

# Flow Around a Plunging Airfoil: A Comparative Study of Immersed Boundary and Arbitrary Lagrangian-Eulerian (ALE) Methods in OpenFOAM

Manjil Sitoula<sup>1</sup> and Chandan Bose<sup>2</sup>

<sup>1</sup>Bachelors in Aerospace Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University

<sup>2</sup>Assistant Professor, Aerospace Engineering, College of Engineering and Physical Sciences, The University of Birmingham

September 24, 2024

## Synopsis

The present study is to apply the Immersed Boundary Method (IBM) on a plunging airfoil using FOAM-Extend, a fork of the OpenFOAM open-source library and compare the aerodynamic load coefficients with the Arbitrary Lagrangian-Eulerian Method (ALE). The main interest of this study is to observe the significance of this method in capturing the moving boundary accurately since this method works on an algorithm of momentum forcing and interpolations, unlike mesh deformation in ALE. The 2D plunging airfoil is simulated with a flow of Reynolds number=100 in IBM, and the results are compared with those of the body-fitted mesh. The reduced frequency( $k$ ) of 4Hz and amplitude( $h$ ) of 0.25m is taken for the plunging motion. FOAM-Extend offers immersed boundary solvers to handle the flow problem uniquely. In this paper, the results show good agreement between these methods and justify the accurate prediction of the dynamic motion by the Immersed Boundary Method. However, a detailed study of the immersed boundary cells and time step size is essential for the accuracy of this method.

## 1 Introduction

The Immersed Boundary method (IBM) is a computational technique where a rigid boundary is immersed in a fixed background Cartesian mesh. This technique was introduced by Peskin for the simulation of the blood flow patterns in the heart [1]. A numerical technique was formulated to solve the Navier-Stokes equation in the presence of a immersed boundary. A fast Laplace Solver

could be used due to the representation of the boundary as a field of forces defined on the Cartesian background mesh [1]. This approach has developed into an effective technique for addressing fluid-structure interaction problems. IBM uses Eulerian variables formulated on a Cartesian mesh, which is fixed and Lagrangian variables on a curvilinear mesh, which are able to move freely across the Cartesian mesh. The interaction equations relates these two variables involving a major role of the Dirac delta function [2].

IBM is a simple and useful method for mesh generation in CFD where a non-conformal boundary surface presence is accounted for the Cartesian Mesh through the boundary forces, interpolations and modifying the governing equation in the cells near the immersed boundary. The forces are imposed to mimic the effect of no-slip conditions at the boundary. The use of non-deformable grids cuts down the computational effort and makes the method relatively simple. [3].

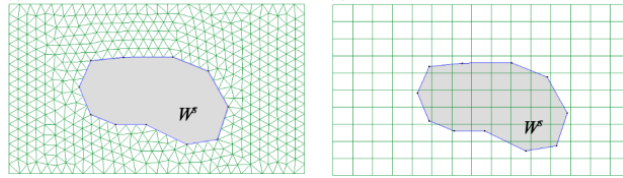


Figure 1: ALE(left) vs IBM(right) mesh approach [3]

IBM provides an advantage in handling complex geometries and moving boundaries in terms of mesh generation and computational cost. Since mesh deformation does not occur, the issues related to the highly deformed fluid grid for large displacement do not impose a problem in IBM, as in the ALE approach. The effect of boundary in IBM is transferred to the Cartesian mesh through interpolation functions, without a real interface existing between the two. [4]. This leads to a less precise description of the fluid-structure interface, so refinement of mesh locally around the moving boundary can be adopted to enhance the quality of surface definition in the fluid domain.

The Immersed Boundary Method is integrated in FOAM-Extend by Jasak [5]. The immersed boundary is treated using different approaches and has been modified to date for better accuracy. In indirect forcing, the forcing term is included in the immersed cells after the equations are discretized, and these forces are spread over to a band of cells in the Cartesian Mesh whereas in direct forcing, the discretized equations close to the immersed boundary are modified using Dirichlet or Neumann B.C. to apply the boundary condition directly to the Immersed Boundary (IB) cells. The dependent variables are then interpolated using the neighbouring cell values and values at the boundary point. Foam extend 4.0 and below adopts the direct forcing method and this has now been modified in version 4.1. With the previous algorithms, problems such as the loss of information in cut cells, precision loss at the boundary, and inaccurate evaluation of forces at the IB surface prevailed, degrading the solution. So, in the Foam extend 4.1 and above, the Immersed Boundary Surface(IBS) algorithm is implemented [5]. The Immersed Boundary is included within the Cartesian Mesh as in the case of body-fitted mesh. In the cut-cell approach, the cells are reshaped to accommodate the boundary instead of interpolating the velocities from the surface to the IB cells [6]. The interpolation methods and procedures used in IBM are presented in [7] [8].

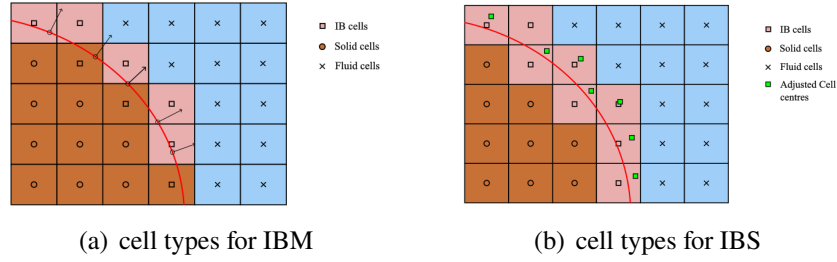


Figure 2: Comparison of old vs new IB algorithm

In IBM, the cells can be classified into IB, Solid and Fluid cells. IB cells are the cells intersected by the immersed surface, with their cell centre located within the Fluid region. If cell centre lies in the solid region, it is classified into solid cells. This definition was implemented in older versions. In IBS implementation, all the intersected cells by the IB are categorized into IB cells. The IB cells can be further categorized into live and dead surfaces. The intersected cells whose centre lies in the solid region and adjacent to fluid cells are considered live cells, and those adjacent to solid are dead parts. These live cells make up a new IB cell. All dead cells and faces are excluded from the discretization matrix [5] [9].

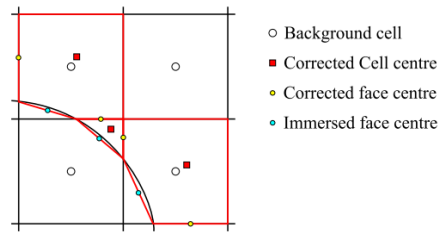


Figure 3: Corrected cell centres and face centres in cut cell [9]

The boundary faces are considered as new IB faces, and the IBS cell centre is adjusted based on the living IB cells. In the classical cut-cell method, the cut cells whose centre lies in the solid region are absorbed by the neighbouring fluid cell, whereas in IBS, they are considered a single IB cell, and the topology of the mesh in IBS remains unchanged. IBS algorithm improves the accuracy and stability of IBM by allowing a conventional body-fitted FVM discretization instead of forcing and interpolating values from the boundary [9].

## 2 Governing Equations and Models

### 2.1 Problem definition

This study aims to model the flow around a plunging airfoil using IBM and ALE methods. The IBM approach is examined in foam extend 4.1, and ALE is in standard OpenFOAM version 2312. This study aims to compare these approaches by analyzing the aerodynamic load coefficients for the Reynolds number flow of 100.

## 2.2 Governing equations

The continuity and momentum conservation equations are solved in the Cartesian Mesh for incompressible Laminar flow.

**Continuity Equation:**

$$\nabla \cdot \mathbf{u} = 0$$

**Momentum Equation:**

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} (\nabla \cdot p) + \nu \nabla^2 \mathbf{u}$$

where  $\nu$  is the kinematic viscosity,  $\rho$  is the density,  $\mathbf{u}$  is the velocity vector and  $p$  is the pressure. The N-S equations and numerical discretization in foam extend 4.1, where IBS is introduced, are similar to the ALE method.

PimpleFoam, a pressure-based transient, incompressible solver is selected for this study. For dynamic motion, a pimpleDyMFoam solver is used for IBM.

## 2.3 Geometry and Mesh

A rectangular domain with a length of 30 metres in the x-direction and 10 metres in the y-direction is set up. The flow is simulated around the NACA 0012 airfoil, and the leading edge of the airfoil lies at the origin(0,0,0). The airfoil is symmetrical and has a chord length of 1 m. The schematic diagram of the computational domain is shown below:

The computational domains for IBM and ALE are the same.

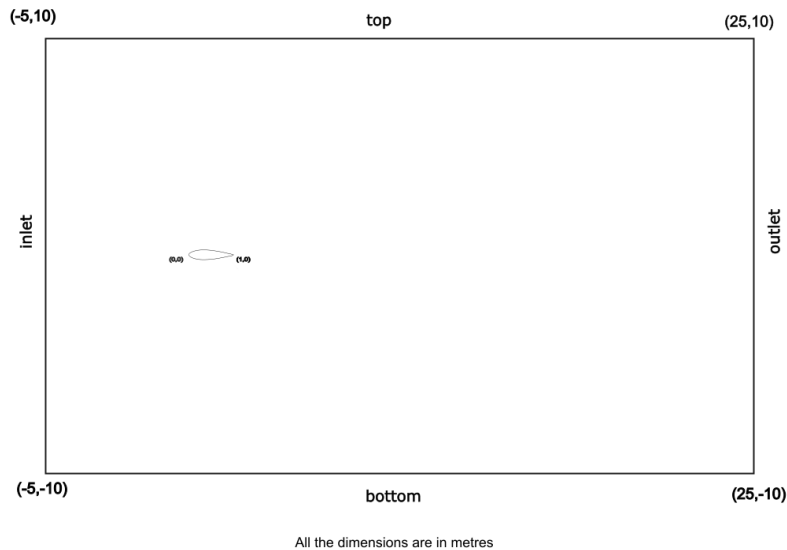


Figure 4: Computational Domain

The meshing procedure is quite different in IBM. For IBM, meshing is done using the blockMesh utility for the background Cartesian Mesh. The immersed surface geometry is imported in STL format and immersedBoundaryPatch is included in the boundary file. The surface does not cut the

background mesh and is treated separately. The resolution of the IB surface is not important [5], but the refinement of cells near the boundary is prior important. The coarseness of Mesh is one of the limitations of the IBS method. The cartesian mesh size near the boundary is 10mm in the y direction and 15mm in the x direction, respectively. The selected mesh after the grid convergence study is shown below:

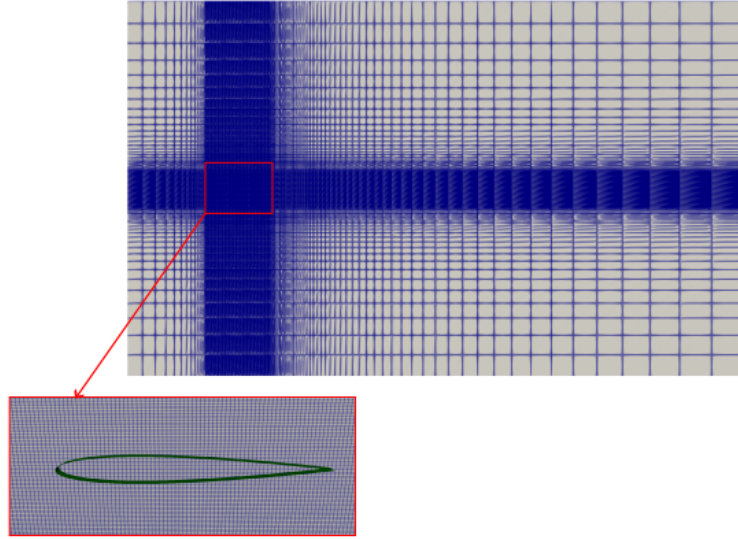


Figure 5: Mesh Generation in IBM

The meshing generation in ALE is done using the open-source meshing tool Gmsh. The edge grading is applied to make finer mesh near the airfoil and smooth the transition between the mesh. Since the flow is laminar with a low Re number, the y-plus value was not examined. The mesh size in the y direction adjacent to the boundary is 40mm, which is larger than that of IBM. The ALE mesh is shown below:

## 2.4 Solver Setup

### 2.4.0.1 Fluid Properties

An incompressible Newtonian Fluid is used to simulate the flow around the airfoil. To maintain a Reynolds number of 100, the kinematic viscosity is set to  $0.01 \text{ m}^2/\text{s}$ .

$$Re = \frac{Uc}{\nu}$$

where U is the inlet velocity set to  $1 \text{ m/s}$  and c is the chord length.

### 2.4.0.2 Initial and Boundary Conditions

The boundary conditions for both ALE and IBM methods are kept the same in order to make comparisons. For the plunging airfoil, the boundary conditions in IBM are:

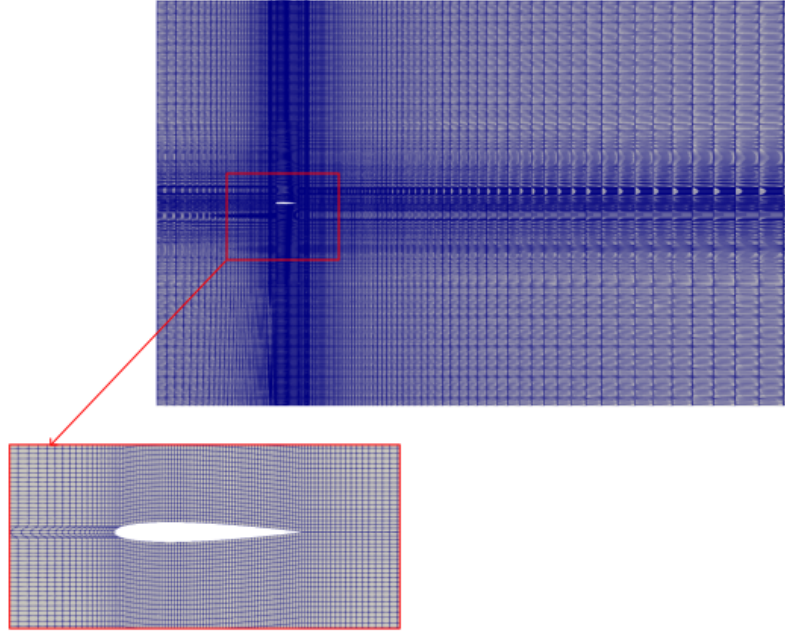


Figure 6: Mesh Generation in ALE

Flow variable	Value
$U$	$0 \text{ m/s}$
$p$	$0 \text{ m}^2/\text{s}^2$

Table 1: Initial Conditions

Patch	Condition	Value( $\text{m}^2/\text{s}^2$ )
inlet	zeroGradient	-
outlet	fixedValue	0
top bottom	fixedValue	0
frontAndBack	empty	-
naca_0012	zeroGradientIb	-

Table 2: Boundary Conditions for  $p$ 

Patch	Condition	Value( $\text{m/s}$ )
inlet	fixedValue	1
outlet	inletOutlet	1
top bottom	fixedValue	0
frontAndBack	empty	-
naca_0012	movingImmersedBoundaryVelocity	-

Table 3: Boundary Conditions for  $U$ 

In ALE, the initial and boundary conditions are the same as above. Instead of movingImmersed-

BoundaryVelocity in naca\_0012, movingWallVelocity is used in ALE. This both represents the same condition and ensures no slip at the wall. This mesh motion and velocity information is provided in the dynamicMeshDict and treated accordingly.

### 2.4.0.3 Dynamic Mesh Motion

The motion of the plunging airfoil is governed by its amplitude( $h$ ) and reduced frequency( $k$ ). The oscillating amplitude is chosen as  $0.25m$ , and the reduced frequency is taken as 4 such that their product is unity. This results in an angular velocity of 4 rad/s and a time period of 1.57 s.

$$k = \frac{\omega * c}{U}$$

where  $k$  is the reduced frequency,  $\omega$  is the angular velocity,  $c$  is the chord length of airfoil and  $U$  is the inlet velocity.

For the IBM approach, the plunging airfoil is solved using the pimpleDyMFoam solver. Since this does not involve mesh deformation, the solver treats the motion differently. The mesh motion parameters are configured in the dynamic Mesh dictionary. The immersedBoundarySolidBody-MotionFvMesh is used for dynamic Mesh type and the linearOscillation motion function is used to prescribe the mesh motion. The oscillation amplitude and period are set up for the plunging airfoil. Similarly, in ALE, the motion parameters used are the same. The displacement Laplacian solver is set in the dynamicMeshDict with a diffusivity coefficient assigned as InverseDistance 1. The airfoil's motion is defined in the pointDisplacement file with the required amplitude and angular velocity. This solver handles mesh movement and deformations according to the airfoil's plunging motion.

### 2.4.0.4 Solution Method and Control

The numerical Discretization schemes used for both methods are alike. The Laminar model is used to capture the flow phenomenon since the Reynolds number is low. The numerical schemes used in IBM are:

Discretization	Scheme
Temporal	Euler
Gradient	cellLimited Gauss Linear 1
Divergence	Gauss Upwind
Laplacian	Gauss Linear Limited 0.5
Interpolation	Linear

Table 4: Numerical Schemes in IBM

The cellLimited Gauss Linear 1 is used to ensure stability and boundedness of the solution. This was also changed to Gauss Linear but the results were similar. Similarly, for ALE, the numerical schemes used are:

Discretization	Scheme
Temporal	Euler
Gradient	cellLimited Gauss Linear 1
Divergence	Gauss Upwind
Laplacian	Gauss Linear corrected
Interpolation	Linear

Table 5: Numerical Schemes in ALE

Gauss linear corrected is used in LaplacianSchemes for better accuracy. Setting it to Gauss Linear limited 0.5 resulted in a large peak in observed values. Other numerical schemes are the same as those used in IBM.

Both methods use the PIMPLE algorithm solved using the pimpleFoam solver. For IBM, solvers for pressure and velocity fields are assigned with CG (Conjugate Gradient) and BiCGStab (Bi-Conjugate Gradient Stabilized), respectively, and preconditioners are applied. Similarly, for ALE, pressure is solved using a GAMG solver and a smoothSolver for the velocity field. Under relaxation is also used for better convergence.

## 3 Results and Discussions

### 3.1 Convergence Tests

#### 3.1.1 Grid Size Convergence Test

A grid independence study is significant in evaluating the impact of mesh refinement on the accuracy and stability of the solution. A finer grid enhances accuracy but at the cost of increased computational time. So, mesh sensitivity analysis is necessary in CFD to maintain a balance between accuracy and computational time. Three different grid sizing are taken for IBM, and the coefficient of drag( $C_d$ ) at the airfoil is selected as the convergence test parameter. The grid refinement ratio ( $r$ ) is approximately set to 2. These  $C_d$  values are the average values from the 1<sup>st</sup> cycle to the 3<sup>rd</sup> oscillation cycle. The mesh independence study for IBM is given in the table below:

Grid level	Number of Elements	Coefficient of Drag( $C_d$ )	Error(%)
M1	36432	0.3178	11.4
M2	72800	0.2941	3.08
M3	145332	0.2853	-

Table 6: Grid Independence Study for IBM

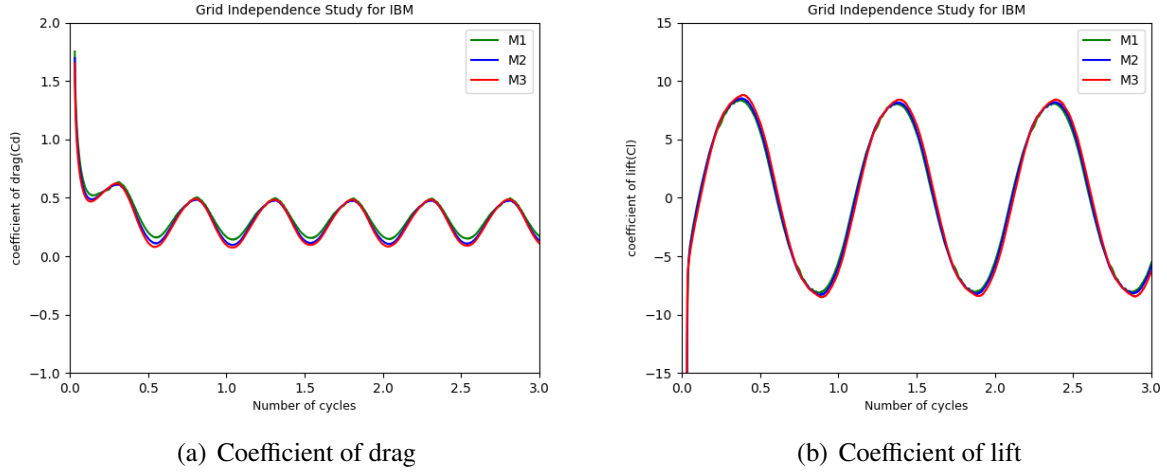


Figure 7: Grid Independence Study for IBM

From the data above, since the error decreases with mesh refinement, M2 mesh was selected for the study. The error is calculated from the finest mesh. Due to high fluctuations observed in the values, the moving averages were taken to smooth the graphs.

Similarly, for ALE, a separate mesh independence study was conducted with three different grid sizes. The grid refinement ratio ( $r$ ) is approximately set to 2.

Grid level	Number of Elements	Coefficient of Drag( $C_d$ )	Error(%)
M1	25920	0.248	17.77
M2	52470	0.228	8.51
M3	105161	0.21	-

Table 7: Grid Independence Study for ALE.

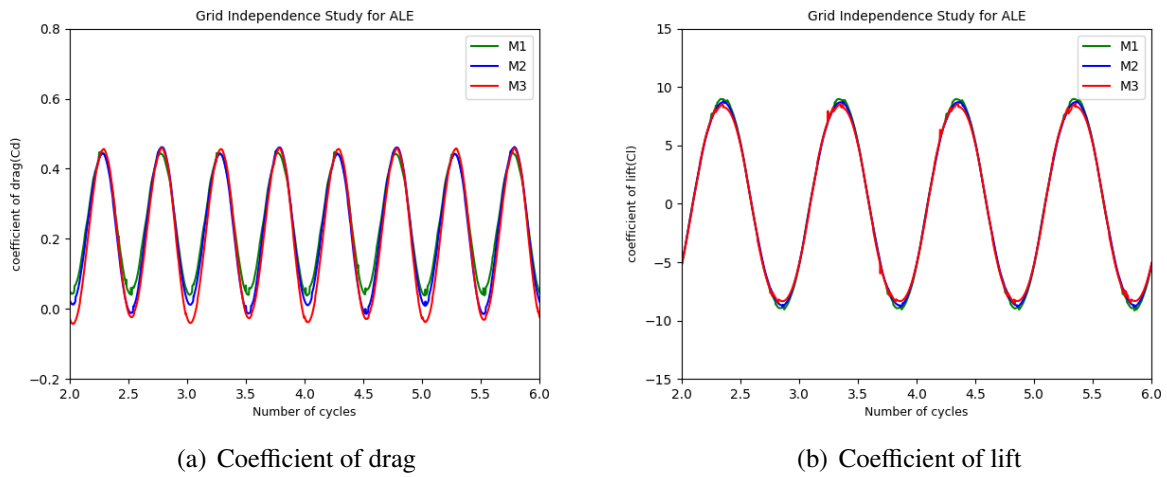
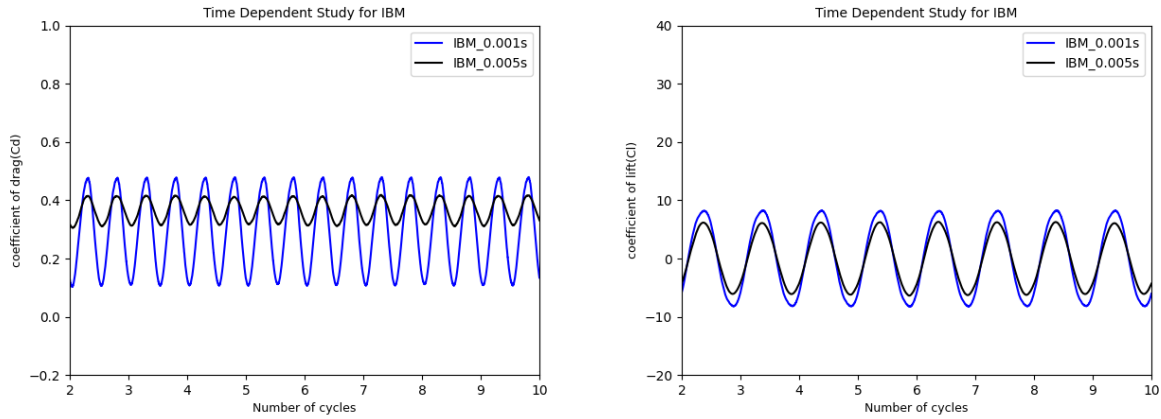


Figure 8: Grid Independence Study for ALE

The peak and trough values quite resemble the finest mesh, and the error percentage also shows a decreasing trend. M2 mesh was selected for the study and comparison. M2 mesh has an element size of 18mm and 40mm adjacent to the boundary in the x and y directions, respectively.

### 3.1.2 Time Step Size

Time Step plays an important role in dynamic mesh simulation. A small time step is required for solution stability and convergence, keeping the Courant Number within the desired value. Moreover, time step size also has a prominent effect on the accuracy of the solution, especially in IBM, and their effective study is required.



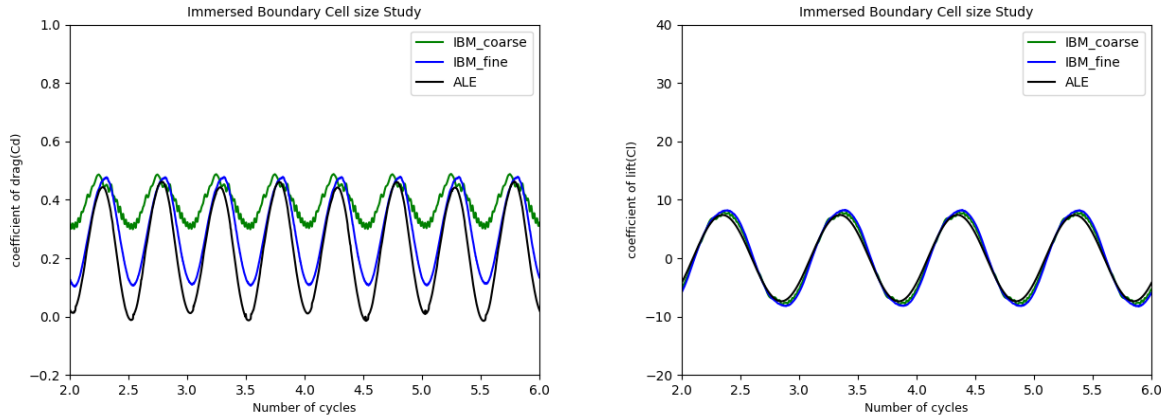
(a) Coefficient of Drag with a time Step size of 0.001s and 0.005s (b) Coefficient of Lift with a time Step size of 0.001s and 0.005s

Figure 9: Time Step Size Study for IBM

For the same mesh, the solution is carried out using two-time step sizes of 0.001s and 0.005s. Even though the Cl values are similar, Cd is significantly different. Time step immensely affects the accuracy of the solution even with minor differences. From the above plots, a small time step is able to produce a closer result towards ALE. Further lowering the time step increases the accuracy, but the computational time also spikes. Therefore, IBM needs to conduct a systematic time independence study. From the above graphs, a time step of 0.001s is chosen for the accuracy of the solution in IBM.

### 3.1.3 Immersed Boundary Mesh Refinement

The mesh size near the boundary cells or the size of immersed boundary cells creates drastic alterations in the values of aerodynamic coefficients. The mesh coarseness is also one of the limitations of IBS, and a fine mesh is required for accuracy.



(a) Coefficient of drag for different mesh size near the boundary  
(b) Coefficient of lift for different mesh size near the boundary

Figure 10: Boundary mesh size study

IBM\_coarse in the graph signifies mesh with coarser mesh near the immersed boundary (i.e. 28.5mm in x and y direction) with 79968 no of elements whereas IBM\_fine signifies refined immersed Boundary Mesh with an element size of 15mm and 10mm in x and y direction respectively and with 72800 number of cells. Even though the number of elements is similar for both these meshes, the mesh size near the boundary is different, which drastically changes the results, which can be observed in figure 10. Time Step size of 0.001s is taken for the simulation. Therefore, a boundary mesh size independent study is also required in IBM, and fine mesh near the boundary is essential for the accurate calculation of the aerodynamic coefficients, especially the coefficient of drag. IBM also has an inbuilt command for the refinement of boundary cells (IB cells).

Various considerations are required to accurately predict IBM's results. Mesh size and time step size are crucial factors in determining IBM's accuracy.

### 3.2 Results

The aerodynamic load coefficients for the plunging airfoil at Re 100 are compared between IBM and ALE. The coefficient values are obtained using inbuilt force functions. The flow time was 30 seconds. Since the periodic behaviour of Cd and Cl repeats after each cycle and the average values are constant, the simulation time was not increased further. The periodic nature of drag is due to the vortex-shedding phenomenon caused by the plunging motion. The plunging motion causes the change in the airfoil's angle of attack, thereby changing the aerodynamic coefficients.

From the above study, final comparisons are done with a mesh size of 10mm near the boundary and a time step of 0.001s for IBM. In the case of ALE, a time step of 0.005s is taken. Since the flow is laminar, no special considerations are made in first cell heights in ALE. The Cd is slightly different between IBM and ALE as shown in figure 11(a). Although the peak value of Cd in both methods is the same and in phase with each other, the trough differs. The Cl, on the other hand, exhibits a closer resemblance in these methods, as can be seen in figure 11(b). The average values of Cl and Cd are calculated from 10<sup>th</sup> to 18<sup>th</sup> cycle.

Parameters	IBM	ALE
Cd	0.2966	0.2275
Cl	0.002	0.01

Table 8: Plunging Airfoil

Table 9: Aerodynamic Load coefficients

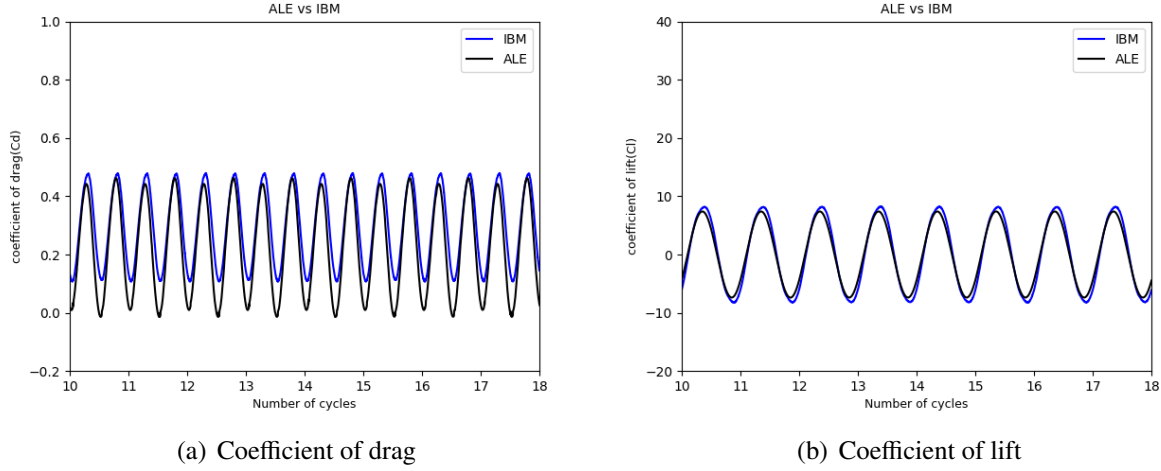


Figure 11: Comparison of aerodynamic load coefficients between IBM and ALE for the plunging airfoil

The average value of the aerodynamic coefficients is provided in table 8. As can be observed from this data, IBM closely predicts the Cd and Cl values with minor deviations.

Vortex Shedding is an unsteady flow phenomenon in an incompressible low Re flow. These vortices affect the aerodynamic characteristics and are also called Von Karman vortex rings. The Von Karman vortices resulting from the plunging motion are shown below and compared for ALE and IBM. In the IBM method, more number vortices are visualized at the same time of 15.8 s (i.e. 10<sup>th</sup> cycle) than ALE, which can be seen in the figure 12.

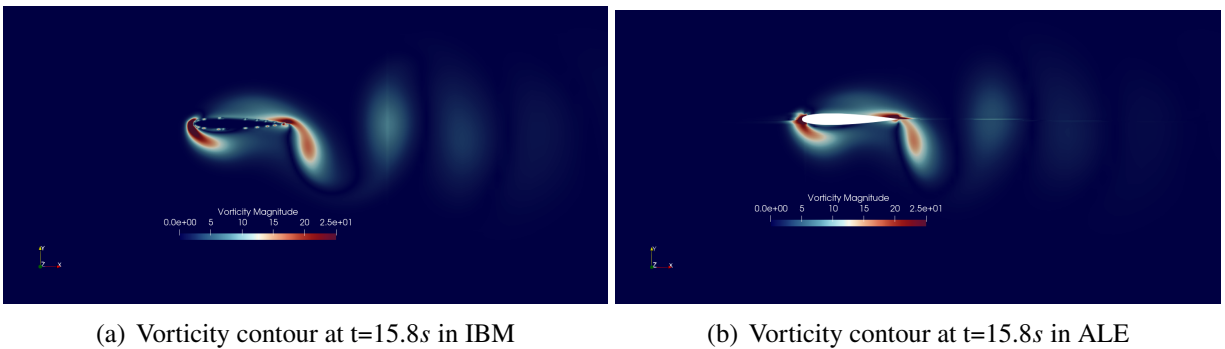
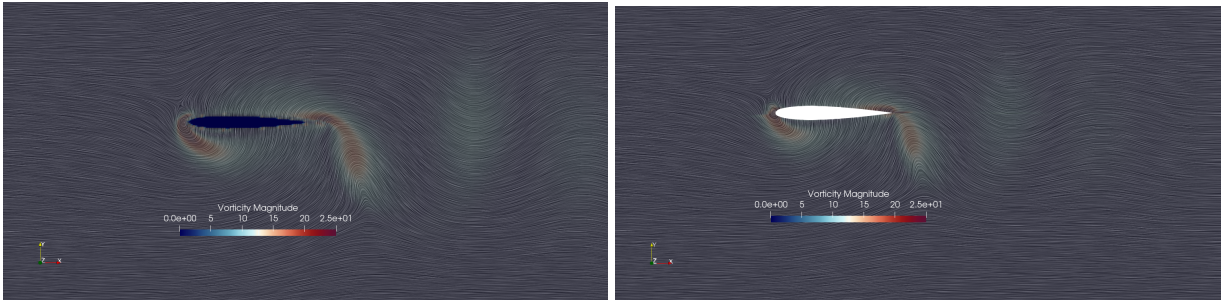


Figure 12: Comparison of Vorticity contour for plunging airfoil

The SurfaceLic representation of the vorticity is illustrated in figure 13. The streamlines can be visualized in the vortex region where the velocity curls and vorticity are generated.

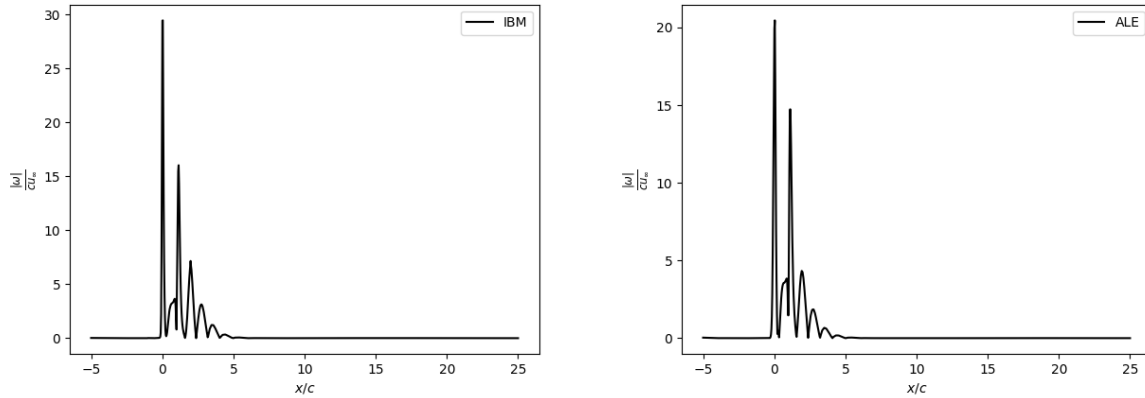


(a) SurfaceLic representation at  $t=15.8s$  in IBM

(b) SurfaceLic representation at  $t=15.8s$  in IBM

Figure 13: Comparison of Vorticity LIC for plunging airfoil

At time= $15.8s$ , the vorticity magnitude is compared between the IBM and ALE methods. It is measured along a horizontal line. The magnitude of the vorticity is plotted and compared in figure 14. The vortices are formed at the leading edge and the trailing edge, where the peak magnitude of vorticity is obtained. The vortex then sheds into the wake region, creating a pattern of alternating swirling motions. The magnitude of the vorticity at the leading edge, trailing edge and other points along the line in the domain are alike in both these methods, which is illustrated in figure 14. At the leading edge, the vorticity magnitude is higher in IBM but is similar elsewhere. So, the vortex formation and the magnitudes captured in IBM are similar to those captured in ALE.



(a) Vorticity Magnitude at  $t=15.8s$  in IBM

(b) Vorticity Magnitude at  $t=15.8s$  in IBM

Figure 14: Comparison of Vorticity magnitude for plunging airfoil

## 4 Conclusion

This study provides a comparison and analysis of the aerodynamics of a plunging airfoil using latest IBS implementation and ALE methods. For the plunging airfoil, the aerodynamic load coefficients

are similar in both IBM and ALE. For the coefficient of drag, time step size and cell size near the immersed boundary play the utmost role in accuracy. However,  $Cl$  is not affected by these parameters and its value is similar to that of ALE. Therefore, extensive analysis is required for the correct prediction of drag in IBM, which can also be a limitation since  $Cl$ , on the other hand, is like ALE, irrespective of other parameters. The vortex formation and magnitude of the vorticity are also compared, which agrees with both of these methods. The leading edge vorticity magnitude is a little higher in IBM but is similar in other sections of the domain.

This study intends to compare these methods and IBM's accuracy with ALE. IBM makes mesh generation easy without mesh deformation for dynamic motion. This can be a powerful method to simulate moving boundaries, but necessary considerations are required for the solution's accuracy. Various modifications in the algorithms have been made for better accuracy. This study is a simple comparison with low  $Re$  flow, and in conclusion, the immersed boundary method is effective in capturing the unsteady flow behaviour of the plunging airfoil.

## 5 Acknowledgement

I would like to extend my heartfelt gratitude to Prof. Chandan Bose for his unwavering support and guidance throughout this project. The weekly discussions and his recommendations were very insightful, broadening my understanding and enriching my knowledge of CFD. I would also like to thank Mr. Biraj Khadka for assisting me with the challenges encountered during the project.

I am also grateful to Ms Payel Mukherjee for her assistance and support during my physical internship at IIT Bombay. I would also like to extend my sincere thanks to Prof. Amol Subhedar and Mr. Malyadeep Bhattacharya for the lab facilities and guidance at IIT Bombay. Lastly, I am thankful to the FOSSEE team for their support and for providing me with this internship opportunity.

## References

- [1] C. S. Peskin, "Numerical analysis of blood flow in the heart," *Journal of computational physics*, vol. 25, no. 3, pp. 220–252, 1977.
- [2] C. Peskin, "The immersed boundary method," *Acta numerica*, vol. 11, pp. 479–517, 2002.
- [3] A. Ballatore, "Immersed boundary method in openfoam: numerical validation and application to wheel geometries," 2019.
- [4] A. M. Bavo, G. Rocatello, F. Iannaccone, J. Degroote, J. Vierendeels, and P. Segers, "Fluid-structure interaction simulation of prosthetic aortic valves: comparison between immersed boundary and arbitrary lagrangian-eulerian techniques for the mesh representation," *PloS one*, vol. 11, no. 4, p. e0154517, 2016.
- [5] H. Jasak, "Immersed boundary surface method in foam-extend," in *Workshop OpenFOAM in Hydraulic Engineering*, vol. 21, 2018, p. 22.

- [6] T. Ye, R. Mittal, H. Udaykumar, and W. Shyy, “An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries,” *Journal of computational physics*, vol. 156, no. 2, pp. 209–240, 1999.
- [7] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, “Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations,” *Journal of computational physics*, vol. 161, no. 1, pp. 35–60, 2000.
- [8] T. Kajishima, K. Taira, T. Kajishima, and K. Taira, “Immersed boundary methods,” *Computational Fluid Dynamics: Incompressible Turbulent Flows*, pp. 179–205, 2017.
- [9] J. E. Döhler, “An analysis of the immersed boundary surface method in foam-extend,” 2022.