

# An Immersed Boundary Method OpenFOAM Solver for Structure - Two-phase Flow Interaction

*Vu Q. Do, Bac Viet Nguyen, Phu-Khanh Nguyen, Van-Sang Pham\**

*Hanoi University of Science and Technology – No. 1, Dai Co Viet Str., Hai Ba Trung, Ha Noi, Viet Nam*

*Received: June 04, 2019; Accepted: November 28, 2019*

## Abstract

*In this study, a numerical solver has been developed for simulating the interaction of solid bodies with two-phase flow. The simplified direct forcing method is adopted to describe the presence of structures in flow, combining with the Pressure-Implicit Split-Operator (PISO) algorithm to address the velocity-pressure coupling in Navier-Stokes equations. The fluid-fluid interface reconstruction is performed by an advanced free-surface capturing model based on the volume-of-fluid (VOF) model, thus higher interface resolution can be achieved without the need of other special treatment. The computational mesh is distributed more efficiently and economically at the interested regions, e.g. fluid-fluid interface/solid-fluid interfaces, through an adaptive mesh refinement process, which significantly reduces computation time, while maintains excellent accuracy, demonstrated by simulation results involving surface breakup, water impact, and bodies – free surface interaction.*

Keywords: Immersed boundary method, Volume of Fluid, Adaptive mesh refinement, Two-phase flow.

## 1. Introduction

The interaction of incompressible two-phase flows with solid structures is usually encountered in scientific research and industrial applications such as ship hydrodynamics and maneuvering, ocean and coastal engineering. Numerical modeling of such problems brings significant advantages, yet very challenging due to numerous complicated physical phenomena involved, i.e. free surface breakup, wave impacts, and wave-bodies interaction.

There are generally two methods to simulate the interaction of fluid flows and solid bodies: meshless methods, and grid method. The meshless method employs a set of small moving particles to represent the continuous fluid motion. Although many improvements of the mesh-free method have been made, its applications are still limited due to the complexities of algorithm and strict requirement of computer resources.

The grid method can be classified into moving grid methods and fixed grid methods. The moving grid method is widely used to represent solid bodies or interfaces between two immiscible fluids by employing a grid that is conformal to the solid surfaces or the interfaces. To maintain the conformity between the cell boundary and tracked surfaces when these objects are in motion, a mesh deformation or a remeshing process is required, depending on the

movement is small or large. Various improvements of this method have been made to reduce computation cost such as using multi-block mesh [4] instead of single-block mesh system, which was adopted by Chen and Liu [5] to investigate the flow induced by a ship in harbor, and by Carrica et al. [6] to study the wave diffraction problem for a surface ship. However, the process of grid deformation, grid re-generation and interpolation between overlapped grids are still time-consuming and prone to errors.

The fixed grid methods, on the other hand, can handle unrestricted motions of solid bodies and phase interfaces on a single fixed Cartesian grid that usually cover the entire computational domain. Two broadly known methods, the cut-cell method, and the immersed boundary (IB) method are both very flexible, cost-saving and can deal with objects of arbitrary shape. In the cut-cell method, however, the processes of cutting grid cells intersected by the interfaces can generate infinitesimal mesh elements that cause serious computational errors. Moreover, the treatments required for a large number of interface cells when the interfaces/boundaries evolve make this method difficult to be applied to three-dimensional (3D) problems.

In this study, we modify the direct forcing IB method [1] for modeling the interaction of solid structures and two-phase flows. The free surface

---

\*Corresponding author: Tel.: (+84) 966.633.683  
Email: sang.phamvan@hust.edu.vn

reconstruction is handled by employing an advanced model based on a two-fluid formulation of the Volume of Fluid (VOF) method [3]. The endured issue when dealing with two-phase flow is the evolution of large liquid-liquid interfaces may lead to an enormous number of mesh elements required for capturing sharp interfaces, which significantly increase the simulation time and possibly outweigh the above advantages of the IB method. To overcome this issue, an adaptive mesh refinement (AMR) engine is integrated into our solver to distribute mesh elements near solid-fluid and fluid-fluid interfaces more economically, enabling sharp interface capturing, save computation time while maintain excellent accuracy.

## 2. Numerical method

### 2.1. Governing equations

The volume of fluid (VOF) method is utilized in resolving two-phase flow. In VOF, an indicator function  $\gamma$  that takes values between the ranges  $0 \leq \gamma \leq 1$  is introduced for liquid phases indicating. A unity value represents a cell full of fluid; a zero value represents a cell full of air, whereas cells with a value between zero and one contains a free surface. Considering two immiscible fluids as one single effective fluid throughout the domain, its physical properties can be defined as a weighted average:

$$\mathbf{U} = \gamma \mathbf{U}_l + (1 - \gamma) \mathbf{U}_g \quad (1)$$

$$\rho = \gamma \rho_l + (1 - \gamma) \rho_g \quad (2)$$

$$\mu = \gamma \mu_l + (1 - \gamma) \mu_g \quad (3)$$

where  $\mathbf{U}, \rho, \mu$ , are the velocity vector, density and dynamic viscosity, respectively, with the subscripts  $l$  and  $g$  denote the liquid and gaseous phase. The conservation of mass and momentum of the effective fluid are then written as:

$$\nabla \cdot \mathbf{U} = 0 \quad (4)$$

$$\frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot \mathbf{T} + \rho \mathbf{f}_b \quad (5)$$

where  $p$  is the pressure field in by the two fluids domains. Assuming that both fluids are Newtonian and incompressible, the divergence of the viscous stress tensor term is decomposed into:

$$\begin{aligned} \nabla \cdot \mathbf{T} &= \nabla \cdot \{ \mu [ \nabla \mathbf{U} + (\nabla \mathbf{U})^T ] \} \\ &= \nabla \cdot (\mu \nabla \mathbf{U}) + (\nabla \mathbf{U}) \cdot \nabla \mu \end{aligned} \quad (6)$$

The external body force is the sum of gravity, surface tension and feedback force exerting on the two-phase flow by immersed bodies:

$$\mathbf{f}_b = \mathbf{g} \cdot \mathbf{x} + \mathbf{f}_\sigma + \mathbf{f}_b \quad (7)$$

The surface tension is evaluated using the continuum surface force model [14]:

$$\mathbf{f}_\sigma = \sigma \kappa \nabla \gamma \quad (8)$$

with 
$$\kappa = -\nabla \cdot \left( \frac{\nabla \gamma}{|\nabla \gamma|} \right) \quad (9)$$

and the evaluation of the forcing term  $\mathbf{f}_b$  is presented in the section 2.2. The transport equation of the indicator function is solved together with the mass and momentum conservative equations:

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\mathbf{U} \gamma) + \nabla \cdot [\mathbf{U}_r \gamma (1 - \gamma)] = 0 \quad (10)$$

The last convective term in the equation (10) is referred to as the compressive term [13], with the compressive velocity defined as:  $\mathbf{U}_r = \mathbf{U}_l - \mathbf{U}_g$ . This term is active only in the free surface region (where  $\gamma$  is neither one nor zero) and intends to “compress” the interface to achieve a higher interface resolution, thus avoiding the need to employ a special scheme for sharp interfaces.

### 2.2. Immersed boundary method

The presence of structure in the two-phase flow is represented by the forcing term  $\mathbf{f}_b$ . In this study, the simplified direct forcing proposed in Zhang *et al.* [2] is adopted to deal with two-phase flows.

To enforce no-slip boundary condition on the solid surfaces, a volume of solid (VOS) function is used. The function takes values of either zero or one for cells within the fluid or the solid region, respectively. The forcing term can be specified by rearranging the Eq. (5):

$$\mathbf{f}_b = \eta \left( \frac{\mathbf{U}_b - \mathbf{U}^n}{\Delta t} - \mathbf{rhs} \right) \quad (11)$$

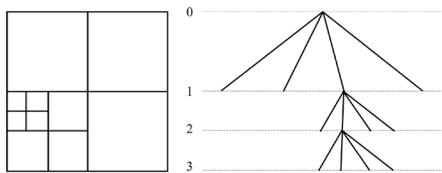
This calculation of the forcing term is simple and does not involve the back and forth interpolations between Cartesian grid points and set of points that approximate the solid surfaces [1][7][8], thus reduce computational errors during the interpolation process.

## 3. Adaptive meshing refinement

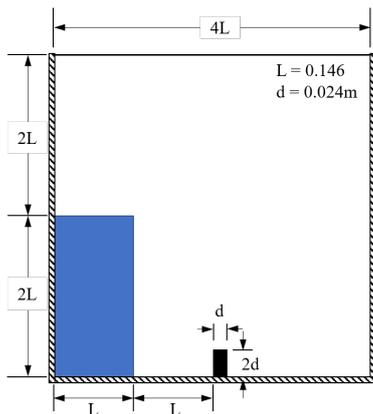
In this study, the tree-based AMR technique described in [9] has been employed to locally refine mesh at the regions of interest, i.e. liquid-air interface and surfaces of the immersed objects. The technique splits an original hexahedral mesh element (*parent cell*) into four (eight in 3D) smaller hexahedral elements (*children cells*). The data structure is organized hierarchically as a quadtree (octree in 3D),

with root cell being the base of the tree and leaf cells are cells without any child. Each cell has its corresponding *level*, which starts from zero for all elements at the beginning of the simulation. The level of the children is then determined by adding one to their parent's level. A schematic illustrating the spatial discretization and corresponding tree representation is shown in Fig. 1.

The local mesh refinement is performed at elements whose values of indicator function  $\gamma$  lie between one and zero, which contain the free surface, and at elements whose edges intersect with the immersed objects surfaces. The refinement can also be customized to produce smooth transition between regions with different cell levels.



**Fig. 1.** Quadtree discretization and its corresponding tree representation. The initial mesh resolution is referred to level 0. Every time a cell is split into four new cells, the levels of the new cells increase by one.



**Fig. 2.** Simulation model of the dam break problem.

**4. Results and discussion.**

**4.1. Dam break with an obstacle in the flow**

The simulation model is depicted in Fig. 2 with all parameters taken as those introduced by Ubbink [10]. No-slip boundary condition is applied to all boundaries except the top surface which is assumed to be an open boundary. The initial mesh resolution is  $25 \times 25$  and the free surface and the obstacle surface is refined up to level 4 during simulation.

Simulation results at different time instants are shown in Fig. 3a. From the beginning until  $t=0.3s$  the collapsing water column obstructed by the obstacle

bounces up from the upper left corner of the obstacle, forming a tongue-shaped flow and heading to the opposite wall. After the tongue reaches the right wall at  $t=0.4s$ , water starts to fall under the effect of gravity and compresses the trapped air beneath it. The air bubble eventually bursts through the water sheet while the water continues its downward motion due to gravity, creating a mixing region of water and air. The phenomena observed in the current study are in excellent agreement with that produced by Ubbink [10] in Fig. 3b.

To demonstrate the robustness of the present solver, we also conducted simulations on two different meshes, i.e., one is the conventional uniform cartesian mesh without AMR and another with AMR technique, as depicted in Fig. 3c. It is clearly shown from Table 1 that to achieve the same level of sharpness of the free surfaces in such a 2D problem, the current solver with application of the AMR technique can significantly reduce the size of the required mesh (i.e. more than 5 times), hence greatly save the computation time (about 60%).

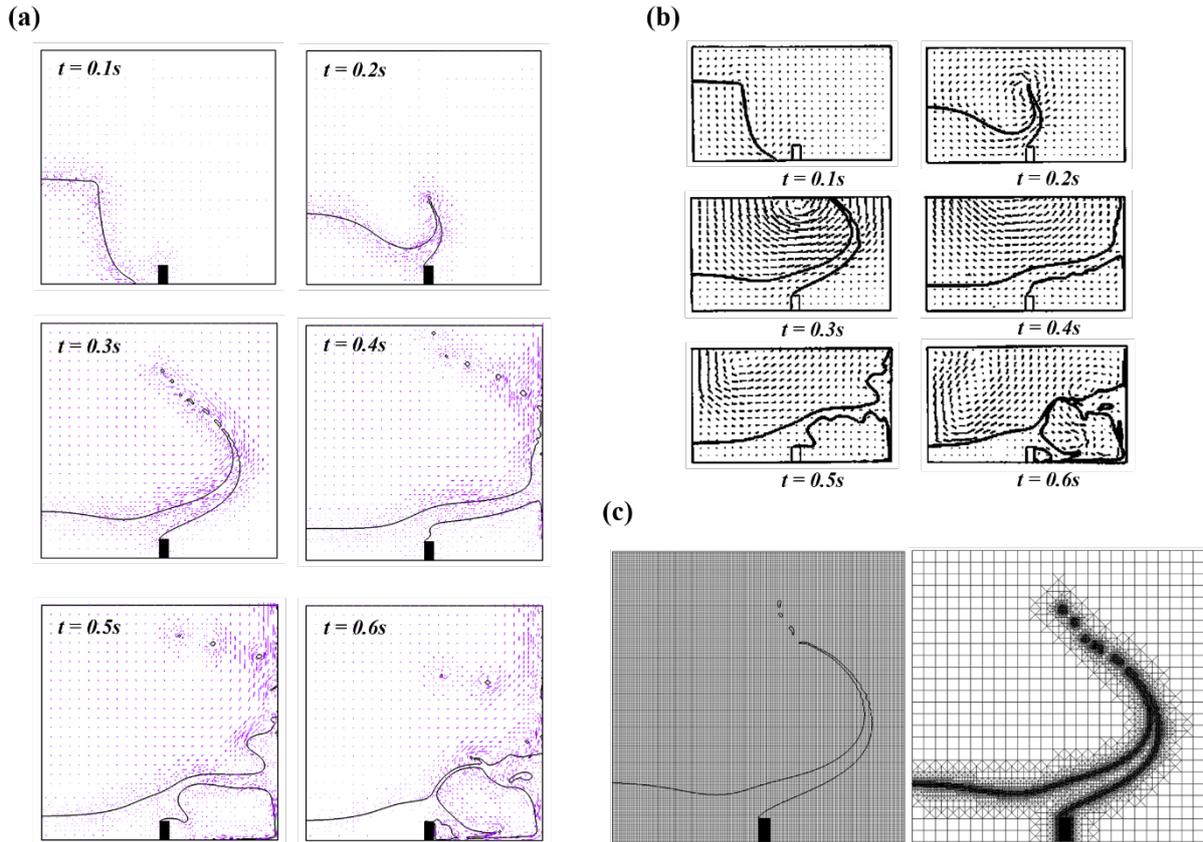
**Table 1.** Computation time for different mesh refinements of dam break problem.

	Without AMR	With AMR
No. of elements	160000	~30000
Computation time (s)	6381	2517

**4.2. Water entry and exit of a circular cylinder**

The problem of water entry/exit is closely related to practical applications in marine operation during which objects are either lowered into water or lifted out of the water, which involves complicated physical processes such as the breakup of free surface and interaction between solid body, free-surface, and vortices. The simulation setup is the same as those in [12]. The domain size is  $[40 \times 24]m$ , filled by 20m of water. The dynamic viscosity and density of water are  $1 \times 10^{-3} \text{ kg/ms}$  and  $1000 \text{ kg/m}^3$ , respectively, while those of air are  $1.8 \times 10^{-5} \text{ kg/ms}$  and  $1 \text{ kg/m}^3$ . The gravitational acceleration is  $g = 1.0 \text{ m/s}^2$ .

A uniform mesh with  $\Delta x = \Delta y = 0.8 \text{ m}$  (level 0) is used for the entire region, while cells in the vicinity of the free surface and the cylinder surface are refined to level 4 ( $\Delta x = \Delta y = 0.05 \text{ m}$ ). The cylinder has radius  $R = 1 \text{ m}$ . Its center is initially located at  $d = 1.25 \text{ m}$  beneath the free surface and is given a constant velocity  $V = 0.39 \text{ m/s}$  and  $V = -0.39 \text{ m/s}$  for the exiting and sinking problem, respectively. For easier to compare our results with previous studies, the dimensionless time  $T = Vt/d$  is considered instead of the the real-time  $t$ .



**Fig. 3.** (a) Temporal evolution of the free surface and velocity field; (b) Simulation conducted by Ubbink [10]; (c) Uniform mesh (left, 160.000 cells) in comparison with AMR mesh (right, ~30.000 cells) at  $t=0.3s$ .

**Table 2.** Computation time for different mesh refinements of the water exit problem.

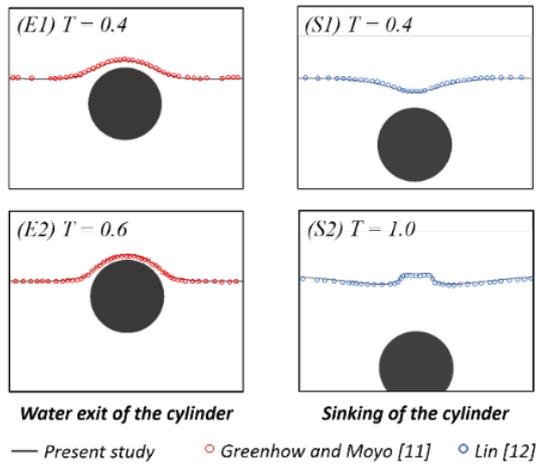
	Without AMR	With AMR		
		M1	M2	M3
Min cell size (m)	0.05	0.1	0.05	0.025
No. of elements	49280	8142	15747	36552
Computation time (s)	8507	1137	3456	18948

The simulation of the water exit problem is conducted on four different meshes to compare the difference in computation time. As can be seen from Table 2, the AMR technique in case M2 helps significantly to reduce 68% the required mesh size, thus save about 59.4% simulation time compares with case without AMR. In Fig. 4, the air-water interface profiles at different dimensionless time instants in the water exit and sinking problem are in excellent agreement with the simulation results using the boundary element method of Greenhow and Moyo [11] and the results based on cut-cell method of Lin [12], respectively. In addition, the evolution of the free surface interacting with the rising cylinder at different dimensionless time step illustrated in Fig. 5, are well

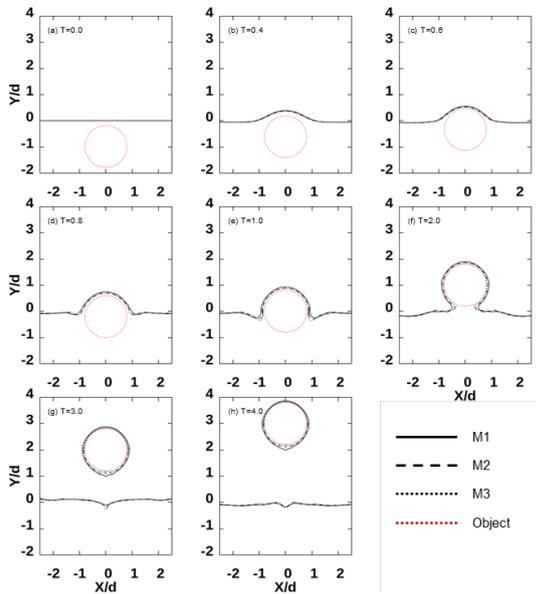
matched with the results reported in [12], which prove the accuracy of our solver.

### 5. Conclusion

In this study, a numerical method coupling IB method and VOF method has been developed to simulate the problem of interaction between solid objects and two-phase flow. The method was implemented in OpenFOAM making a new solver integrated in the framework. The fluid-air interface was sharply reconstructed by the advanced VOF model without any other special treatment. The solver has been tested using well-documented problems, i.e. breaking of a dam, water entry and exit of a circular cylinder, obtained results demonstrated an excellent agreement with published data. The AMR process integrated into the solver was able to substantially reduce computation time (i.e. 59.4% in the water exit problem and 60% in the damBreak problem), which proves the robustness of the solver over conventional approach in resolving structure – two-phase flow interaction problems.



**Fig. 4.** The free surface profile of the water exit (E1), (E2) and sinking (S1), (S2) of a circular cylinder in comparison with previous studies.



**Fig. 5.** (a) – (h) Snapshots of the water exit problem at different dimensionless time instants. The results are in good agreement with [12].

**Acknowledgments**

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 107.03-2016.11

**References**

[1] Do Quoc Vu, Pham Van Sang; A new fluid-structure interaction solver in OpenFOAM; Journal of Science and Technology, 135 (2019) 023-027.

[2] C. Zang et al, A two-phase model coupling with volume of fluid and immersed boundary methods for free surface and moving structure problems, Ocean Engineering, 74 (2013) 107-124.

[3] OpenCFD Ltd., <http://www.opencfd.co.uk>

[4] A. Benek, P.G. Burning, J.L Steger, A 3D Chimera grid embedding technique, AIAA Paper, (1985) 85-1523.

[5] H.C. Chen, T. Liu, Turbulent flow induced by full-scale ship in harbor, J. Eng. Mech. 125 (1999) 827-835.

[6] P.M. Carrica, R.V. Wilson, F. Stern, An unsteady single-phase level set method for viscous free surface flows, Int. J. Numer. Methods Fluids 53 (2007) 229-256.

[7] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2005) 448-476.

[8] E. Constant, C. Li, J. Favier, M. Meldi, P. Meliga, E. Serre, Implementation of a Discrete Immersed Boundary Method in Openfoam, 2016 arXiv:1609.04364.

[9] Z. Li, P. Song, An adaptive mesh refinement strategy for immersed boundary /interface method, Commun. Comput. Phys. 12 (2012) 515-527.

[10] O. Ubbink. Numerical Prediction of Two Fluid Systems with Sharp Interfaces. PhD thesis, Imperial College, University of London, 1997.

[11] M. Greenhow, S. Moyo, Water entry and exit of horizontal circular cylinders, Philos. Trans. Math. Phys. Eng. Sci. 355 (1997) 551–563.

[12] P. Lin, A fixed-grid model for simulation of a moving body in free surface flows, Comput. Fluids 36 (2007) 549-561.

[13] E. Berberovic, N.P. Van Hinsberg, S. Jakirlic, I.V. Roisman, C. Tropea, Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution, Physical Review E, 79 (2009).

[14] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modelling surface tension. J. Comp. Phys., 100:335–354, 1992.

[15] Peskin, C.S., "Numerical Analysis of blood flow in the heart", J. Comput. Phys. 25:220- 252, 1977.

[16] Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S., and Zanetti, G., "Modelling merging and fragmentation in multiphase flows with SURFER", J. Comput. Phys. 113:134-147,1994.