FOSSEE, IIT Bombay
OpenFOAM Research Migration Project
August 2024

Flow Around a Plunging Airfoil: A Study Using the New Immersed Boundary Method Algorithm in foam-extend 4.1

Sugam Lamsal[1], Manjil Sitoula[2], and Chandan Bose[3]

[1]Undergraduate Student, Aerospace Engineering, IOE Pulchowk Campus, Tribhuvan University
[2]Aerospace Engineering, IOE Pulchowk Campus, Tribhuvan University
[3]Assistant Professor, Aerospace Engineering, College of Engineering and Physical Sciences, The University of Birmingham

August 23, 2025

Synopsis

This study reports a numerical analysis of unsteady flow over a plunging elliptical airfoil utilizing Immersed Boundary Surface Method (IBS) implemented FOAM-Extend 4.1 framework. The aim of this work is to validate force coefficient results using the recent discrete-forcing IB algorithm from a published study which used the continuous-forcing IB algorithm. The motivation comes from the necessity of precise flow modeling around oscillating airfoils in engineering applications like flapping-wing aerodynamics and bioinspired propulsion. Compared to its previous version, the more recent IB implementation is intended to enforce boundary conditions more precisely like the body-fitted method, particularly when there are moving boundaries. Simulations are performed at a Reynolds number $Re = 500$ with a constant reduced frequency $\kappa = 6.283$, while two distinct values of κh (0.31 and 1.0) are considered to explore different oscillation amplitudes. The airfoil is subjected to a vertical plunging motion defined by a sinusoidal profile. The study focuses on evaluating the lift and thrust coefficients obtained from the new IB implementation and comparing it with the literature. The result shows a strong agreement in force predictions, validating the accuracy and robustness of the IBS formulation. This study serves as a benchmark for upcoming enhancements and three-dimensional extensions and demonstrates the capacity of FOAM-Extend's more recent IB algorithm to simulate changing boundary flows.

1 Introduction

The numerical simulation of unsteady flows around dynamic or deformable barriers is fundamental for modern computational fluid dynamics (CFD), with applications including bio-memetic propulsion, flapping wing micro-aerial vehicles, energy harvesting, and turbulence control. Traditional body-fitted mesh approaches, while accurate, require dynamic mesh deformation techniques for moving boundaries, making them computationally intensive and prone to numerical instability for large or frequent deformations [1]. The Immersed Boundary Method (IBM), first put forward by Peskin (1972), has emerged as a robust technique for simulating flow around complex, dynamic geometries using fixed Cartesian grids, thereby eliminating the necessity for dynamic meshing [2]. In IB methods, the presence of solid boundaries is imposed indirectly by modifying the governing equations using additional source terms or by interpolating boundary conditions from nearby fluid nodes [3].

IBM can be classified into two primary algorithmic methodologies: direct forcing and indirect forcing. In the indirect forcing method, body forces are incorporated into the momentum equations post-discretization, and the effect of the immersed surface is distributed throughout a band of cells. The direct forcing approach imposes boundary conditions by directly altering the discretized equations adjacent to the immersed boundary using Dirichlet or Neumann conditions [4]. The previous IBM algorithms had significant limitations, including precision issues, challenges in evaluating accurate surface forces, loss of information in cut cells, and inaccurate interpolation near the immersed surface [5].

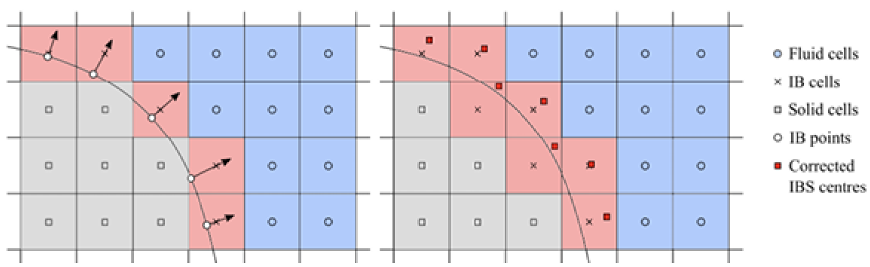


Figure 1: Cell types for IBM in a) Foam-Extend 4.0 and b) Foam-Extend 4.1 [6]

The Immersed Boundary Surface (IBS) method was implemented in foam-extend 4.1 to mitigate the drawbacks of traditional IBM [7]. Unlinke previous algorithms that depended on interpolation from immersed surface to IB cells, the IBS method modifies the cells intersected by the immersed boundary to more accurately align with the geometry, similar to a cut-cell or body-fitted approach. All intersected cells are categorized as IB cells and are further categorized into live and dead IB cells. The intersected cells whose center lies in the solid region and adjacent to fluid cells are considered live cells, and those adjacent to solid are dead cells. These live cells form a new IB cell. All dead cells and faces are excluded from the discretization matrix [6] [7].

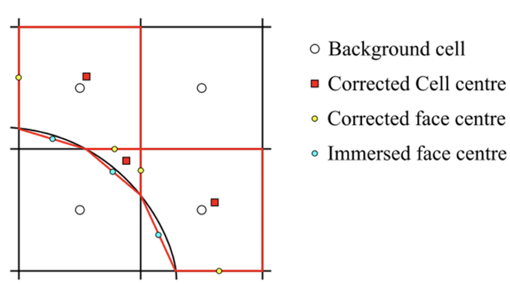


Figure 2: Corrected cell centres and face centres in cut cell [6]

The surfaces at the boundaries are treated as new IB faces, while the cell center of the IBS is modified according to the living IB cells. In the classical cut-cell method, the cut cells whose center lies in the solid region are absorbed by the neighboring 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.

A number of studies have explored the aerodynamics of plunging airfoils using various IBM implementations. For instance, [8] used the old IBM in FOAM-Extend to compute vortex-induced forces on an elliptical airfoil undergoing sinusoidal plunging motion at moderate Reynolds numbers (i.e., $Re = 500$). While their results capture expected physical trends, the older IBM may have introduced diffused boundary effects, particularly at higher amplitudes. Meanwhile, the newer IBM in FOAM-Extend now includes sharper enforcement and better local velocity reconstruction [7].

2 Governing Equations and Models

2.1 Problem Definition

This study is focused on modeling the flow around a plunging airfoil using the Immersed Boundary Surface Method implemented in foam-extend 4.1. The primary objective is to validate the force coefficient results obtained from the new IBS algorithm by comparing it with the published data from the older IBM implementation. The simulation is performed at a Reynolds number of $Re =$ 500, with the airfoil undergoing harmonic plunging motion defined by a fixed reduced frequency of $\kappa = 6.283$ and two distinct amplitudes: $\kappa h = 0.31$ and $\kappa h = 1.0$. The aim is to assess the accuracy and performance of the IBS algorithm in modeling unsteady aerodynamic loads.

2.2 Governing Equations

The dynamics of the flow are governed by the incompressible Navier–Stokes equations, which include the continuity equation for mass conservation and the equations for momentum.

Continuity Equation

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

This ensures the conservation of mass in an incompressible flow.

Momentum Equation

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

where:

- $\mathbf{u}$ is the velocity vector,
- p is the pressure field,

- ρ is the fluid density,
- ν is the kinematic viscosity.

The equations are discretized and solved with the finite volume approach in foam-extend. The immersed boundary method (IBM) is applied to incorporate complex geometries into a fixed Cartesian mesh, with further improvements in the Immersed Boundary Surface (IBS) algorithm [9].

3 Geometry and Mesh

The computational domain is a rectangle with dimensions of 35 m in length and 20 m in height. The leading edge is positioned at the origin (0, 0), situated 10 m downstream from the inlet boundary and symmetrically located between the upper and lower walls of the area. The airfoil used in the simulation is an elliptical airfoil with a chord length of 1 m.

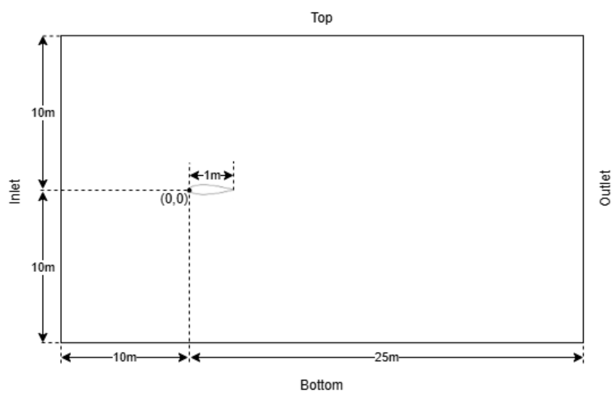


Figure 3: Computational Domain

In IBM, a uniform cartesian mesh is generated over the entire domain without conforming to the exact shape of the geometry. The airfoil is imposed into the mesh using the IBS algorithm introduced in foam-extend 4.1 [7]. The STL format of the airfoil is embedded into the mesh, and cells intersected by the surface are classified as IB cells. The IBS method reconstructs the immersed geometry within the cut cells and implements boundary conditions directly through a revised discretization technique, avoiding the interpolation-based forcing methods that were utilized in previous IBM versions [7].

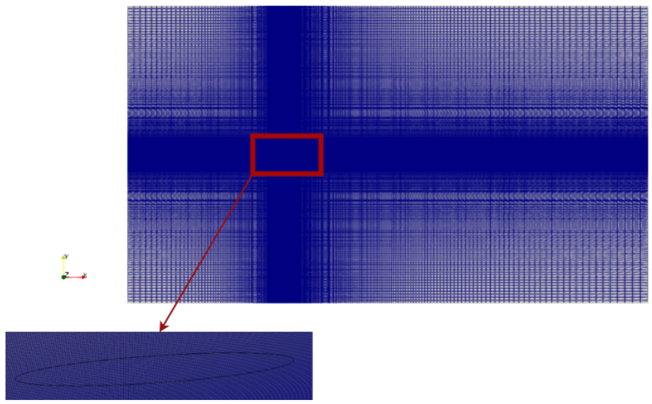


Figure 4: Mesh Generation

3.1 Solver Setup

3.1.1 Fluid Properties

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

$$Re = \frac{U \cdot L}{\nu} \tag{3}$$

where:

- U is the characteristic velocity,
- L is the characteristic length (chord length of the airfoil),
- ν is the kinematic viscosity.

3.1.2 Initial and Boundary Conditions

In this simulation, the initial and boundary condition are set in a manner appropriate for modeling a plunging elliptical airfoil using the immersed boundary method (IBM) in foam-extend 4.1. The problem is set in two dimensions with a uniform inflow, and the airfoil executes vertical oscillations governed by defined parameter.

Table 1: Initial values of flow variables

Flow Variable	Value
U	0 m/s
p	0 $\mathrm{m^2/s^2}$

Table 2: Velocity boundary conditions

Region	**Condition Type**	**Value (m/s)**
Internal Field	uniform	(1 0 0)
Inlet	fixedValue	(1 0 0)
Outlet	inletOutlet	(1 0 0)
Top & Bottom	fixedValue	(0 0 0)
Front & Back	empty	-
ibAirfoil	movingImmersed-BoundaryVelocity	-

Table 3: Pressure boundary conditions

Region	**Condition Type**	**Value** (m^2/s^2)
Internal Field	uniform	0
Inlet	zeroGradient	-
Outlet	fixedValue	0
Top & Bottom	zeroGradient	-
Front & Back	empty	
ibAirfoil	zeroGradientIb	-

3.1.3 Dynamic Mesh Motion

The plunging motion of the elliptical airfoil is prescribed as a vertical harmonic oscillation governed by amplitude h and reduced frequency k. The reduced frequency is defined by:

$$k = \frac{\omega c}{U} \tag{4}$$

where:

- ω = angular velocity,
- c = chord length (1 m), and
- U = freestream velocity (1 m/s).

For this study, different kh values (kh=0.31 and kh=1) and with reduced frequency $k = 6.283$ are chosen.

Table 4: Parameters used for the plunging airfoil motion

Parameter	Symbol	Value
Chord length	c	1 m
Freestream velocity	U	1 m/s
Amplitude of oscillation	h	0.05m , 0.16m
Reduced frequency	k	6.283
Angular velocity	ω	6.283rad/s
Oscillation period	T	1 s

In foam-extend 4.1, the plunging airfoil is solved using pimpleDyIMbFoam, the motion is implemented using the immersedBoundarySolidBodyMotionFvMesh class, which prescribes motion without causing mesh deformation. The motion parameters are defined in the dynamicMeshDict file using the linearOscillation function.

This approach avoids the need for mesh smoothing or remeshing, as required in ALE-based methods, making it computationally efficient and robust, particularly at high oscillation frequencies. The immersed boundary moves within a fixed Cartesian mesh, updating its location at each time step according to the specified parameters.

4 Solution Method and Control

The unsteady flow around a plunging elliptical airfoil was simulated using the pimpleDyMIbFoam solver in foam-extend 4.1, designed to handle dynamic immersed boundaries without explicit mesh deformation. This solver implements the PIMPLE algorithm, which merges the strengths of both SIMPLE (for pressure–velocity coupling stability) and PISO (for transient accuracy), making it suitable for time-dependent simulations involving moving geometries.

The linear solver settings for pressure and velocity fields are summarized below:

Table 5: Solver settings for pressure and velocity fields

Field	Solver	Preconditioner	Tolerance	relTol
p, pFinal, pcorr	CG	Cholesky	1×10^{-7}	0
U, UFinal	BiCGStab	ILU0	1×10^{-8}	0

The numerical discretization schemes used in the simulation are summarized in Table 6. These schemes define how the various differential terms in the governing equations are approximated in the computational mesh.

Table 6: Discretization schemes used in the simulation

Term	Scheme
Temporal	Euler
Gradient	cellLimited Gauss linear 1
Divergence	Gauss upwind
Laplacian	Gauss linear limited 0.5
Interpolation	linear

5 Results and Discussions

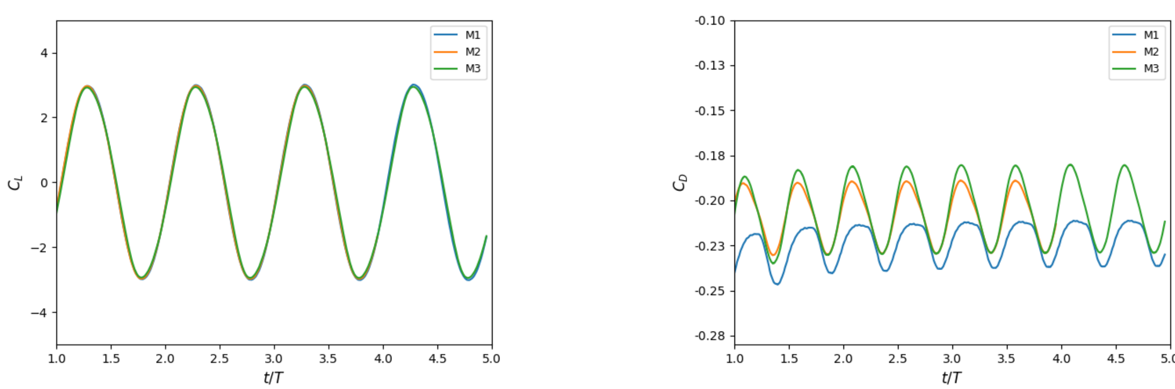
5.1 Convergence Test

5.1.1 Grid Size Convergence Test

To ensure mesh-independent results, a grid convergence study was performed using three mesh resolutions: coarse, medium, and fine. The time-averaged drag coefficient was used as the comparison metric. Table 8 summarizes the results and corresponding percentage errors relative to the finest mesh.

Table 7: Grid independence study

Mesh	Number of Cells	C_D	Error % (vs. Fine)
Coarse (M1)	581,306	-0.2236	8.64
Medium (M2)	1,453,106	-0.2084	1.25
Fine (M3)	2,916,020	-0.2058	–



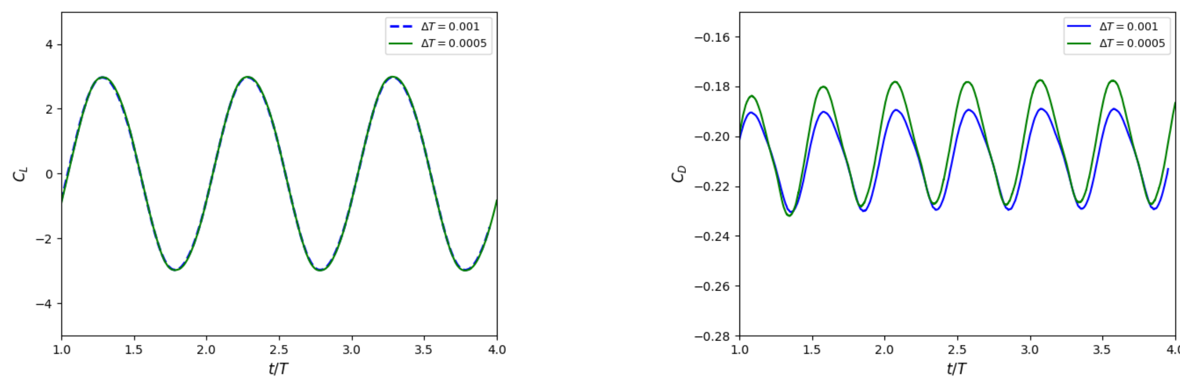
(a) Coefficient of lift (b) Coefficient of drag

Figure 5: Grid Independence Study $kh = 0.31$ & $h = 0.05$

From the results, the variation in C_D between the medium and fine mesh is approximately 1.25%, which indicates sufficient convergence with respect to mesh resolution. Thus, the medium mesh was chosen for the rest of the simulations to maintain a balance between computational cost and accuracy.

5.1.2 Time Step Size Test

To ensure temporal accuracy, a time step sensitivity analysis was performed by comparing results obtained with two different time step sizes: $\Delta t = 0.001$ s and $\Delta t = 0.0005$ s. The comparison focused on the predicted force coefficients, particularly drag and lift. While both time steps were equally effective for coefficient of lift, noticeable difference were observed in capturing the coefficient of drag.



(a) Coefficient of lift (b) Coefficient of drag

Figure 6: Time Step Size Study $kh = 0.31$ & $h = 0.05$

Table 8: Time step size test

Time step size	C_D	**Error %**
0.001	-0.2084	3.07%
0.0005	-0.2160	-

The smaller time step of $\Delta t = 0.0005$ s provided more consistent and accurate results. Consequently, $\Delta t = 0.0005$ s was chosen for the final simulations to ensure numerical stability and accuracy.

5.1.3 Grid Resolution

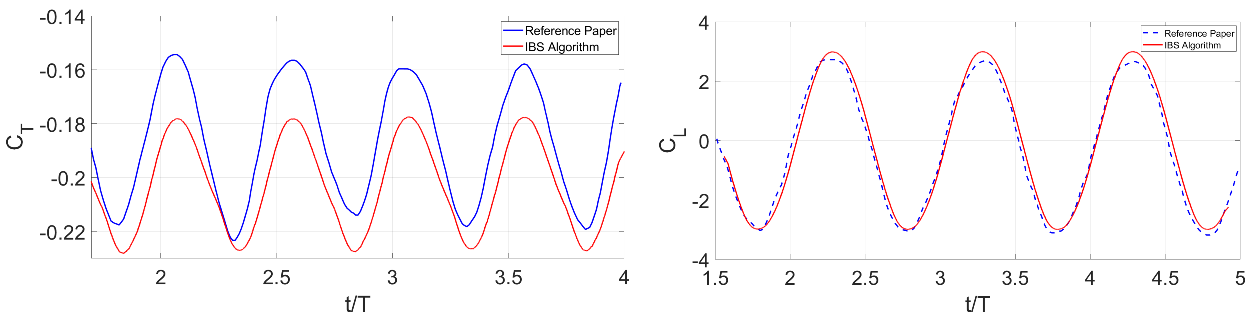
Mesh refinement was done around the plunging airfoil for better accuracy of the simulation. Three different mesh refinement levels were used in the smallest block that encloses the airfoil. The table below summarizes the grid spacing (Δx and Δy) for each mesh:

Table 9: Grid Resolution Near Airfoil

Mesh Level	Δx (m)	Δy (m)
Coarse	0.00849	0.00625
Medium	0.00625	0.00441
Fine	0.00441	0.00312

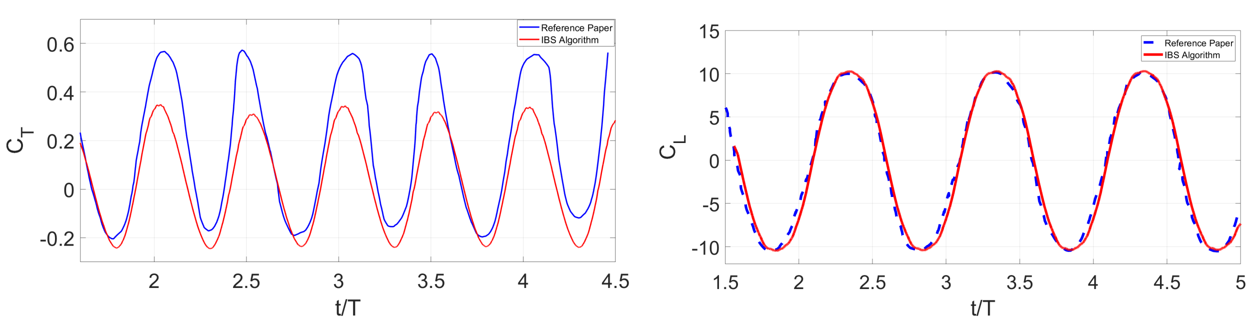
5.2 Results

The simulation study uses the Immersed Boundary Surface Method (IBS), which is implemented in the pimpleDyMIbFoam solver within foam-extend 4.1, to examine the unsteady aerodynamic behavior of an elliptical plunging airfoil. Based on a chord length of 1 m and an input velocity of 1 m/s, all simulations ran at a Reynolds number of 500. In particular, the product of the reduced frequency (k) and the amplitude (h) was used to compare the aerodynamic performance by varying the amplitude while maintaining the reduced frequency at (k) = 6.283: $kh = 1$ and $kh = 0.31$.



(a) Coefficient of drag (b) Coefficient of lift

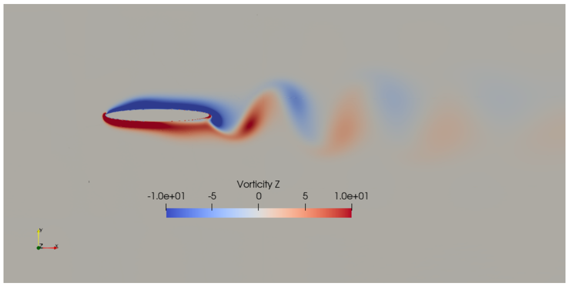
Figure 7: Comparison for $kh = 0.31$, $h = 0.05$ and $k = 6.283$



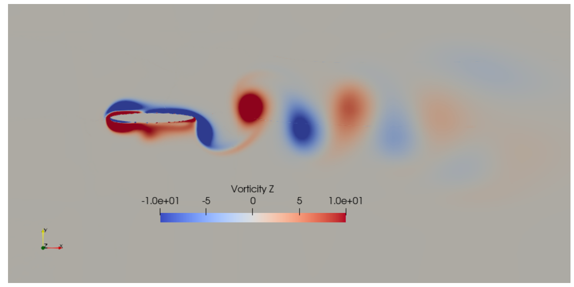
(a) Coefficient of drag (b) Coefficient of lift

Figure 8: Comparison for $kh = 1$, $h = 0.16$ and $k = 6.283$

The lift coefficients (C_L) and thrust coefficients (C_T) were extracted over several oscillation cycles for both scenarios after the flow field was allowed to grow until periodicity was reached. The higher amplitude in the condition $kh = 1$ was found to produce stronger aerodynamic forces. Stronger interactions between the airfoil and the surrounding fluid were indicated by the lift and thrust coefficients, which showed higher peak values. The overall waveform of the force coefficients remained consistent, but their magnitudes were greatly reduced, demonstrating that aerodynamic performance depends on the plunging amplitude and frequency.



(a) kh=0.31



(b) kh=1

Figure 9: Vorticity contour at t=4s

The vorticity contours in figure [9] illustrates the flow structures generated as a result of plunging motion of airfoil. At $kh = 1$, the contours shows strong vortex near the trailing edge. In contrast, for the lower motion intensity of $kh = 0.31$, the vortex formation was weaker, this resulted in the reduction of aerodynamic forces and a comparatively steadier flow field.

6 Conclusions

This study includes the simulation of an elliptical plunging airfoil using the Immersed Boundary Method in foam-extend 4.1 within the pimpleDyIMbFoam solver. All simulations were performed at a Reynolds number of 500 where the airfoil underwent vertical oscillations with a reduced frequency of $k = 6.283$ at different amplitudes resulting in: $kh = 1$ and $kh = 0.31$. The results showed the effect of reduced frequency and amplitude on the aerodynamic forces. The case with $kh = 1$ showed higher peak values of the lift and thrust coefficients, while the case with $kh = 0.31$ showed lower values.

This study illustrates the accuracy of the new IBS algorithm in foam-extend 4.1 for accurately capturing unsteady aerodynamic phenomena without the requirement of mesh deformation, hence providing a robust and efficient framework for simulating moving body problems in fluid mechanics.

7 Acknowledgment

I would like to express my sincere gratitude to Prof. Chandan Bose, for his valuable guidance and support throughout this project. His insights and encouragement have been instrumental in shaping the direction and depth of this work.

I am also immensely grateful to my mentor, Mr. Manjil Sitoula, from the FOSSEE team, for his continuous mentorship, timely feedback, and technical assistance during the internship. His support was crucial in navigating through the complexities of working with OpenFOAM and specially foam-extend. I would also like to thank the entire FOSSEE team for providing this opportunity and for maintaining an environment that fosters learning and research in open-source computational tools.

References

[1] R. Mittal and G. Iaccarino, ''Immersed boundary methods,'' *Annu. Rev. Fluid Mech.*, vol. 37, pp. 239–261, 2005.

[2] C. S. Peskin, ''The immersed boundary method,'' *Acta numerica*, vol. 11, pp. 479–517, 2002.

[3] R. Mittal and G. Iaccarino, ''Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations,'' *Journal of Computational Physics*, vol. 161, no. 1, pp. 35–60, 2000.

[4] M. Uhlmann, ''An immersed boundary method with direct forcing for the simulation of particulate flows,'' *Journal of Computational Physics*, vol. 209, no. 2, pp. 448–476, 2005.

[5] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, ''A tensorial approach to computational continuum mechanics using object-oriented techniques,'' *Computers in Physics*, vol. 12, no. 6, pp. 620–631, Nov. 1998.

[6] J. E. Döhler, ''An analysis of the immersed boundary surface method in foam-extend,'' 2022.

[7] H. Jasak, ''Immersed Boundary Surface Method foam-extend,'' 2018.

[8] D. Majumdar, C. Bose, and S. Sarkar, ''Capturing the dynamical transitions in the flow-field of a flapping foil using immersed boundary method,'' *Journal of Fluids and Structures*, vol. 95, p. 102999, 2020.

[9] F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*, ser. Fluid Mechanics and Its Applications. Cham: Springer International Publishing, 2016, vol. 113.