

Integration of Immersed Boundary Method with Two-Stage Correction and Dynamic Adaptive Mesh Refinement for Sharp-Edges Thin body Object

Mohammed Suleman Aldlemy*¹

¹ Department of Mechanical Engineering, Collage of Mechanical Engineering Technology, Benghazi-Libya.

Naser S.Sanoussi²

² Department of Mechanical Engineering, Collage of Mechanical Engineering Technology, Benghazi-Libya.

Saleh S S Elfallah³

³ Department of Mechanical Engineering, Collage of Mechanical Engineering Technology, Benghazi-Libya.

Walid Omar A Salem⁴

⁴ Department of Mechanical Engineering, Collage of Mechanical Engineering Technology, Benghazi-Libya.

Mustafa Agripi Farhat Ahmida⁵

⁵ Department of Mechanical Engineering, the Higher Institute for Refrigerating & Air Conditioning – Sokna.

Salah Elbadri⁶

⁶ Department of Mechanical Engineering, Collage of Mechanical Engineering Technology, Benghazi-Libya.

Mohammad Rasidi Rasani⁷

⁷ Centre for Integrated Design for Advanced Mechanical Systems, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia.

A.K. Ariffin⁸

⁸ Department of Mechanical Engineering, the Higher Institute for Refrigerating & Air Conditioning – Sokna.

T.M.Y.S, Tuan Ya⁹

⁹ Department of Mechanical Engineering, Faculty of Engineering, Universiti Teknologi Petronas, Tronoh, Perak, Malaysia.

Osama Sabir¹⁰

¹⁰ Department of Mechanical Engineering, Faculty of Engineering, Universiti Teknologi Petronas, Tronoh, Perak, Malaysia.

Abstract –

For most fluid-structure interaction (FSI) problems, the solution normally lies in solving the characteristics of the flow at the system's fluid-structure (FS) interface. Arterial and venous circulatory blood flow system is an example of such a system. FSI problems are mostly solved using the immersed boundary method (IBM). IBM treats the interface between the cells near the sharp edges and nodes representing the moving/stationary rigid thin object in a 2-D domain. This study presents the integration of IBM (with 2-stage pressure-velocity corrections) with dynamic adaptive mesh refinement (DAMR) to address the issues associated with FSI simulations for thin solids. The successful incorporation of the solid and fluid motions at the boundary requires the computation of the body forces in proportion to the volume fraction of the solid in the fluid cells. The DAMR-IBM algorithm was used to discretize and solve the boundary layer flow governing equations at the FS-interface. To accurately

capture the flow physics in the boundary layer, the DAMR was used to control the mesh resolution at the FS-interface, while a uniform Cartesian grid with relatively wide cells was used in the freestream region far from the object to ensure significant decrease of the computational cost. The validation of the DAMR-IBM algorithm was done using a laminar flow benchmark case around a cylinder; the results showed a good agreement between the lift coefficients, drag coefficients, and Strouhal numbers produced by the model and those published in the literature. Simulations were conducted for three test cases in order to demonstrate the performance and accuracy of the DAMR-IBM algorithm. The DAMR-IBM algorithm was implemented on a staggered grid for the fluid flow simulation around thin object and for the computational cost determination. The results verify the accuracy and robustness of the DAMR-IBM algorithm in capturing the flow physics of the flow at the FS-interface of thin solid. Hence, the algorithm is suitable for uniform flows simulation around rigid thin solid, including sharp-edged ones. The DAMR-IBM algorithm can also be used to simulate flows past thin solid with high geometrical complexity, rendering it suitable to simulate the motion of amoeboid, fish fins, and insect wings.

Keywords : Immersed boundary method, Dynamic, AMR, Two-stage velocity-pressure correction, thin objects, and Laminar flow.

INTRODUCTION

FSI problems involve determining the flow characteristics at the FSI of a system [1]–[7]. Examples of FSI include the rustling of leaves due to the wind, flapping of insect wings, and swimming of microorganisms. FSI problems are typically solved using body-fitted mesh methods based on the arbitrary Lagrangian-Eulerian approach [8]–[17]. However, to date, many numerical models are incapable of accurately capturing the flow physics at the FS-interface of ultra thin solid with complex geometries. The flow parameters at the surface of thin solid are determined in FSI problems via estimation of the velocities and pressure of the flow at the boundary layer. When the thin object is displaced or distorted, re-meshing is performed, which increases the computational cost. Because of its simplicity and cheap processing cost, fixed Cartesian mesh techniques are commonly employed for the discretization of the governing equations of fluid dynamics for the estimation of the flow properties of arbitrarily ultra- thin solid. IBM, on the other hand, uses a rectangular mesh and adds a few extra stages to determine the cell fractions occupied by arbitrarily shaped objects [18]–[29]. IBM has been widely used to simulate complex flows in medical applications and biological sciences such as simulating blood flow in the heart.

IBM was originally developed by Peskin [30] where the moving and stationary boundaries were computed by introducing an external additional force field into the governing equations of fluid dynamics in order to fulfil the no-slip boundary conditions at the FS-interface. IBM can be used to solve flows within vicinity of deformable thin solid such as the response of a filament in flowing soap film [31]–[34]. The Dirac delta function with smooth estimation was used to apply forces to the fluid and interpolate the fluid velocity during the soap film-filament interactions.

Kajishima [35] devised a basic approach (named cut-cell IBM) for simulating particle-laden flows, in which the determination of the forces acting on the body was done by the velocity of the solid domain. Solid volume fractions for the border cells were used to determine the solid & fluid velocities. Other researchers, such as [36], [37] and [38] took a different technique by incorporating the solid and its velocity into the solid nodes in the last time step. On the other hand, these numerical models do not comply to the divergence-free criterion, which is essential for FS-interface flows [39]. The technique by Kajishima et al has the FSI connected by body forces. The fundamental disadvantage of fixed grid designs is the inaccuracy of flow resolution in the presence of moving structures [40].

The maximum grid resolution in fixed grid schemes must be defined in all the zones within the computational space prior to simulations. Much effort has been made to improve fixed grid schemes in recent years. Löffler et al. [41] developed a parallel scheme for adaptive mesh refinement based on the AMSS-NCKU framework. Brehm et al. [42] developed a stabilized IBM approach for compressible Navier-Stokes equations. Ji et al. [43] developed a graphics processing unit-accelerated adaptive mesh refinement method for use with IBM. It is evident that all of these studies are focused on the development of innovative methods to expedite computational processing and

enhance accuracy of the solutions, as well as overcome the drawbacks of existing numerical models.

In IBM, the interpolation function determines the FS-interaction and blurs the space between the fluid and structure to within the width of the fluid cells. However, it is difficult to accurately capture the flow physics in the boundary layer at the FS-interface without implementing body-aligned boundary layer meshes. The AMR techniques have been proposed for the refinement of mesh zones using the flow characteristics to achieve a proper mesh resolution in any part of the computational domain [44]–[49] AMR is useful because it can alter the mesh dynamically when there are changes in the flow characteristics, and therefore, AMR can be used to solve flow problems with high accuracy [50], [51]. AMR methods can be used in conjunction with linear and non-linear finite element formulations and therefore, these methods are typically used to solve FSI problems [52]–[55].

Many important applications involve incompressible viscous flows past arbitrary-shaped thin objects. However, the pressure discontinuities and velocity derivatives are neglected from the analytical solution using IBM. Second-order errors are also detected upon examining the interaction between the incompressible viscous fluid and thin object. Most IBMs are not really suitable to solve FSI problems involving thin solid since these problems require a high level of accuracy of the variables at the FS-interface. In other words, a high level of precision is required for the flow variables in order to capture the flow physics in the boundary layer. Kajishima et al.'s cut-cell method for incompressible viscous flows past thin objects results in second-order errors. In addition, a number of IBMs are dependent on the mesh size at the FS-interface in order to precisely capture the flow kinetics. In IBM, volumetric coupling is used to represent the FS-coupling and the artificial viscosity may either affect the structure or alter the fluid property such that the fluid becomes incompressible.

The maximum mesh resolution in fixed grid schemes must be specified for all the zones within the computational space, but for IBM, this step may not be needed. This issue can be addressed using the Dynamic adaptive mesh refinement (DAMR), which will minimize computational time during the simulations. In addition, DAMR can be used to achieve satisfactory performance for parallel computing using supercomputers in order to solve extremely demanding computational fluid dynamics (CFD) problems. The advantages of DAMR has spurred the development of a novel DAMR-IBM method with 2^d-stage velocity- pressure corrections for two-dimensional FSI problems involving thin objects [56], [57]. While Kajishima et al.'s method is efficient in treating the complexities at the FS-interface (which in turn, increases the accuracy of the solution), the DAMR algorithm deals with the momentum exchange between the fluid and solid for thin arbitrary-shaped objects. Changes in the velocity field are associated with the average local solid volume while the fluid velocity affects the conservation of mass.

Therefore, the purpose of this research is to create a novel DAMR-IBM method that combines IBM with 2^d-stage velocity- pressure corrections with local adaptive mesh refinement using sub-mesh layers of variable mesh sizes. These sub- grid layers constitute a structured Cartesian grid for the computational space. When compared to fixed grid schemes, the DAMR-IBM technique is designed to tackle FSI issues that involve incompressible viscous flows (IVFs) around sharp-edged thin solid with high accuracy and reduced computing time. The algorithm is created by layering deformable fluid nodes near the solid's immersed boundary. Accurate capturing of the flow characteristics at the FS-interface requires the use of a suitable mesh size for the boundary layer to reduce the computational cost. The developed DAMR-IBM algorithm in this paper is used to evaluate the numerical model's capabilities for 2-dimensional IVFs past a thin object. Finally, validation is carried out in order to define the accuracy and reliability of the DAMR-IBM algorithm. It is believed that the DAMR-IBM algorithm will be useful to academics and researchers involved in simulations of FSI problems involving thin objects.

METHODOLOGY

The focus of this research is a dynamic system that presents the interface between the limit of movement of a thin object and the flow of fluid. The thin object was thought to be completely rigid in this work. The interactions of the thin object's moving boundaries will be used for the description of the basic governing equations & the numerical approach. The fluid is assumed to be incompressible & Newtonian. The modelling of fluid flow is carried out by the finite difference method; the modelling is done in a Cartesian coordinate system using

rectangular grids.

2.1 Equations governing fluid dynamics

The Navier-Stokes (NS) equations were used for the description of the fluid dynamics for the considered FSI problem in this work. The NS-equations are given by:

$$\nabla \cdot u_f = 0 \tag{1}$$

$$\frac{\delta u_f}{\delta t} + u_f \cdot \nabla u_f = -\frac{1}{\rho_f} \nabla p + \nu_f \cdot [\nabla u_f + (\nabla u_f)^T] \tag{2}$$

where u_f is the velocity of the fluid, P is the pressure, ρ_f is the density of the fluid, and ν_f is the kinematic viscosity of the fluid. The Cartesian grid was used to organize the Eulerian variables. The second-order finite difference method in space and time was used to define Eqs. (1) and (2). Then, the exchange of momentum at the FS-interface was addressed using IBM as detailed in the following sub-section.

2.2 Sharp interface immersed boundary method

Predicting the flow properties and related forces, which requires prior understanding of the moving boundaries, is one of the hurdles in FSI problems. As a result, in this work, a coupling method was adopted, in which the fluid-structure representation was done as a node derived from a dynamic system. This was accomplished by iteratively and simultaneously embedding the entire governing equations. IBM [35] was used to study the exchange of momentum in cells that are partly occupied by solid particles. The velocity field in this work was introduced at the FS-interface, where solid particles partly occupied the cells, and IBM was used to solve the velocity field. To introduce the velocity field, the solid velocity (u_p) and the volume of local fluid velocity (u_f) in each nod was averaged. The velocity field is given by:

$$u = (1 - \alpha) u_f + \alpha u_p \tag{3}$$

where " $\alpha(0 \leq \alpha \leq 1)$ " is the solid particle's volume fraction within a cell (as seen in Fig 1). The velocity of the solid particle (u_p) was then decomposed for the translation and rotation of the elements such that $u_p = v_t + r \times \omega_p$, where v_t is object's velocity, (ω_p) is the angular velocity, and r is the relative position from the gravitational centre to a given point within an integral" area.

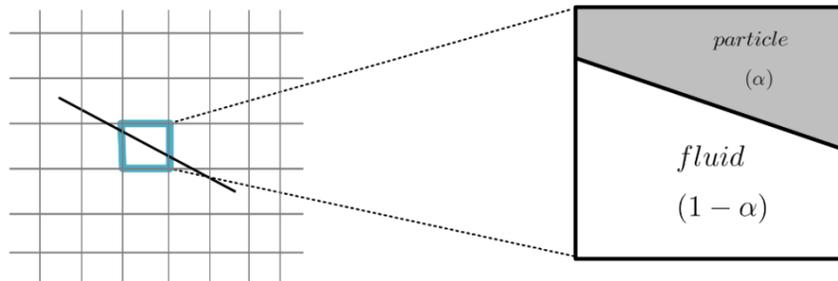


Figure 1: Fluid-solid volume fractions of a cell in the uniform Cartesian mesh.

The fraction of volume was determined using the hyperbolic tangent function as follows:

$$\alpha = \frac{1}{2} \left[1 - \tanh \left(\frac{\delta s}{\sigma \lambda \Delta} \right) \right] \tag{4}$$

$$\lambda = |n_x| + |n_y| + |n_z| \tag{5}$$

$$\sigma = 0.05 (1 - \lambda^2) + 0.3 \tag{6}$$

where $n = (n_x, n_y, n_z)$ is the normal outward unit vector at the surface of the element while δs is the distance from the element's surface to its midpoint. This formulation was first presented by Yuki [58] and termed surface digitizer. Hence, the fluid-solid boundary interaction was determined via integration of the velocity field using the NS-equations:

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho_f} \nabla p + H_u + f_p \quad (7)$$

$$H_u = -u \cdot \nabla u + \nu_f \nabla \cdot [\nabla u + (\nabla u)^T] \quad (8)$$

The following system was compute for the determination of the time improvement for u :

$$u^{n+1} = u_l - \Delta t \frac{1}{\rho_f} \nabla p + \Delta t f_p \quad (9)$$

$$u_l = u^n + \Delta t \left(\frac{3}{2} H_u^n - \frac{1}{2} H_u^{n-1} \right) \quad (10)$$

$$f_p = \frac{\alpha (u_p - u_l)}{\Delta t} \quad (11)$$

where Δt denotes the time increment while f_p represents the intruder effect on the fluid inside the object at the FS-interface. It shall be noted that when $(\alpha = 0)$ and there is zero interaction force, $f_p = 0$

For single-phase fluids, the time advancement was calculated using the second-order Adams-Bashforth technique. The fraction of particle in the cell was subjected to a particle force comparable to that found in Eq. (11) but with the opposite sign. The body force was used to replace the hydrodynamic tension at the FS-interface contact, and the particle's advancement phase was completed via integration of the particle force throughout the particle volume. Application of pressure at the FS-interface modified the numerical technique in a way that the force of body (f_p) was removed from Eq (7). Hence, the velocity of the particle can be expressed as follows based on the Navier-Stokes equations:

$$\frac{\partial u_f}{\partial t} = -\frac{1}{\rho_f} \nabla p + H_u \quad (12)$$

$$H_u = -u_f \cdot \nabla u_f + \nu_f \nabla \cdot [\nabla u_f + (\nabla u_f)^T] \quad (13)$$

The following system was compute for the determination of the time advancement for u_f :

$$u_l = u_f^n + \Delta t \left(\frac{3}{2} H_u^n - \frac{1}{2} H_u^{n-1} \right) \quad (14)$$

$$\frac{1}{\rho_f} \nabla^2 p^{n+1} = \frac{\nabla \cdot u_l}{\Delta t} \quad (15)$$

$$u_f^{n+1} = u_l - \Delta t \frac{1}{\rho_f} \nabla p^{n+1} \quad (16)$$

Where the time step is represented by the superscripts; Δt represent the increase in time, and u_l is the intermediate velocity. The computation of the time advancement was done in a single continuum using the

fractional step and second-order Adams-Bashforth methods. The deviation of (u_l) was used to solve the Poisson equation; this serves as a source term for pressure (p^{n+1}) . Then, the intermediate velocity was corrected to accommodate the time-step velocity (u_f^{n+1}) integration.

2.3. Second correction step after momentum exchange

Most IBMs are susceptible to pressure oscillations, particularly in flows around thin solid. After momentum exchange, a second pressure correction step can be implemented to eliminate oscillations. As demonstrated in Eq. 3, the contact force was exposed to the velocity field in fluid cells filled with solid particles. As a result, proper treatment was necessary to retrieve the pressure field that was affected by the alteration. A second pressure derivative was applied to the affected cells if there is error in the pressure field. A small area was selected near the object for the correction of u in Eq. (3), denoted as u_2^{n+1} . The scalar values for the pressure and velocity corrections $(u_3^{(n+1)}$ and $p_2^{(n+1)})$ were also determined using the simplified marker and cell approach. The following equations were computed to rectify the pressure readings at the surface of the object using the projection method [56], [59]:

$$\nabla^2 \phi^{n+1} = \frac{\nabla \cdot u_2^{n+1}}{\Delta t} \quad (17)$$

$$u_3^{n+1} = u_2^{n+1} - \Delta t \nabla \phi^{n+1} \quad (18)$$

$$p_2^{n+1} = p_1^{n+1} + \phi^{n+1} \quad (19)$$

Here, " u_3^{n+1} and p_2^{n+1} " represent the velocity & pressure respectively, that satisfy the continuity equation for the following time" step.

BOUNDARY LAYER AND LOCAL ADAPTIVE MESH REFINEMENT

The goal of adding AMR is to improve the resolution and precisely capture the flow behaviour in areas of expected high flow gradients, such as areas near a thin object's surface; this is expected to improve the IBM solution's accuracy. In the boundary layer, the requirements for the mesh resolution may differ dramatically from the requirements in the computational domain's free stream region, where there are minimal flow gradients. One technique to increase the solutions accuracy at the FS-interface is to use AMR systems. Because the flow field might change substantially owing to the presence of a thin object, it's critical to record the flow mechanics in the boundary layer appropriately. For different sub-blocks of the refined mesh in this work, the connected nodes in the cells were registered using a local AMR approach. A mesh hierarchy with specified grid size was only defined for the computational domain's critical region in these sub-blocks. Being that each mesh layer has a temporary time step that is adjusted according to the cell size, the manner of the mesh layers overlap allows for time refinement. To assure the numerical scheme's stability and accuracy, the permissible time for meshes was ensured to match with the fine mesh cells size. This allows that larger cells can be modified with longer time steps selectively if considering meshes with mixed cell sizes in isolation. It should be noted, however, that each cell must be modified for each set time step and iteration. Smaller cells are connected with shorter time steps in the local AMR, and vice versa. The cells are divided into independent meshes with the same mesh size for each group of cells.

The computations were done using the AMR technique; for the computation, the mesh were interpolated to produce new ones via computation static flow in the very thin mesh refinement layer. The preliminary fluid mesh nodes were built with the nodes at intervals $\Delta X = 0.1$; the final nodes size was 160×40 (6400) with $\Delta x = 0.0125$. A level-wise increment in the refinement the indirect boundary, the refinement was increased by

one level at each step. In summary, the addition of the finest AMR nearby the thin solid can increase the accuracy of the fixed grid approach. It should be emphasized, however, that while accuracy of the fixed grid scheme can be improved by AMR, it does not help in estimating the structure's motion. The mesh may require frequent update to adhere to the implicit interface due to transient instability (particularly for large-item motion). Therefore, it is beneficial to ensure a layer of small sized cells along the object's surface to ensure that the boundary layer has the optimal mesh size.

A rectangular grid is created in the AMR “by properly layering the sub-blocks to represent the whole computational domain while assuring a low degree of error and a short computation time. The modification of the time step is based on the mesh refinement level. The mathematical representation of the mesh refinement time step can be represented as $\Delta t = \Delta T / 2^N$, where $N = 1, \dots, p_{lev}$ and p_{lev} is the number of cell size-correlated patches.” Because of the pre-calculation of the time step for the solution at any mesh refinement level, a mechanism was designed to achieve the solution level synchronization. As indicated in Figure 2, the grid sub-blocks are introduced from the rectangle's boundaries. The AMR concept is depicted in Figure 2 for a single layer. The intermediate starting points are the original background mesh and the mesh bounds for Grid 1 while the fluid motion is used to determine the mesh for Grid 2. To achieve finer resolution mesh refinement, a rectangular grid is created from the boundaries of Grid 2.

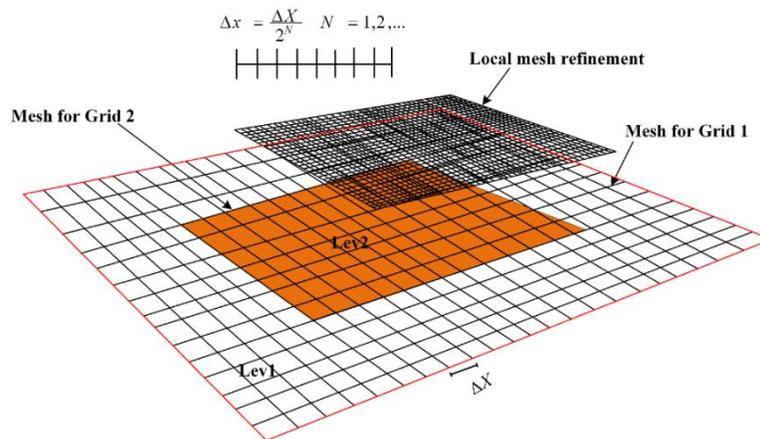


Figure 2: Two-dimensional adaptive mesh refinement for a single layer

3.1 Dynamic adaptive mesh refinement around thin object

In computationally demanding regions, DAMR was employed to solve partial differential equations. The first step of hierarchical adaptive mesh refinement is to identify the sections of the computational domain that needed special attention, followed by a dynamic layering of fine grids in these areas. A coarse grid is utilized as the base in the DAMR algorithm, with the computational domain's minimum permitted resolution. To improve the resolution of the solution, finer grids are used for regions with increasingly demanding computations as the solution develops. Using the same previous method, the new fine grid is improved further to produce a finer grid, resulting in a mesh hierarchy that is dynamically adaptive. Figure 3 showed the dynamically adaptive mesh hierarchy created with the DAMR scheme. The DAMR-IBM approach improves thin object boundary solutions, which need substantial computations to resolve flow physics in high flow gradient regions while coarse mesh is implemented in the regions with low flow gradients. The Cartesian mesh was coarsened or refined based on the distribution of the fluid nodes. Nonetheless, two mathematical issues arise as a result of grid resolution adjustments, which can be solved by dispersing fluid nodes across multilayer interfaces. The fine and coarse sections must have a similar cell shape; it is critical to avoid the formation of false errors at the boundaries, as these errors might cause pseudo-waves. Hence, the DAMR-IBM was developed to address this issue. The shape function in this case is constructed without implementing multiple geometrical functions within the computed regions. The Cartesian mesh is used for the IBM simulations while DAMR is effective to address complex structural motions. Table 1 shows the monolithic DAMR-IBM algorithm used to coarsen and refine the

mesh for a single time step. For example, when a cell enters into a fine mesh, the cell is split into two sub-cells. The DAMR-IBM algorithm is implemented using an in-house FORTRAN code in order to enhance the resolution of two-dimensional incompressible viscous flows within proximity of the thin object surface and boost computational efficiency. Figure 3 shows the grouping of the cells for the background mesh and mesh refinement for dynamic surface tracking. The Cartesian mesh with two levels helps in overcoming the issues associated with simulations of flows around thin solid. The finest mesh level is applied close to the surface of the thin object and requires higher computational power.

Mesh hierarchy computations must be performed by interpolating data from the coarse and finer mesh blocks. Due to the unpredictability of time, all node numbers must be inserted from the old grid to the new grid. The linear interpolation strategy, as seen in Figure 3, was used in this extremely easy procedure. The first stage is the positioning of the initial node (N_{old}) in the Cartesian grid, and the second stage is the formation of new nodes (N_{new}) for the new grid. In this case, the number of in (N_{new}) comes from the local representation (N_{old}) . A specific procedure is used to locate (N_{old}) . The smooth set function is employed as an advanced stage in the interpolation technique in continuous space. A one-dimensional search is used to locate the primary node (N_{1new}) in relation to N_{old} . This one-dimensional search is implemented using the steepest descent algorithm.

The next step is to interpolate the next node (N_{2new}) that is lodging onto (N_{1new}) in search of the related N_{old} . This process is repeated until all the new nodes have been located on the old mesh. For the search algorithm, the computational cost varies linearly with the number of nodes N in the new mesh.

Table 1: Monolithic IBM-AMR algorithm used to coarsen and refine mesh for a single time step.

“Solve the Navier-Stokes equations for the entire computational domain using the method of projection.”

“Compute d_x and d_y for the small rectangular region around the thin object with respect to the degree of refinement:

$$dx = DX, DY / 2^N \quad dt_{new} = DT / 2^N \quad \text{where } N = 1, 2, \dots, n \text{ .”}$$

“Interpolate U_1 and V_1 for the boundaries of the rectangular region around the thin object”

“Compute the boundaries of the refined mesh node around the thin object:”

$$u_{C_{1:NX+1}}^{2:NX+1} \quad v_{C_{1:NX+1}}^{2:NX+1}$$

“Determine the volume fraction for the solid in the local cells and compute the velocity.”

“Compute the momentum exchange.”

“Compute U_D and V_D for the volume fraction based on the solid digitizer.”

“Compute the intermediate velocity and pressure $(u_1^n \text{ and } p_1^n)$ using the second-order Adams-Bashforth method and fractional step method.”

“Call CONS to compute d_x and d_y and update the values accordingly.”

“Correct the velocity and pressure $(u_3^{n+1} \text{ and } p_2^{n+1})$.”

“Compute the velocities $(u_{pr} \text{ and } v_{pr})$ for angular particles using the second-order Adams-Bashforth method and update θ ”

“Compute the coordinates $(x_{pr} \text{ and } y_{pr})$ for the moving particles using the Crank-Nicholson method.”

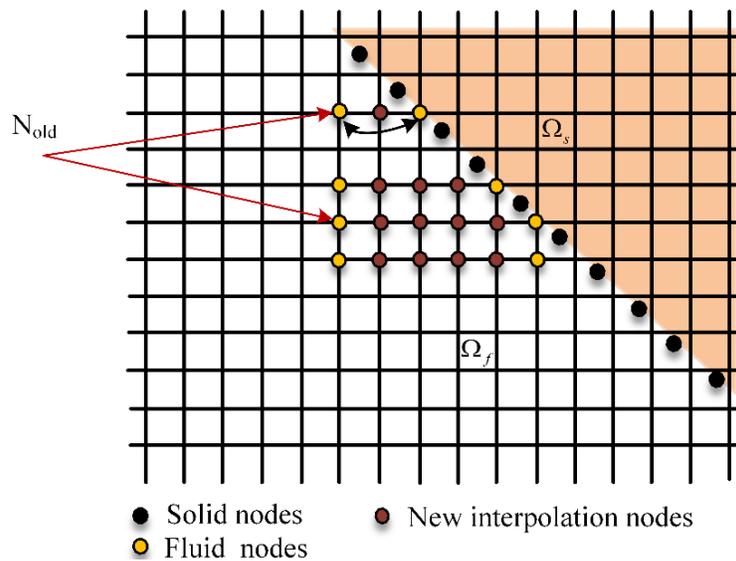
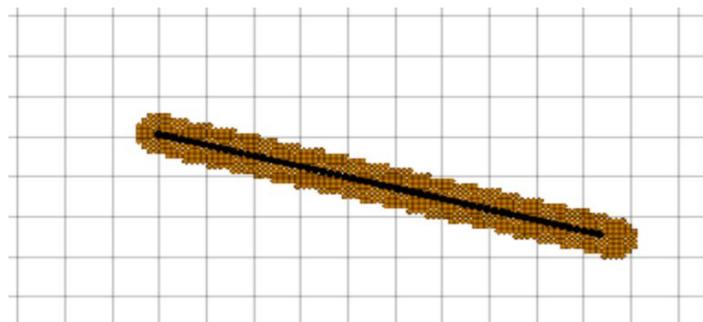
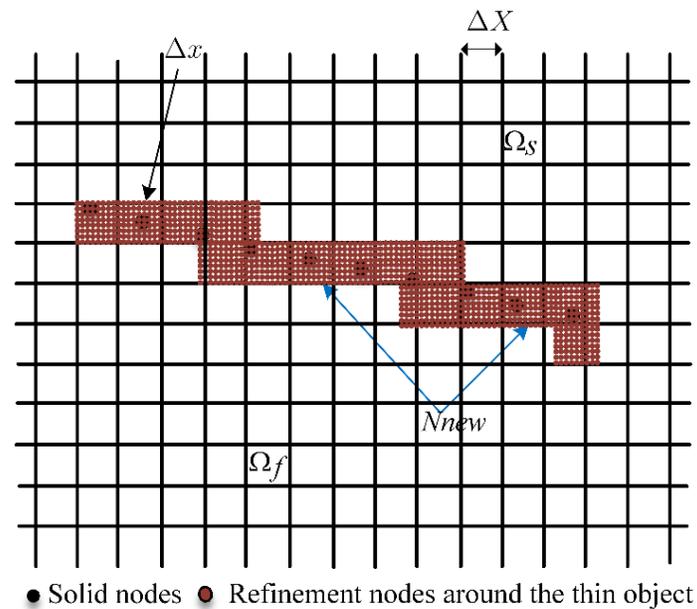


Figure 3: Depiction of the Eulerian & Lagrangian meshes (where Ω_s = solid area, Ω_f = fluid area). Also represented are the coefficients of velocity interpolation from the locally refined mesh at the fluid-solid boundary. The interpolation of the velocities at the new nodes from the old nodes was done within the square region.

3.2 Coupling between DAMR and IBM

This section is focused on the development of an algorithm to control the local resolution when the mesh is generated for thin solid. Figure 4(a) shows the box window, which is the user-input keyword used for mandatory resolution of the mesh whereas Figure 4(b) shows the surface refinement, which is the user-input keyword used to improve the mesh resolution within vicinity of the thin solid. Figure 4(c) shows the layer window, which is the user-input keyword used to strengthen the new mesh resolution at a distance from the thin object. Surface refinement is necessary in order to achieve the desired mesh resolution within proximity of the thin solid, which in turn, will increase the accuracy of the solutions achieved using IBM. The layer window aids in the creation of a seamless mesh transition from the thin object's boundaries to the far-field region. The Ubuntu 16.04 LTS OS was used to write the codes in FORTRAN. The open-source code VisIt 2.12.3 was used in this work to show the majority of the 2-D simulation results. Also, the VisIt 2.12.3 programme can handle massive datasets and aid in visualization of complex fluid flows via parallel graphical analysis.

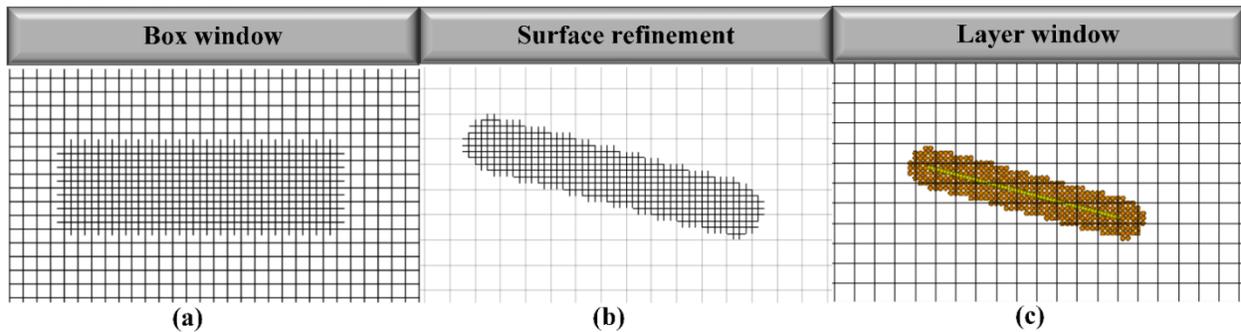


Figure 4: User-input keywords for local DAMR close to the thin object.

The hybridization of DAMR and IBM was tested using the test scenario below. The proposed algorithm was used to create a prototype grid with 30,000 nodes and a revised mesh with 6,400 nodes. For the 2-D nal simulations, the mesh refinement was applied using the mesh generator. Figure 5 depicts a mesh that has been fine-tuned locally around a thin solid. The major goal is to keep track of the mesh's dynamic mobility while avoiding unnecessary computations in a limited section of the computational domain. In the improved mesh, there are around 6,400 nodes in the neighborhood of the thin solid, where $\Delta t = 0.0000625$ s.

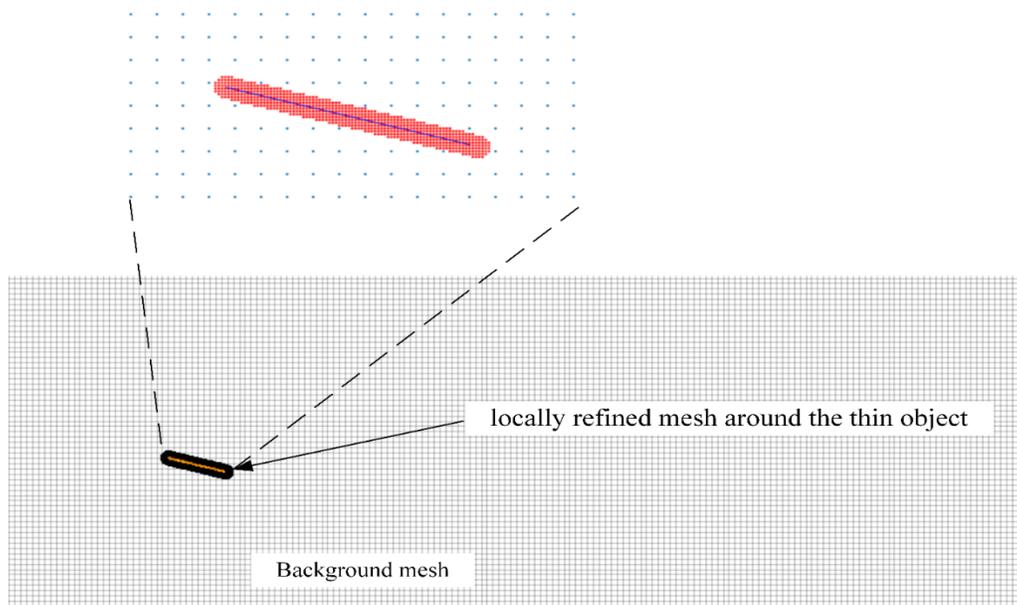


Figure 5: “Illustration of the background mesh and locally refined mesh within vicinity of the rigid thin object.”

SIMULATION SETUP

The features of 2D incompressible viscous flow near a thin object were investigated using simulations. As illustrated in Figure 6, the computational domains size was $20L \times 15L$ and the thin object's length was L . The thin object's leading edge was placed at $(5, 7.5)$ from the inlet. The parameters settings for the flow characteristics simulation using IBM are listed in Table 2. A consistent Cartesian grid was utilized, with equal length & width $\Delta x = \Delta y = 0.1$ in each cell. The number of nodes was set to $N_x \times N_y = 200 \times 150$, and the time step was set to $\Delta t = 0.005s$. For the initial velocity, it was set to $U_o = 1 m/s$. The no-slip boundary conditions for the computational domain (upper and lower walls) were defined. Following the recommendations of [60], the fluid density was fixed to $\rho = 1.0 kg/m^3$ and the (Re) was set to 200. The refined mesh zone has a size of $2L \times 1L$, with each cell having equivalent length & width of $\Delta x = \Delta y = 0.00125$. The refined mesh zone has a time step of 0.000625 . The inclination of the “rigid thin object was done at 3 different angles of attack (15, 30, & 45°). The computational domain’s outlet pressure was set as gradient-free. The thin object was represented by 21 Lagrangian nodal” points.

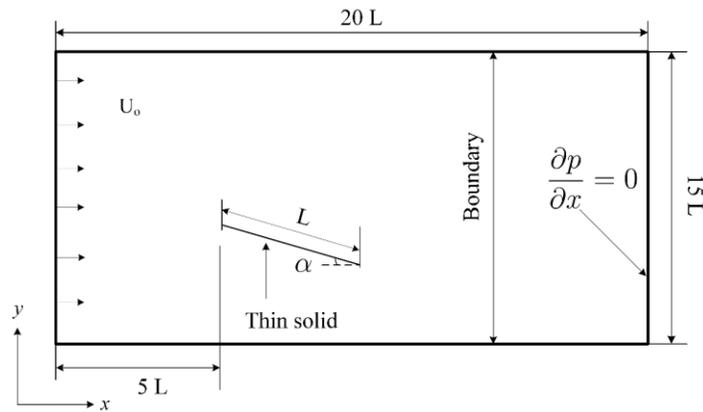


Figure 6: Depiction of the utilized computational domain for the simulation of the 2-D incompressible viscous flow around a thin solid inclined at an angle of attack, (α) .

Table 2: Simulation parameters.

Parameter	Preliminary mesh	Refined mesh
Number of cells	4. 200×150	160×40
$Re = \bar{U}D / \nu$	200	200
Time step, Δt	0.005	0.000625
Cell size, $\Delta x, \Delta y$	0.1	0.0125
Angle of attack, $\alpha (^{\circ})$	5. 15, 30, 45	6. 15, 30, 45

5. VALIDATION OF THE DAMR-IBM ALGORITHM

The DAMR-IBM algorithm was validated using the work of [61] as the benchmark case. The Reynolds number (Re) for laminar flow past a cylinder was set at 100. The Re can be calculated depending on the cylinder's diameter D in 2-D simulations using the relation $Re = \bar{U}D / \nu$. Based on the results of

the simulation, the coefficient of drag C_D and coefficient of lift C_L were determined using $C_D = 2F_D / \rho U_m^2 D$ and respectively. The integration of the Lagrangian space with the immersed boundary forces yielded the lift & drag coefficients. The DAMR-AMR algorithm was also validated by calculating the Strouhal number (St) given by $St = Df / \bar{U}$. The pressure interval approach was used to determine the pressure from the fluid node nearest to the cylinder's boundary. The Strouhal numbers and the lift and drag coefficients for laminar flow around a fixed cylinder at $Re = 100$ are shown in Table 3.

Two sets of results were obtained in this study — the first set pertains to results obtained from the DAMR-IBM algorithm whereas the second set represents the results obtained by the OpenFOAM software. The DAMR-IBM algorithm clearly obtained similar drag coefficients, Strouhal numbers, and lift coefficients as those obtained by [61] and [62] as well as the OpenFOAM software. Figure 7 shows the variations of the drag and lift coefficients at $Re = 100$ and it can be seen that there is very good agreement between the results obtained from the DAMR-IBM algorithm and OpenFOAM software throughout the simulations.

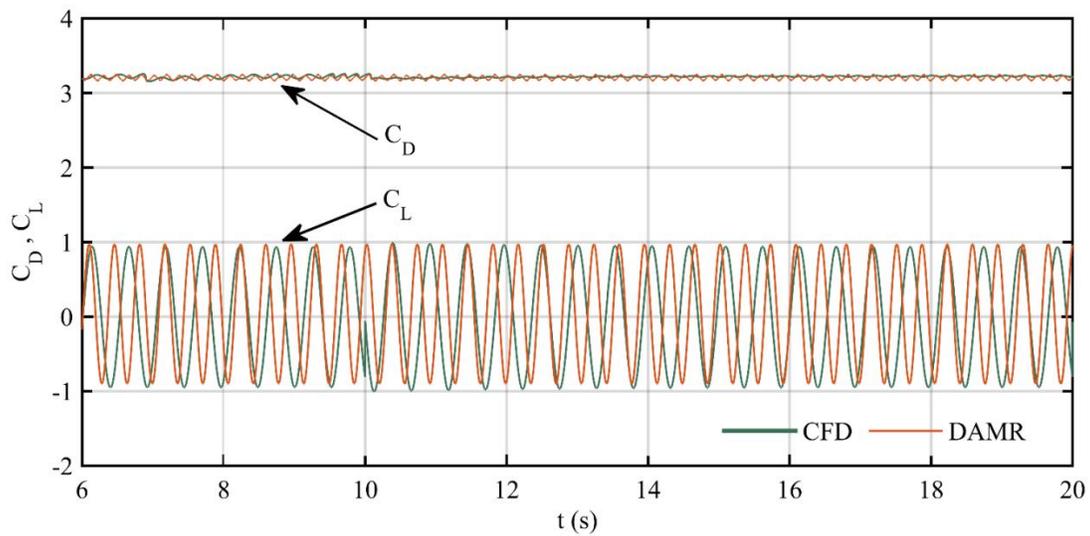


Figure 7: Comparison of the obtained coefficients lift and drag from AMR-IBM and open-source CFD software for 2-D laminar flow nearby a cylindrical fixed object at $Re = 100$.

Table 3: Comparison of the obtained coefficients of drag C_D , lift C_L , and Strouhal numbers St from AMR-IBM and those obtained the published studies and OpenFOAM software for 2-D laminar flow around a cylindrical stationary at $Re = 100$.

Source	C_D	C_L	St
Present work	3.255	0.9913	0.2997
[61]	3.22–3.24	0.990–1.010	0.295– 0.305
[62]	3.2258	0.98934	0.30061
OpenFOAM software	3.227	0.9723	0.2985

RESULTS AND DISCUSSION

The DAMR-IBM algorithm was tested for its accuracy and robustness in the simulation of 2-D incompressible viscous flows around a thin solid; the outcome of this analysis was analysed and presented in this section. At this point, it's worth repeating the goal of this work, which is to design an AMR technique that works for both moving and stationary thin solid. The AMR technique is expected to be capable of addressing the interactions of deformable structures (both compressible and incompressible) subjected to extreme deformation in incompressible flows, as well as resolving the flow characteristics in the boundary layer near the thin object's surface. The described DAMR-IBM in this paper is capable of addressing thin object deformations in a cost-effective way while maintaining high resolutions around the thin object's boundaries throughout simulations; hence, the proposed approach is ideal for simulating real-world FSI situations. Mesh layers of different grid sizes were generated in the DAMR-IBM algorithm in the layers of boundary where there are large flow gradients, such as flow re-attachment and flow separation. The effectiveness of the DAMR-IBM was simulated using different algorithms in 3 case scenarios which are “flow around a stationary rigid thin object (simulated with IBM with no mesh refinement), flow around a moving thin object inclined at different angles of attack (simulated with DAMR-IBM), and flow around a stationary thin object inclined at different angles of attack (simulated with AMR-IBM)”.

6.1 Flow around a fixed rigid thin solid

The performance of the IBM algorithm (without mesh refinement) in modelling flows around a fixed rigid thin solid was focused on this section. The grid size of the fluid is larger than the length of the solid (thin object). The velocity contours of the flow around the thin object without mesh refinement are depicted in Figure 8, where Figure 8(b) shows a significantly higher pressure near the thin object's leading edge as illustrated by the small red region. In contrast, the pressure decreases at the upper surface and trailing edge of the thin object. The fluid particles accumulate and move in a different direction when the particles collide with the leading edge of the thin solid whereas there is a lower number of fluid particles accumulating at the rear of the thin object.

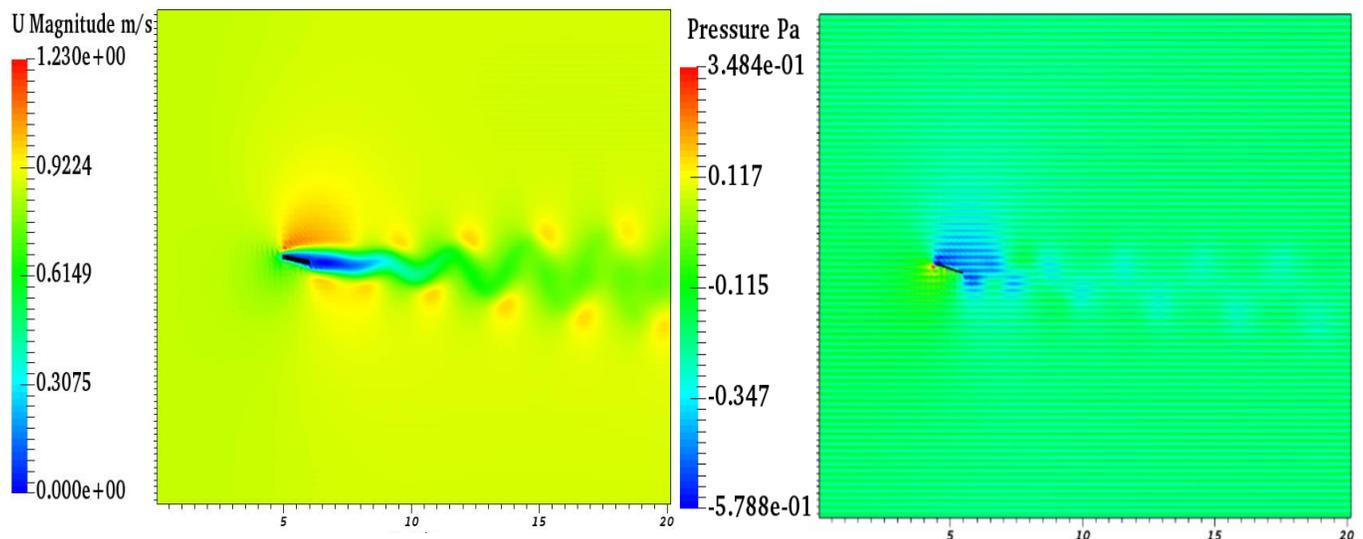


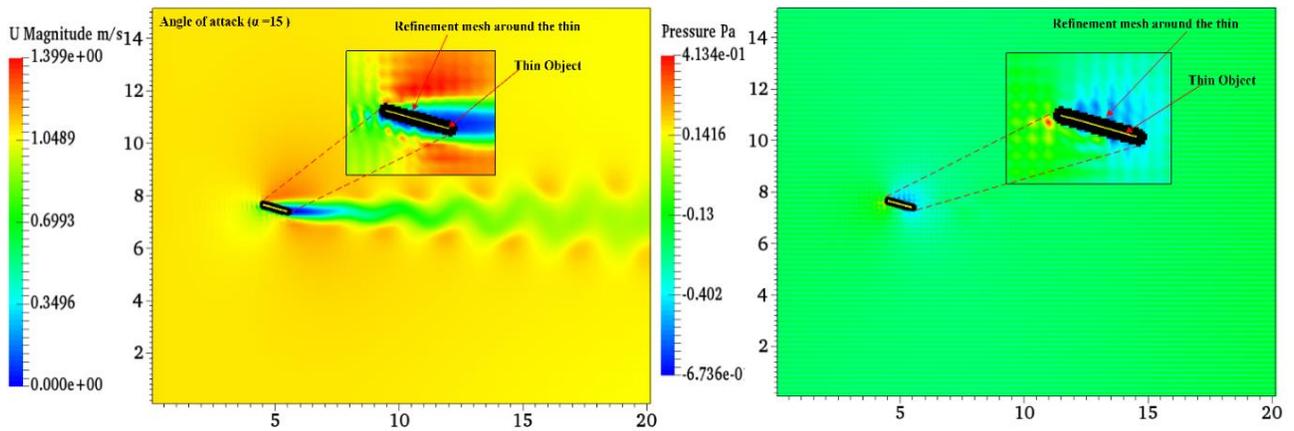
Figure 8: (a) Velocity contour and (b) pressure contour for flow around a fixed rigid thin solid obtained using the IBM algorithm without mesh refinement.

6.2 Flow around a fixed thin object inclined at different attack of angles

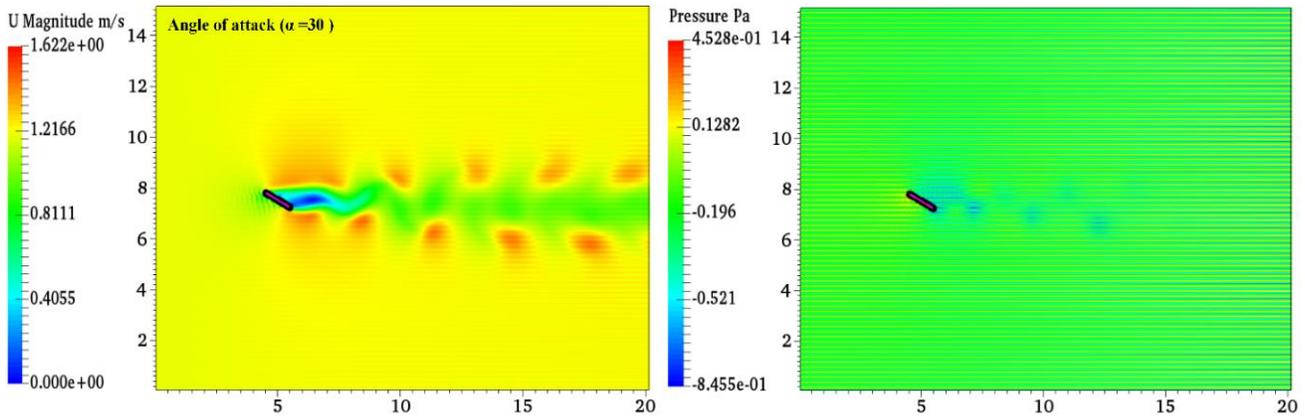
The algorithm was used in this case to model the flow around a fixed thin solid inclined at 15, 30, & 45 degrees. The flow is discovered to be fully developed at $t = 8 s$. The various attack angles were selected to mimic the insect wings flapping scenario. The pressure and velocity contours of the studied flow at $t = 8 s$ are shown in

Figure 9. Owing to the sudden jump in the flow within the computational domain as a result of the inclination of the thin object in the freestream direction, there was a dramatic increase in the surface average pressure, causing oscillations in the force exerted on the object. As the flow reaches a steady state, there was a decline in the average pressure on the surface.

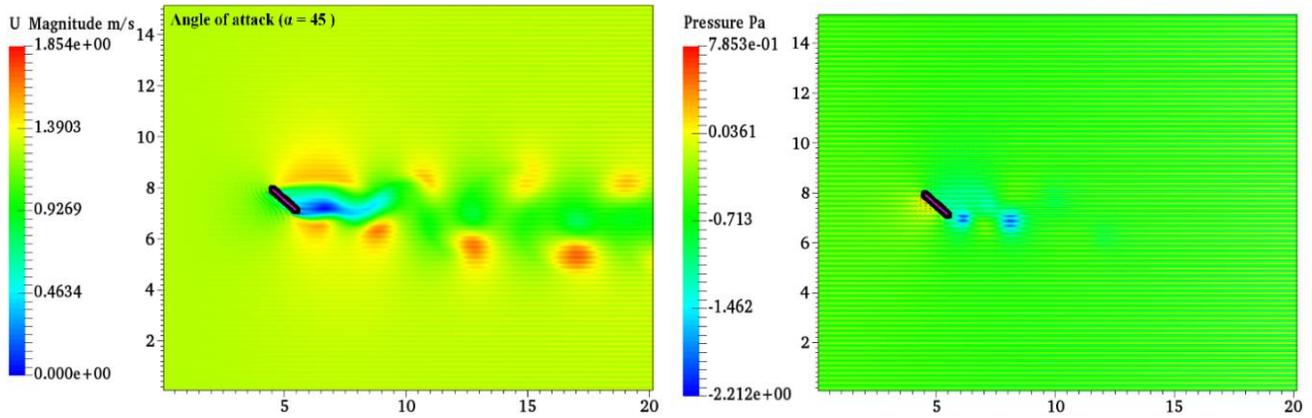
Figure 9a depicts the velocity field in the computational domain, as well as the velocity field near the thin solid tilted at $\alpha = 15^\circ$. Figure 9 also shows the location of the finest mesh. With the AMR-IBM technique, the observed computational time for this test scenario was 2,756.78 s (~45 minutes). But on the IBM with full mesh refinement, the observed computational time was 164,101.27 s (~46 h), meaning that the computational time for the AMR-IBM technique is about ~98.3 percent of the time it takes to complete full mesh refinement. This test scenario demonstrates the effectiveness of AMR in using IBM to compute a solution. In addition, the implementation of AMR based on local grid refinement nearby the stationary rigid thin object improves the degree of accuracy of the solution obtained using IBM.



(a)



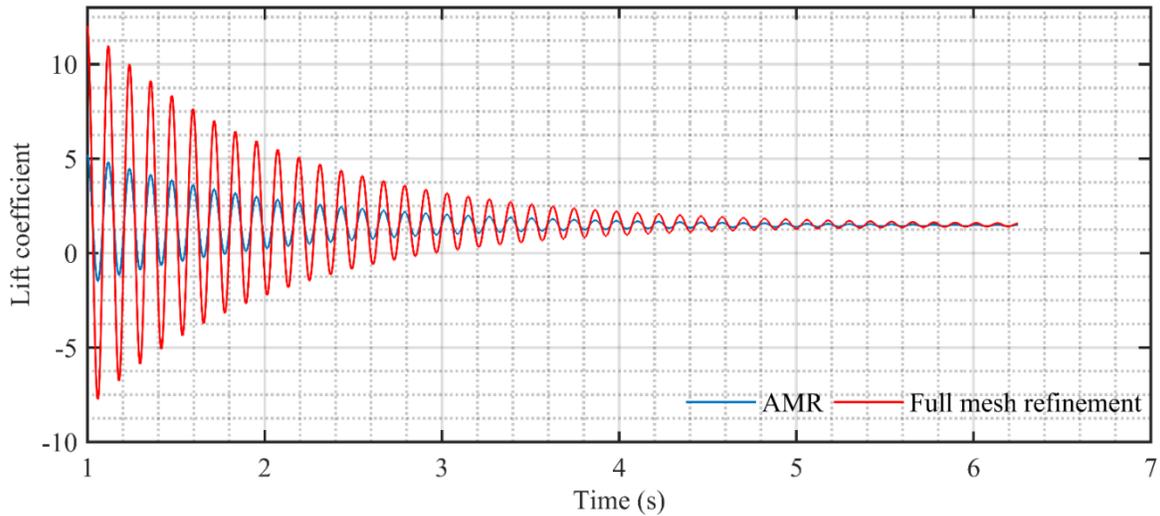
(b)



(c)

Figure 9: Velocity and pressure contours at different inclination angles of attack for flow around a fixed rigid thin solid as obtained using the AMR-IBM algorithm: (a) $\alpha = 15^\circ$, (b) $\alpha = 30^\circ$, and (c) $\alpha = 45^\circ$

Figure 10 illustrates the lift and drag coefficients as a function of time, as calculated using the AMR-IBM method and the IBM technique with full mesh refinement. Based on the OpenFOAM-5.0 software, there is a good agreement between the results obtained with the AMR-IBM and the IBM technique with full refinement mesh. The AMR-IBM approach, on the other hand, is preferable because of its simplicity, which requires no high-level computations, thereby saving time. It may be concluded that the AMR-IBM technique is appropriate for solving real-world FSI problems.



(a)

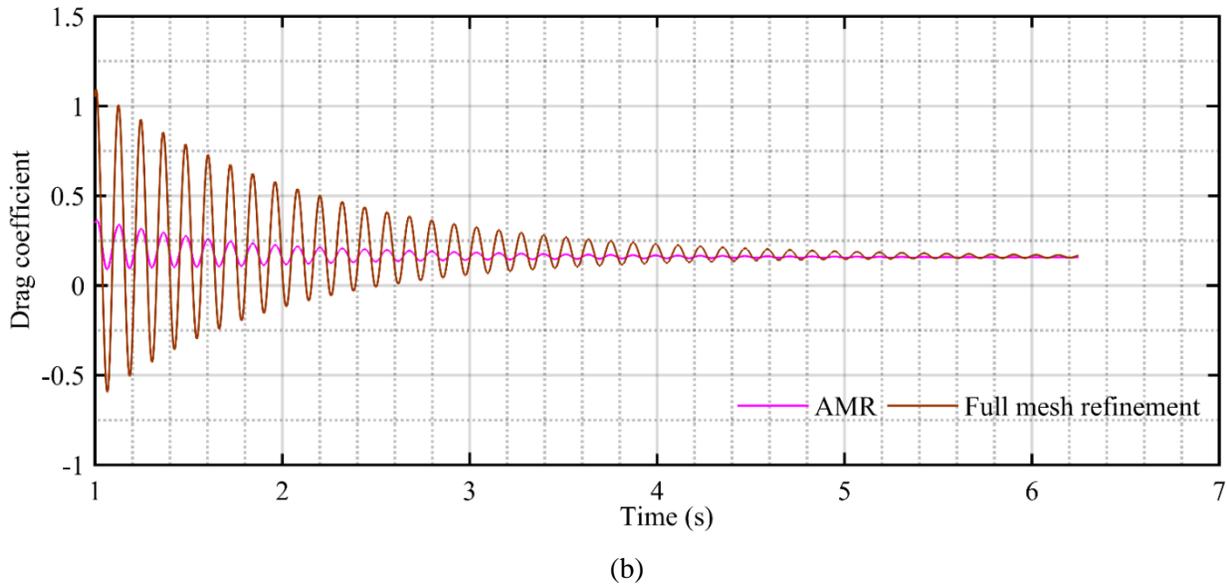


Figure 10: “Variations of the (a) lift coefficient and (b) drag coefficient with time for flow around a fixed rigid thin solid at (a) $Re = 200$ obtained using the AMR-IBM algorithm. The lift and drag coefficients obtained from the IBM technique with full mesh refinement are plotted for comparison.”

FSI problems that involve moving objects (where there is a substantial deformation on the thin solid surface) requires constant update of the mesh to account for changes in the implicit interface. It is preferable in such instances to produce a thin mesh layer near the thin object's boundaries to ensure the maintenance of an optimal mesh in the boundary layer at all times. The next section shows the performance of the DAMR-IBM method in 2-D incompressible V-L flow modelling around moving thin solid.

6.3 Flow around a moving thin object inclined at different angles of attack

This section evaluates the modelling performance of DAMR-IBM in the simulation of flows around a moving thin solid. As in the previous case, the same computational domain was maintained in this case. Similarly, the point of origin in the computational domain for the thin object is the same as in the preceding example. Figure 6 depicts the computational domain while Table 2 presents the employed simulation parameters. No-slip boundary constraints were also imposed on the upper & lower walls of the computational domain. The rigid thin object (depicted by 21 Lagrangian nodal points) was inclined at 15, 30, & 45 degrees. At the outlet of the computational space, the pressure was set to be gradient-free. All the simulations were done at a fixed Re number. The thin object's two edges were unbound, allowing it to move around.

Dynamic changes were noticed on the refined mesh around the thin object's boundaries due to the movement of the object. The pressure & velocity contours presented in Figures 11–13 are generated for the incompressible V-L

flow around moving thin solid for 3 angles of attack at different time steps ($time = 1.5, 3.5, 8.0 s$). The simulations revealed the following motions: (1) thin solid rotation around its gravitational centre, and (2) thin solid motion in the fluid field. The thin solid spins and revolves in a clockwise motion as a function of time. As the simulation progresses, the thin object is horizontally pushed by the fluid forces, while the forces originating from the pressure gap between the lower and upper surfaces pulls up the thin solid. Note that despite the thin solid depicting an air-foil in the simulations, it is also capable of representing other objects, such as various types of insect wings, minuscule fish fins, and valves of aortic. This study is focused on the thin solid response to the fluid forces operating on it. A static Cartesian grid was compute to calculate the flow dynamics while IBM was used to interpolate the flow parameters between adjacent cells. The overall force of fluid that acts on the IB was computed via extrapolation from the node points and used in the structural dynamics solver as the external force.

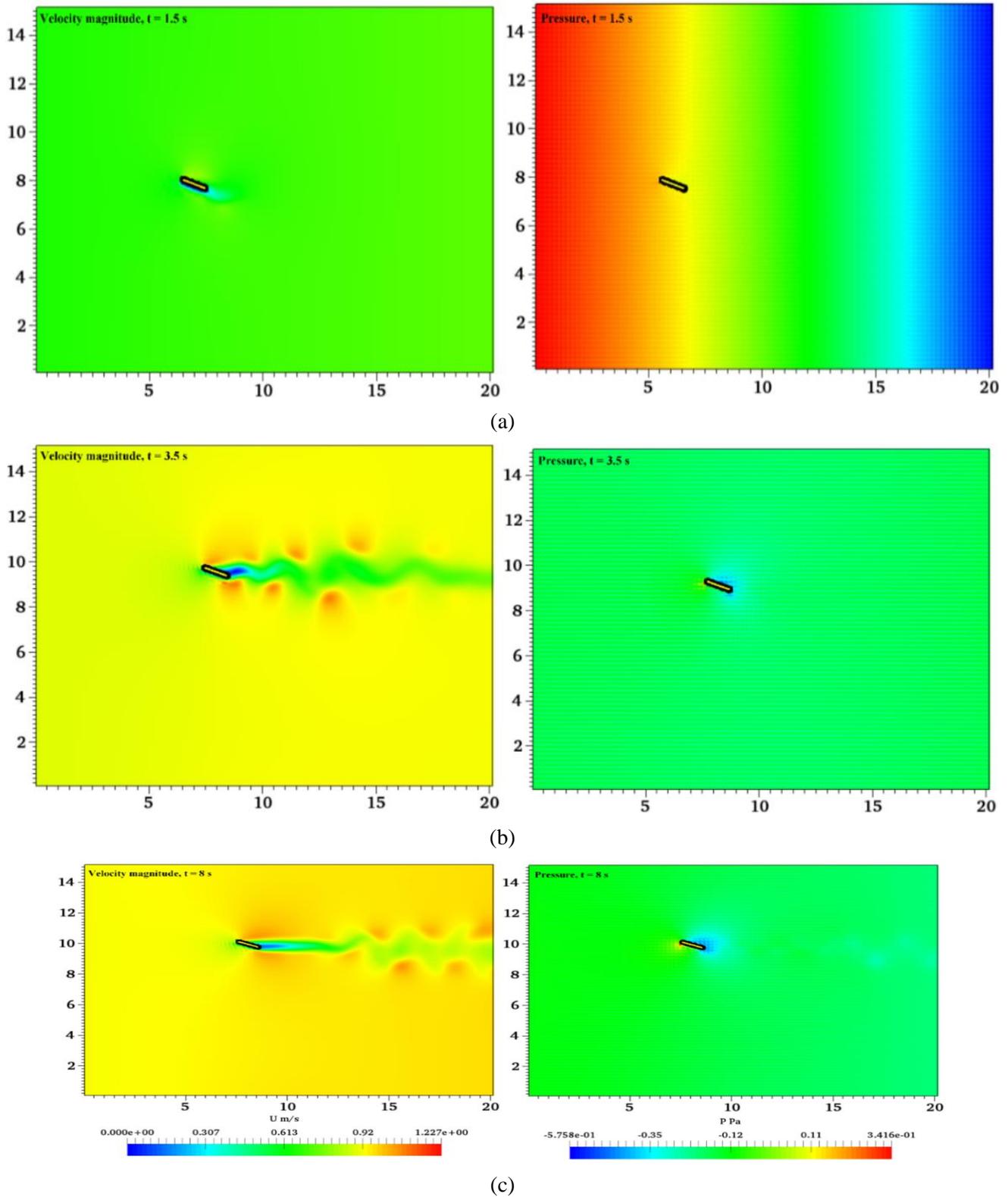


Figure 11: “Velocity and pressure contours of the moving thin object inclined at $\alpha = 15^\circ$ in the x - y plane at (a) $t = 1.5$ s , (b) $t = 3.5$ s , and (c) $t = 8$ s obtained using the DAMR-IBM approach.”

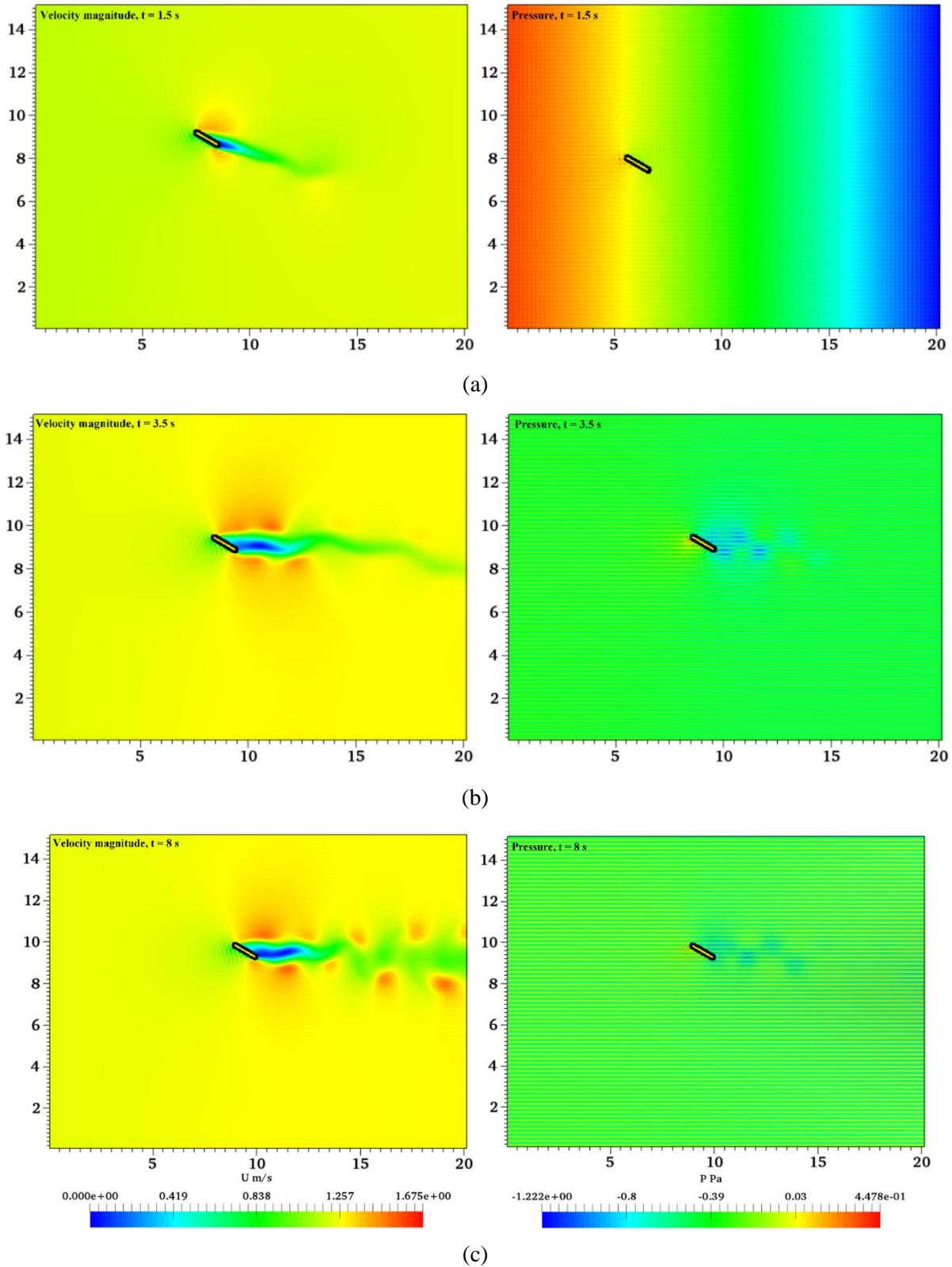
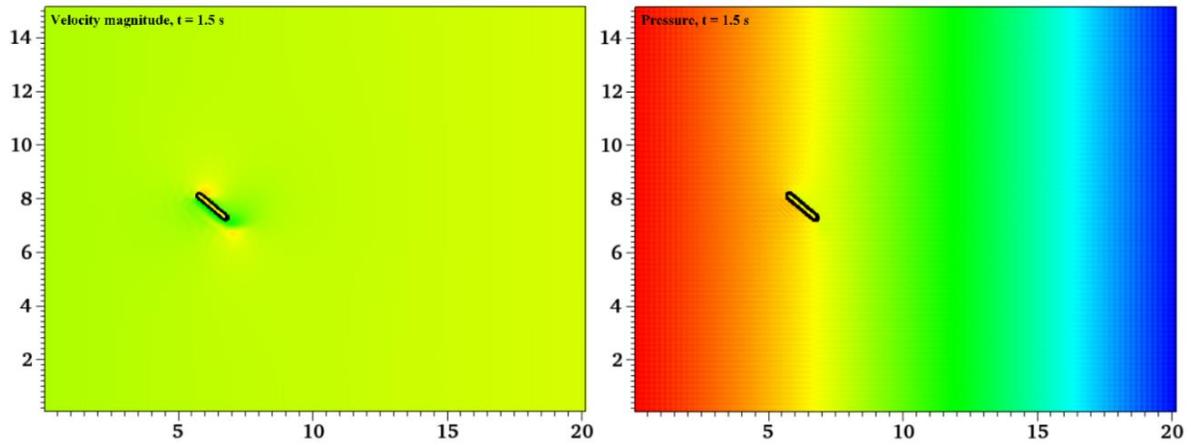
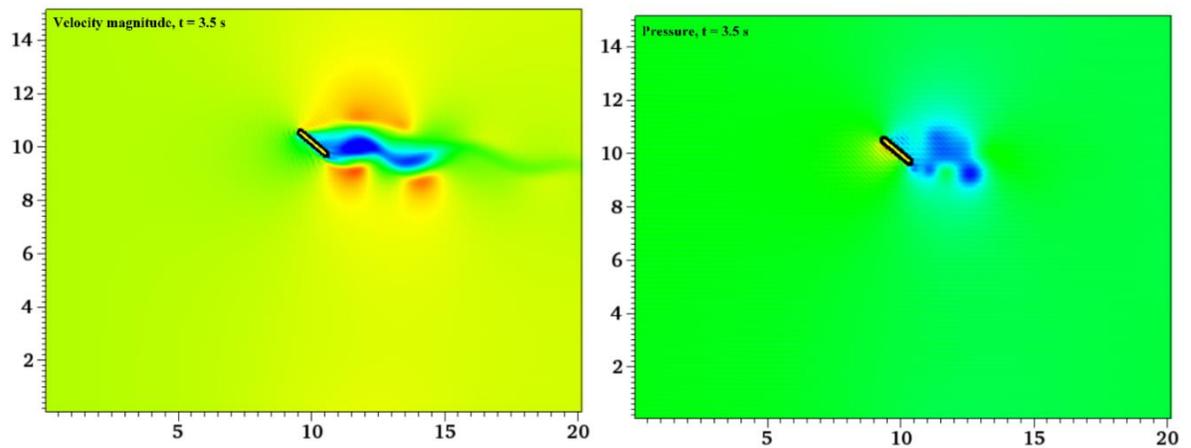


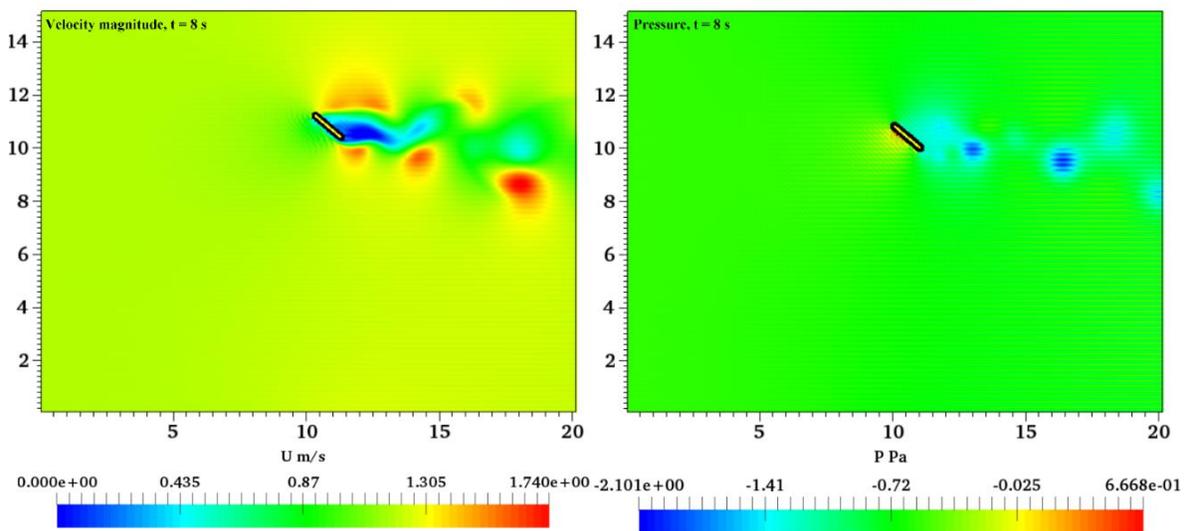
Figure 12: “Velocity and pressure contours of the moving thin object inclined at $\alpha = 30^\circ$ in the x - y plane at (a) $t = 1.5$ s , (b) $t = 3.5$ s , and (c) $t = 8$ s obtained using the DAMR-IBM algorithm.”



(a)



(b)



(c)

Figure 13: “Velocity and pressure contours of the moving thin object inclined at $\alpha = 45^\circ$ in the x - y plane at (a) $t = 1.5\text{ s}$, (b) $t = 3.5\text{ s}$, and (c) $t = 8\text{ s}$ obtained using the DAMR-IBM algorithm”

Figure 14 depicts the thin solid position at 15°, 30°, & 45° angles of attack at $t = 8\text{ s}$. Due to variations in the angle of attack, there were clear variations in the thin solid position. The lift and drag forces acting on the thin object are affected by the angle of attack, which alters its position. Over time, the thin object's rotation around its gravitational center will lessen this impact.

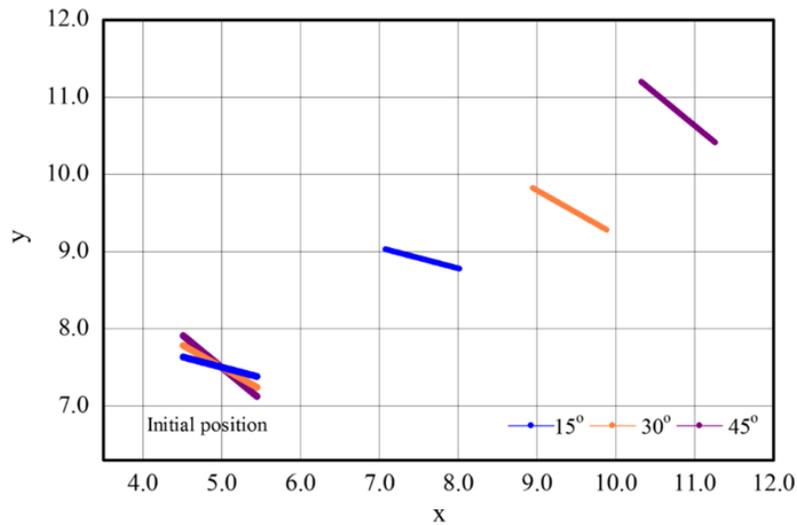


Figure 14: Location of the thin object due to differences angle.

CONCLUSION

The simulation of flows at the FI-interface for thin solid using the DAMR-IBM method with two-stage pressure-velocity correction was proposed in this work. high resolution flow characteristics near the thin solid surface with expected flow gradients were achieved using local adaptive mesh refinement while maintaining a moderate resolution elsewhere in the computational area. In most cases, a very fine mesh is applied throughout the computational domain, which is impractical because of the high computational cost. In this study, DAMR was integrated with IBM, which reduces the computational cost without compromising the accuracy of the solution in simulations of two-dimensional incompressible viscous flows nearby a thin solid. Furthermore, the DAMR-IBM developed algorithm addresses the issues concerning mesh refinement in the boundary layer, where thin layers of nodes are applied close to the thin object. A uniform Cartesian mesh with relatively wide cells was used throughout the computational domain except from the region within proximity of the thin object, which significantly reduces the computational cost. The DAMR algorithm is more suitable to resolve flows around thin solid compared with static mesh algorithm due to the fact that thin solid tend to deform easily. The developed DAMR-IBM outperformed the earlier reported algorithms; hence, the developed algorithm can be used to simulate flows around moving and stationary arbitrarily shaped rigid thin solid. The advantages of the hybridization of AMR-DAMR with IBM was also evaluated using three different test scenarios in this paper. The DAMR-IBM algorithm is advantageous because of its simplicity and lower computational cost compared with other numerical models to date. This study can be furthered to other applications that require high-resolution simulations such as flows around a thin elastic object or high-performance computing applications that are dependent on the parallel scalability (scaling efficiency) of the solver.

Conflict of interest:

None to declare.

REFERENCES

- [1] T. E. Tezduyar, S. Sathe, M. Schwaab, and B. S. Conklin, "Arterial fluid mechanics modeling with the stabilized space – time fluid – structure interaction technique," *Int. J. Numer. Methods Fluids*, no. October 2007, pp. 601–629, 2008.
- [2] H. Mo, F.-S. Lien, F. Zhang, and D. S. Cronin, "An immersed boundary method for solving compressible flow with arbitrarily irregular and moving geometry," *Int. J. Numer. Methods Fluids*, vol. 88, no. 5, pp. 239–263, 2018.
- [3] Z. Liang, L. Wei, J. Lu, and X. Qin, "Numerical simulation of a two-dimensional flapping wing in advanced mode," *J. Hydrodyn. Ser. B*, vol. 29, no. 6, pp. 1076–1080, 2017.
- [4] M. Gay and L. T. Zhang, "Numerical studies on fluid-structure interactions of stent deployment and stented arteries," *Eng. Comput.*, vol. 25, no. 1, pp. 61–72, 2009.
- [5] H.-J. Bungartz and M. Schäfer, *Fluid-structure interaction: modelling, simulation, optimisation*, vol. 53. Springer Science & Business Media, 2006.
- [6] A. man Zhang, P. nan Sun, F. ren Ming, and A. Colagrossi, "Smoothed particle hydrodynamics and its applications in fluid-structure interactions," *J. Hydrodyn.*, vol. 29, no. 2, pp. 187–216, 2017.
- [7] M. R. Rasani, M. S. Aldlemy, and Z. Harun, "Fluid-structure interaction analysis of rear spoiler vibration for energy harvesting potential," *J. Mech. Eng. Sci.*, vol. 11, no. 1, pp. 2415–2427, 2017.
- [8] C. Farhat and V. K. Lakshminarayan, "An ALE formulation of embedded boundary methods for tracking boundary layers in turbulent fluid–structure interaction problems," *J. Comput. Phys.*, vol. 263, pp. 53–70, Apr. 2014.
- [9] S. D. Costarelli, L. Garelli, M. A. Cruchaga, M. A. Storti, R. Ausensi, and S. R. Idelsohn, "An embedded strategy for the analysis of fluid structure interaction problems," *Comput. Methods Appl. Mech. Eng.*, vol. 300, pp. 106–128, 2016.
- [10] A. Gilmanov, T. B. Le, and F. Sotiropoulos, "A numerical approach for simulating fluid structure interaction of flexible thin shells undergoing arbitrarily large deformations in complex domains," *J. Comput. Phys.*, vol. 300, pp. 814–843, 2015.
- [11] F.-B. Tian, H. Dai, H. Luo, J. F. Doyle, and B. Rousseau, "Fluid--structure interaction involving large deformations: 3D simulations and applications to biological systems," *J. Comput. Phys.*, vol. 258, pp. 451–469, 2014.
- [12] B. Schott, C. Ager, and W. A. Wall, "Monolithic cut finite element based approaches for fluid-structure interaction," *arXiv Prepr. arXiv1807.11379*, 2018.
- [13] P. J. Martínez-Ferrer, L. Qian, Z. Ma, D. M. Causon, and C. G. Mingham, "An efficient finite-volume method to study the interaction of two-phase fluid flows with elastic structures," *J. Fluids Struct.*, vol. 83, pp. 54–71, 2018.
- [14] X. Lin, G. He, X. He, and Q. Wang, "Dynamic response of a semi-free flexible filament in the wake of a flapping foil," *J. Fluids Struct.*, vol. 83, pp. 40–53, 2018.
- [15] P. MING and W. ZHANG, "Numerical method for multi-body fluid interaction based on Immersed Boundary Method," *J. Hydrodyn. Ser. B*, vol. 23, no. 4, pp. 476–482, Aug. 2011.
- [16] N. Barral and F. Alauzet, "Three-dimensional CFD simulations with large displacement of the geometries using a connectivity-change moving mesh approach," *Eng. Comput.*, vol. 0, no. 0, pp. 1–26, 2018.
- [17] F. C. Yang and X. P. Chen, "Numerical simulation of two-dimensional viscous flows using combined finite element-immersed boundary method," *J. Hydrodyn.*, vol. 27, no. 5, pp. 658–667, 2015.

- [18] M. G. Gebreslassie, G. R. Tabor, and M. R. Belmont, "Numerical simulation of a new type of cross flow tidal turbine using OpenFOAM - Part II: Investigation of turbine-to-turbine interaction," *Renew. Energy*, vol. 50, pp. 1005–1013, 2013.
- [19] J. Zhong and Z. Xu, "A reduced mesh movement method based on pseudo elastic solid for fluid–structure interaction," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 0, no. 28, p. 095440621770017, 2017.
- [20] J. YANG, "Sharp interface direct forcing immersed boundary methods: A summary of some algorithms and applications," *J. Hydrodyn.*, vol. 28, no. 5, pp. 713–730, 2016.
- [21] M. S. Aldlemy, M. R. Rasani, T. M. Y. S. Tuan, and A. K. Ari, "Dynamic adaptive mesh re nement of uid-structure interaction using immersed boundary method with two-stage corrections," vol. 26, pp. 2827–2838, 2019.
- [22] R. Mittal and G. Iaccarino, "Immersed Boundary Methods," *Annu. Rev. Fluid Mech.*, vol. 37, no. 1, pp. 239–261, 2005.
- [23] J. Kim, D. Kim, and H. Choi, "An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries," *J. Comput. Phys.*, vol. 171, no. 1, pp. 132–150, 2001.
- [24] T. Kajishima and K. Taira, "Finite-difference discretization of the advection-diffusion equation," in *Computational Fluid Dynamics*, Springer, 2017, pp. 23–72.
- [25] L. Wang, G. M. D. Currao, F. Han, A. J. Neely, J. Young, and F.-B. Tian, "An immersed boundary method for fluid–structure interaction with compressible multiphase flows," *J. Comput. Phys.*, vol. 346, pp. 131–151, 2017.
- [26] B. Vadala-Roth, S. Rossi, and B. E. Griffith, "Stabilization approaches for the hyperelastic immersed boundary method for problems of large-deformation incompressible elasticity," *arXiv Prepr. arXiv1811.06620*, 2018.
- [27] C. Merlin, P. Domingo, and L. Vervisch, "Immersed boundaries in large eddy simulation of compressible flows," *Flow, Turbul. Combust.*, vol. 90, no. 1, pp. 29–68, 2013.
- [28] T. M. Nguyen, F. Proch, I. Wlokas, and A. M. Kempf, "Large eddy simulation of an internal combustion engine using an efficient immersed boundary technique," *Flow, Turbul. Combust.*, vol. 97, no. 1, pp. 191–230, 2016.
- [29] A. Cristallo and R. Verzicco, "Combined immersed boundary/large-eddy-simulations of incompressible three dimensional complex flows," *Flow, Turbul. Combust.*, vol. 77, no. 1–4, pp. 3–26, 2006.
- [30] C. S. Peskin, "Flow patterns around heart valves: A numerical method," *J. Comput. Phys.*, vol. 10, no. 2, pp. 252–271, 1972.
- [31] L. Zhu and C. S. Peskin, "Simulation of a Flapping Flexible Filament in a Flowing Soap Film by the Immersed Boundary Method," *J. Comput. Phys.*, vol. 179, pp. 452–468, 2002.
- [32] L. Zhu and C. S. Peskin, "Drag of a flexible fiber in a 2D moving viscous fluid," *Comput. Fluids*, vol. 36, no. 2, pp. 398–406, 2007.
- [33] T. Michael, J. Yang, and F. Stern, "A sharp interface approach for cavitation modeling using volume-of-fluid and ghost-fluid methods," *J. Hydrodyn. Ser. B*, vol. 29, no. 6, pp. 917–925, 2017.
- [34] M. Aldlemy, "EFFECT OF CONDUCTIVITY IN CORROSION PROBLEM USING BOUNDARY ELEMENT METHOD AND GENETIC ALGORITHM," *Knowledge-Based Eng. Sci.*, vol. 1, no. 01, pp. 58–63, 2020.
- [35] T. Kajishima, S. Takiguchi, H. Hamasaki, and Y. Miyake, "Turbulence Structure of Particle-Laden Flow in a Vertical Plane Channel Due to Vortex Shedding," *JSME Int. J. Ser. B*, vol. 44, no. 4, pp. 526–535, 2001.

- [36] W.-P. Breugem, V. van Dijk, and R. Delfos, "An Efficient Immersed Boundary Method Based on Penalized Direct Forcing for Simulating Flows Through Real Porous Media," in ASME 2012 Fluids Engineering Division Summer Meeting collocated with the ASME 2012 Heat Transfer Summer Conference and the ASME 2012 10th International Conference on Nanochannels, Microchannels, and Minichannels, 2012, pp. 1407–1416.
- [37] M. S. Aldlemy, S. A. K. A. R. A. M. A. T. Ya, and R. Alebrahim, "Composite patch reinforcement of a cracked simply-supported beam traversed by moving mass," vol. 14, no. 1, pp. 6403–6416, 2020.
- [38] T. Kempe and J. Frohlich, "An improved immersed boundary method with direct forcing for the simulation of particle laden flows," *J. Comput. Phys.*, vol. 231, no. 9, pp. 3663–3684, 2012.
- [39] M. Uhlmann, "An immersed boundary method with direct forcing for the simulation of particulate flows," *J. Comput. Phys.*, vol. 209, no. 2, pp. 448–476, 2005.
- [40] W. a Wall, a Gerstenberger, and U. M. Mayer, "Advances in Fixed-Grid Fluid Structure Interaction," *ECCOMAS Multidiscip. Jubil. Symp.*, pp. 235–249, 2009.
- [41] F. Löffler, Z. Cao, S. R. Brandt, and Z. Du, "A new parallelization scheme for adaptive mesh refinement," *J. Comput. Sci.*, vol. 16, pp. 79–88, 2016.
- [42] C. Brehm, C. Hader, and H. F. Fasel, "A locally stabilized immersed boundary method for the compressible Navier-Stokes equations," *J. Comput. Phys.*, vol. 295, pp. 475–504, 2015.
- [43] H. Ji, F. S. Lien, and F. Zhang, "A GPU-accelerated adaptive mesh refinement for immersed boundary methods," *Comput. Fluids*, vol. 118, pp. 131–147, 2015.
- [44] M. J. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics," *J. Comput. Phys.*, vol. 82, no. 1, pp. 64–84, 1989.
- [45] M. J. Berger and J. Olinger, "Adaptive mesh refinement for hyperbolic partial differential equations," *J. Comput. Phys.*, vol. 53, no. 3, pp. 484–512, 1984.
- [46] L. Billon, Y. Mesri, and E. Hachem, "Anisotropic boundary layer mesh generation for immersed complex geometries," *Eng. Comput.*, vol. 33, no. 2, pp. 249–260, 2017.
- [47] L. Wang and C. J. Wu, "An adaptive version of ghost-cell immersed boundary method for incompressible flows with complex stationary and moving boundaries," *Sci. China Physics, Mech. Astron.*, vol. 53, no. 5, pp. 923–932, 2010.
- [48] A. Pogorelov, L. Schneiders, M. Meinke, and W. Schröder, "An adaptive cartesian mesh based method to simulate turbulent flows of multiple rotating surfaces," *Flow, Turbul. Combust.*, vol. 100, no. 1, pp. 19–38, 2018.
- [49] R. S. Mohammad, M. S. Aldlemy, S. Al Hassan, A. I. Abdulla, M. Scholz, and Z. M. Yaseen, "Frictional Pressure Drop and Cost Savings for Graphene Nanoplatelets Nanofluids in Turbulent Flow Environments," pp. 1–17, 2021.
- [50] M. Vanella, A. Posa, and E. Balaras, "Adaptive Mesh Refinement for Immersed Boundary Methods," *J. Fluids Eng.*, vol. 136, no. 4, p. 040901, 2014.
- [51] S. Q. Salih et al., "Thin and sharp edges bodies-fluid interaction simulation using cut-cell immersed boundary method," *Eng. Appl. Comput. Fluid Mech.*, vol. 13, no. 1, pp. 860–877, 2019.
- [52] M. J. Berger and R. J. Leveque, "Adaptive Mesh Refinement Using Wave-Propagation Algorithms for Hyperbolic Systems," *SIAM J. Numer. Anal.*, vol. 35, no. 6, pp. 2298–2316, 1998.
- [53] S. Li and J. M. Hyman, "Adaptive Mesh Refinement for Finite Difference WENO Schemes," *Los Alamos Rep. LA-UR-03-8927*, 2003.
- [54] M. Vanella, P. Rabenold, and E. Balaras, "A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid-structure interaction problems," *J. Comput. Phys.*, vol. 229, no. 18, pp. 6427–6449, 2010.

- [55] O. A. Alawi, A. H. Abdelrazek, M. S. Aldlemy, and W. Ahmed, "Heat Transfer and Hydrodynamic Properties Using Different Metal-Oxide Nanostructures in Horizontal Concentric Annular Tube : An Optimization Study," 2021.
- [56] T. M. Y. S. T. Ya, S. Takeuchi, and T. Kajishima, "Immersed Boundary and Finite Element Methods Approach for Interaction of an Elastic Body and Fluid by Two-Stage Correction of Velocity and Pressure," in ASME/JSME 2007 5th Joint Fluids Engineering Conference, 2007, pp. 75–81.
- [57] M. S. Aldlemy, M. R. Rasani, A. K. Ariffin, and T. M. Y. S. T. Ya, "Adaptive mesh refinement immersed boundary method for simulations of laminar flows past a moving thin elastic structure *," vol. 32, no. 1, pp. 148–160, 2020.
- [58] Y. Yuki, S. Takeuchi, and T. Kajishima, "Efficient immersed boundary method for strong interaction problem of arbitrary shape object with the self-induced flow," *J. Fluid Sci. Technol.*, vol. 2, no. 1, pp. 1–11, 2007.
- [59] O. Sabir and T. T. Ya, "Numerical Simulation of Fluid Flow in Three Dimensional Domain Based on Two Stage Pressure-Velocity Correction Method," in ASME 2015 International Mechanical Engineering Congress and Exposition, 2015, p. V07AT09A017--V07AT09A017.
- [60] T. M. Y. S. Tuanya, S. Takeuchi, T. Kajishima, and A. Ueyama, "Technical note: Immersed boundary method (body force) for flow around thin bodies with sharp edges," *Int. J. Mech. Mater. Eng.*, vol. 4, no. 1, pp. 98–102, 2009.
- [61] M. Schäfer and S. Turek, "Benchmark computations of laminar flow around a cylinder," *Flow Simul. with High-Performance Comput. II, Vol. 52 Notes Numer. Fluid Mech. Vieweg*, vol. 52, pp. 547 – 566, 1996.
- [62] A. C. Verkaik, M. A. Hulsen, A. C. B. Bogaerds, and F. N. van de Vosse, "An overlapping domain technique coupling spectral and finite elements for fluid flow," *Comput. Fluids*, vol. 100, pp. 336–346, Sep. 2014.