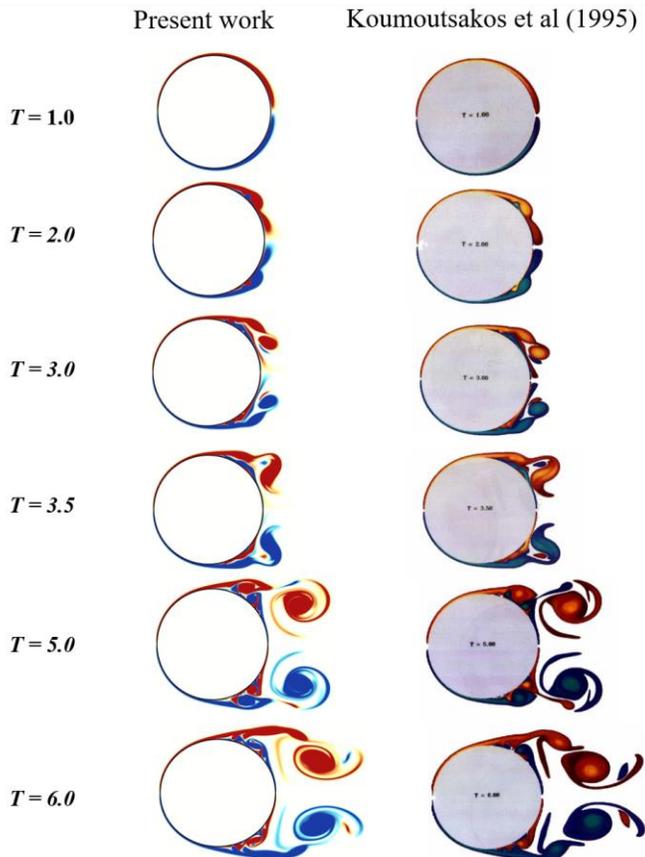


Fig. 11 The vorticity contour for  $Re = 9500$

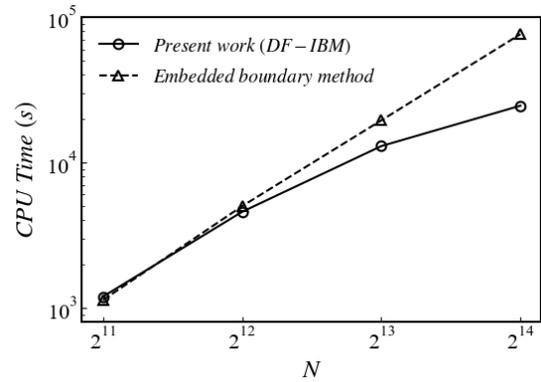


Fig. 12 CPU computation time at different grid resolutions for  $Re = 40$

High Reynolds numbers pose a challenge to the accuracy of the Immersed Boundary Method. As shown in Fig. 11,  $Re=9500$  is selected as a test case to investigate the vortex evolution around the cylinder during the initial stages of flow. The minimum grid size is set to  $0.0005D$ , and the simulation is conducted using the dimensionless time  $T=Ut/R$ . The results in Fig. 11 indicate that the current Immersed Boundary Method effectively accounts for the influence of the solid surface. The generation and rotational evolution of positive and negative vortices at the cylinder's trailing edge are clearly observed.

Fig. 12 shows a comparison of CPU computation time between the present method and the existing Embedded Boundary Method (EBM) in Basilisk for the  $Re=40$  case under 8-core OpenMP parallelization. The grid resolution is set to  $\Delta=50D/N$ . As shown in the figure, the present method demonstrates higher computational efficiency as the number of grid cells increases. For example, for a grid resolution of  $\Delta=50D/2^{13}=0.006D$ , the CPU computation time for the DF-IBM method in this study is 12,950.1s, whereas the EBM method requires 19,464.4s. This represents a 33.47% improvement in computational efficiency.

### Transversely oscillating cylinder in a free-stream

The case of a moving object is considered, as shown in Fig. 13, where a cylinder oscillates laterally in a free-stream flow. The cylinder's motion is described by the equation  $y(t)=A\sin(2\pi f_e t)$ , with  $A=0.2D$ . The motion frequency of the cylinder is  $f_e/f_0=0.8$ , where vortex shedding frequency of a stationary cylinder  $f_0=0.195$ . The Reynolds number  $Re=185$ , and the minimum grid size is  $0.146D$ .

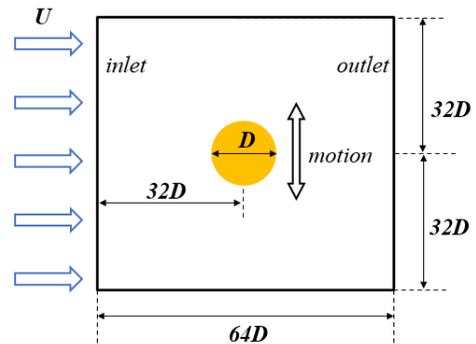


Fig. 13 Computing domain setting for transversely oscillating cylinder

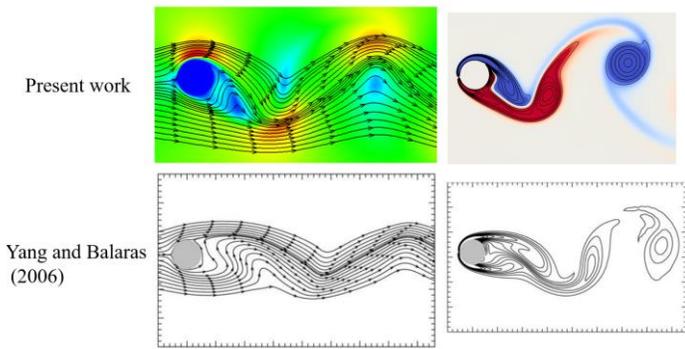


Fig. 14 The velocity and vorticity contour when cylinder is located at its extreme upper position;

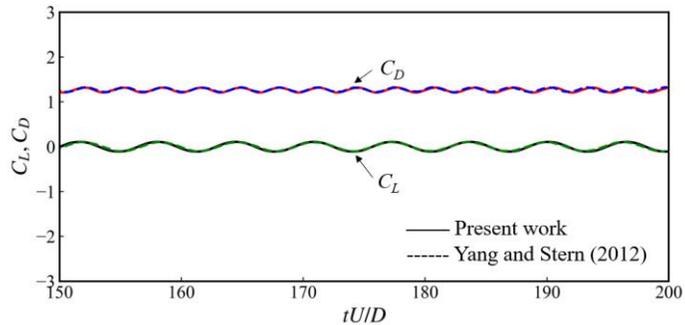


Fig. 15 The time curve of lift coefficient and drag coefficient for transversely oscillating cylinder

Fig. 14 illustrates the instantaneous velocity streamlines and vorticity fields, while Fig. 15 compares the computed forces on the oscillating cylinder with results from other studies. Overall, the present study effectively captures the flow field variations and the forces acting on the cylinder during lateral oscillation.

### Water impact and entry problem

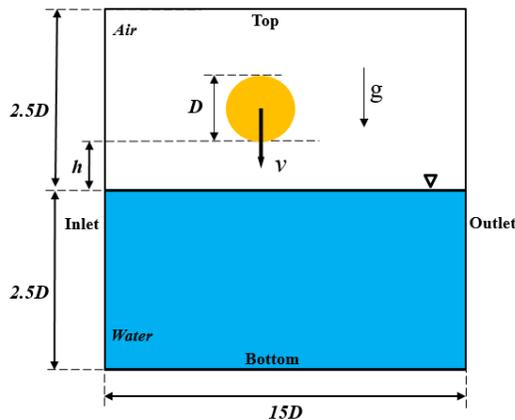


Fig. 16 Computing domain setting for water impact and entry problem

To evaluate the application of the current method in two-phase flow, numerical simulations are performed for two cases: a cylinder entering the water from air and a cylinder exiting the water. As shown in Fig. 16, the cylinder has a diameter of  $D = 2\text{ m}$ , with a minimum grid size of  $0.146D$ , and gravity acceleration is set to  $g = 1\text{ m/s}^2$ . For the water-entry case, the cylinder initially starts at a height  $h = 1.25\text{ m}$ , above the free surface with a velocity of  $U = 1\text{ m/s}$ . For the water-exit case, the

cylinder starts at  $h = -1.25\text{ m}$  below the free surface with a velocity of  $U = -0.39\text{ m/s}$ .

As shown in Fig. 17, upon entering the water, two jets are generated on either side of the cylinder, propagating outward. Subsequently, two vortex centers form in the air near the free surface and detach. As the cylinder moves deeper underwater, the free surface converges, colliding to form a large upward jet.

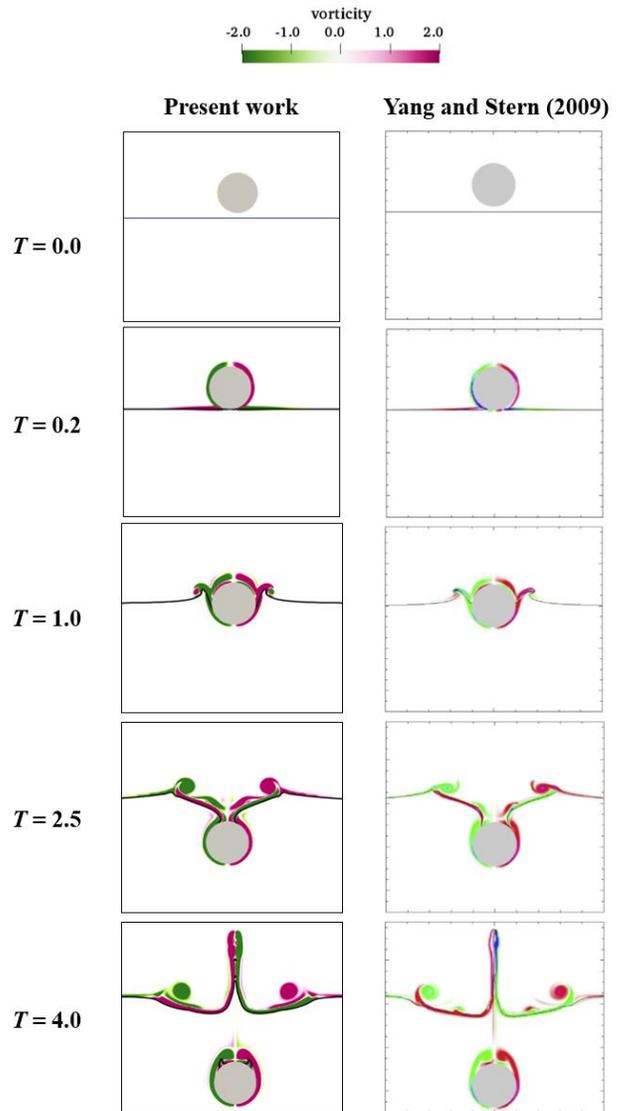


Fig. 17 Free surface evolution and vorticity contour for water entry problem.

As shown in Fig. 18, compared to the water-entry case, the water-exit case produces two vortex dipoles in the wake of the cylinder while it remains submerged. After the cylinder exits the water, some liquid is still carried upward at its trailing edge. Compared to the results of Yang and Stern (2009), the current study uses a finer mesh, leading to the generation of more vortex dipoles. Additionally, after the cylinder fully exits the water, the droplets falling back significantly affect the vorticity field in the wake.

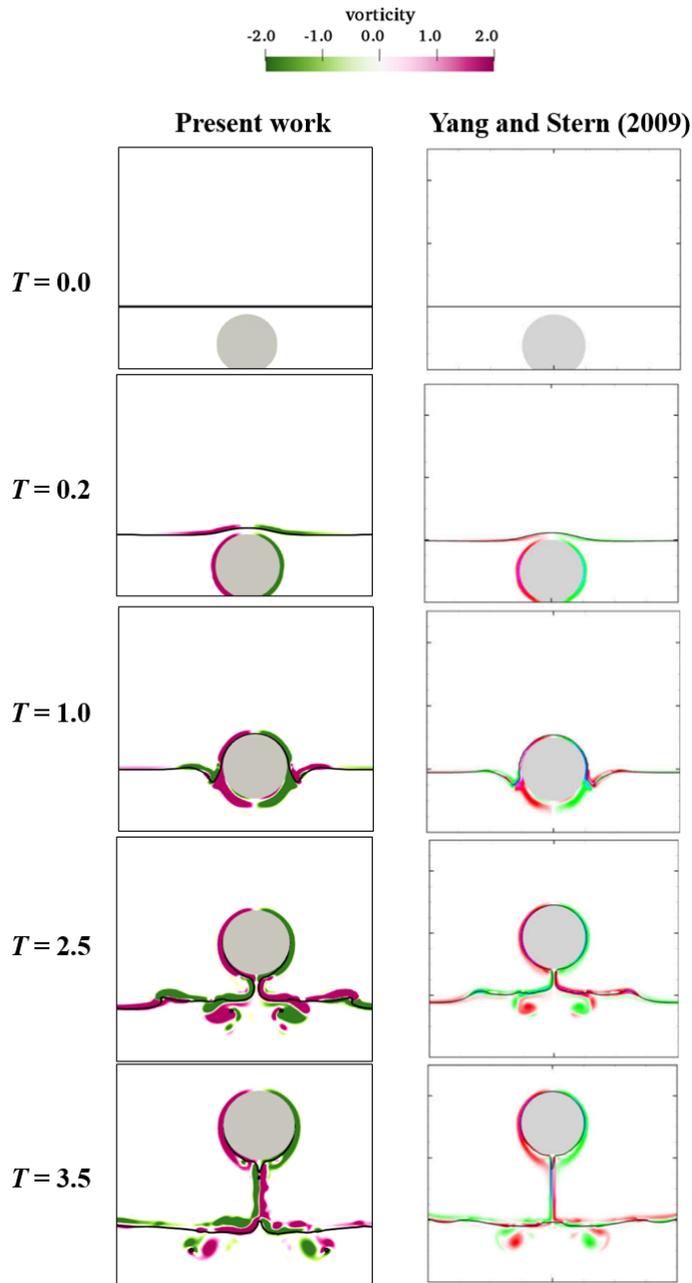


Fig. 18 Free surface evolution and vorticity contour for water exit problem

### Wave breaking phenomena around a wedge-shaped bow

In this section, the ship's bow is simplified into a wedge-shaped structure, and simulations are conducted to analyze the wave-breaking phenomena around the wedge (Wang et al., 2010). As shown in Fig. 19, the wedge has a slanted edge length of  $0.75\text{ m}$ , an incoming flow angle of  $30^\circ$ , and a draft of  $d = 0.0745\text{ m}$ . The computational domain dimensions are  $L_x = L_y = L_z = 2\text{ m}$ , with an incoming flow velocity of  $U = 2.5\text{ m/s}$  and a minimum grid size of  $2\text{ mm}$ . The Reynolds number is approximately  $Re = \rho U d / \mu = 1.64 \times 10^5$ , and the Froude number is  $Fr = U / \sqrt{g d} = 2.93$ .

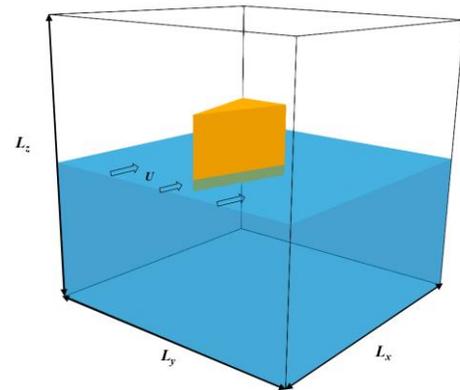


Fig. 19 Computing domain setting for bow wave breaking

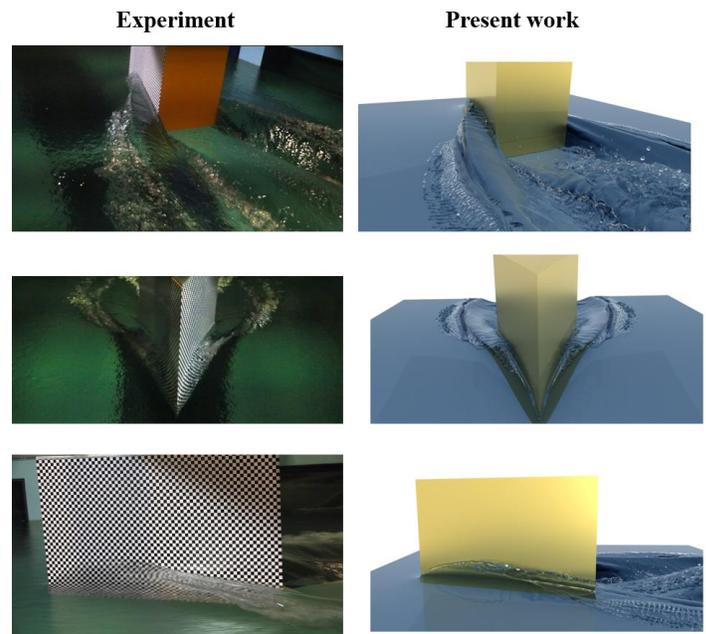


Fig. 20 Comparison between experiment and simulation

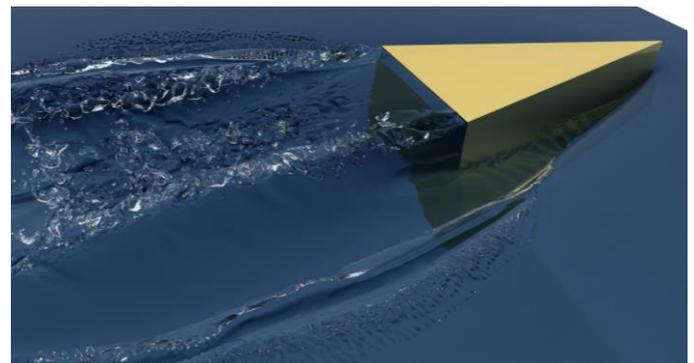


Fig. 21 Underwater bubble entrainment phenomenon

The comparison between the computational results and experimental data is shown in Fig. 20. The current study successfully captures the interaction between the wedge-shaped structure and the free surface. The incoming flow climbs along the sides of the wedge and overturns,

forming breaking waves. The overall wave-breaking pattern closely matches the experimental observations. Moreover, the secondary overturning of the waves and the resulting droplet structures are accurately reproduced. Fig. 21 presents an underwater perspective, where the air tubes formed by wave overturning and the bubble sweep-down phenomena caused by the wedge's sharp trailing edge are also well captured.

## CONCLUSIONS

In this paper, we propose a novel direct-forcing immersed boundary method that is well-suited for implementation in the open-source adaptive mesh solver Basilisk. The core of the algorithm lies in interpolating the velocity at the forcing points in the fluid domain and incorporating the force source terms into the discrete equations. The accuracy of the proposed method in single-phase flow is validated through simulations of flow around a cylinder at  $Re = 40, 100$  and  $9500$ . Additionally, the method's reliability in cases involving simple structure motion is verified using the transversely oscillating cylinder in a free-stream and water impact and entry problems. Finally, wave-breaking phenomena around a wedge-shaped structure are simulated, resembling the bow wave-breaking phenomenon in the ship hydrodynamics field, demonstrating the potential of the method for studies involving simple structures in marine environments. Our future work is to extend present method to the simulation of elastic structures (e.g., ship hull deformation) and multi-body coupled motions (e.g., propeller-rudder interactions).

## ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (52131102), to which the authors are most grateful.

## REFERENCES

- Balaras, E. (2004). "Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations," *Computers and Fluids*, 33(3), 375-404.
- Cheng, Z., and Wachs, A. (2022). "An immersed boundary/multi-relaxation time lattice Boltzmann method on adaptive octree grids for the particle-resolved simulation of particle-laden flows," *Journal of Computational Physics*, 471, 111669.
- Fadlun, E. A., Verzicco, R., Orlandi, P., and Mohd-Yusof, J. (2000). "Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations," *Journal of computational physics*, 161(1), 35-60.
- Fuster, D., Popinet, S. (2018). "An all-Mach method for the simulation of bubble dynamics problems in the presence of surface tension," *Journal of Computational Physics*, 374: 752-768.
- Ghigo, A. R., Popinet, S., and Wachs, A. (2021). "A conservative finite volume cut-cell method on an adaptive Cartesian tree grid for moving rigid bodies in incompressible flows," <https://hal.archives-ouvertes.fr/hal-03948786>.
- Gilmanov, A., and Sotiropoulos, F. (2005). "A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of computational physics*, 207(2), 457-492.
- Goldstein, D., Handler, R., and Sirovich, L. (1993). "Modeling a no-slip flow boundary with an external force field," *Journal of computational physics*, 105(2), 354-366.
- Guo, C., Ji, M., Han, Y., Liu, T., Wu, Y., and Kuai, Y. (2023). "Numerical simulation of the horizontal rotating cylinder and the air entrainment near the free surface," *Physics of Fluids*, 35(9).
- Hu, Y., Liu, C., Hu, C., and Wan, D. (2021). "Numerical investigation of flow structure and air entrainment of breaking bow wave generated by a rectangular plate," *Physics of Fluids*, 33(12).
- Huang, W. X., and Sung, H. J. (2010). "Three-dimensional simulation of a flapping flag in a uniform flow," *Journal of Fluid Mechanics*, 653, 301-336.
- Huet, D. P., and Wachs, A. (2023). "A Cartesian-octree adaptive front-tracking solver for immersed biological capsules in large complex domains," *Journal of Computational Physics*, 492, 112424.
- Kan, K., Li, H., and Yang, Z. (2023). "Large eddy simulation of turbulent wake flow around a marine propeller under the influence of incident waves," *Physics of Fluids*, 35(5).
- Kan, K., Yang, Z., Lyu, P., Zheng, Y., and Shen, L. (2021). "Numerical study of turbulent flow past a rotating axial-flow pump based on a level-set immersed boundary method," *Renewable Energy*, 168, 960-971.
- Kim, Y., and Peskin, C. S. (2007). "Penalty immersed boundary method for an elastic boundary with mass," *Physics of Fluids*, 19(5).
- Koumoutsakos, P., and Leonard, A. (1995). "High-resolution simulations of the flow around an impulsively started cylinder using vortex methods," *Journal of Fluid Mechanics*, 296, 1-38.
- Li, R. Y., Xie, C. M., Huang, W. X., and Xu, C. X. (2016). "An efficient immersed boundary projection method for flow over complex/moving boundaries," *Computers and Fluids*, 140, 122-135.
- Li, Z., Liu, C., Wan, D., and Hu, C. (2021). "High-fidelity simulation of a hydraulic jump around a surface-piercing hydrofoil," *Physics of Fluids*, 33(12).
- Limare, A., Popinet, S., Josserand, C., Xue, Z., and Ghigo, A. (2023). "A hybrid level-set/embedded boundary method applied to solidification-melt problems," *Journal of Computational Physics*, 474, 111829.
- Linnick, M. N., and Fasel, H. F. (2005). "A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains," *Journal of Computational Physics*, 204(1), 157-192.
- Liu, C., and Hu, C. (2014). "An efficient immersed boundary treatment for complex moving object," *Journal of Computational Physics*, 274, 654-680.
- Liu, C., and Hu, C. (2017). "An immersed boundary solver for inviscid compressible flows," *International Journal for Numerical Methods in Fluids*, 85(11), 619-640.
- Mittal, R., and Seo, J. H. (2023). "Origin and evolution of immersed boundary methods in computational fluid dynamics," *Physical review fluids*, 8(10), 100501.
- Peskin, C. S. (1972). "Flow patterns around heart valves: a numerical method," *Journal of computational physics*, 10(2), 252-271.
- Popinet, S. (2003). "Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries," *Journal of computational physics*, 190(2), 572-600.
- Popinet, S. (2009). "An accurate adaptive solver for surface-tension-driven interfacial flows," *Journal of Computational Physics*, 228(16), 5838-5866.
- Shao, Y., Wang, W., Wan, D., and Wang, J. (2024). "Numerical investigations of breaking waves and air entrainment induced by a shallowly submerged hydrofoil," *Ocean Engineering*, 312, 119026.
- Sharaborin, E. L., Rogozin, O. A., and Kasimov, A. R. (2021). "The coupled volume of fluid and Brinkman penalization methods for simulation of incompressible multiphase flows," *Fluids*, 6(9), 334.
- Tavares, M., Josserand, C., Limare, A., Lopez-Herrera, J. M., and Popinet, S. (2024). "A coupled VOF/embedded boundary method to model two-phase flows on arbitrary solid surfaces," *Computers and Fluids*, 278, 106317.
- Tseng, Y. H., and Ferziger, J. H. (2003). "A ghost-cell immersed boundary method for flow in complex geometry," *Journal of computational physics*, 192(2), 593-623.

- Vanella, M., Rabenold, P., and Balaras, E. (2010). "A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid–structure interaction problems," *Journal of Computational Physics*, 229(18), 6427-6449.
- Wang, Z., Yang, J., and Stern, F. (2010). "Numerical simulations of wave breakings around a wedge-shaped bow," *In 28th Symposium on Naval Hydrodynamics*, Pasadena, California.
- Yang, J., and Balaras, E. (2006). "An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries," *Journal of computational Physics*, 215(1), 12-40.
- Yang, J., and Stern, F. (2009). "Sharp interface immersed-boundary/level-set method for wave–body interactions," *Journal of Computational Physics*, 228(17), 6590-6616.
- Yang, J., and Stern, F. (2012). "A simple and efficient direct forcing immersed boundary framework for fluid–structure interactions," *Journal of Computational Physics*, 231(15), 5029-5061.
- Youngs, D. L. (1982). "Time-dependent multi-material flow with large fluid distortion," *Numerical methods for fluid dynamics*.