

**IMMERSED BOUNDARY METHOD FOR ROTORCRAFT
AERODYNAMICS USING OPENFOAM**

**GÖMÜLÜ SINIR METODU İLE DÖNER KANATLI HAVA
ARACI AERODİNAMİĞİNİN OPENFOAM
KULLANILARAK İNCELENMESİ**

ŞEVKİ ÇAĞRI KORALAY

DOÇ. DR. ÖZGÜR EKİCİ
Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Master of Science
in Department of Mechanical Engineering

September 2022

ABSTRACT

IMMERSED BOUNDARY METHOD FOR ROTORCRAFT AERODYNAMICS USING OPENFOAM

Şevki Çağrı KORALAY

Master of Science , Department of Mechanical Engineering

Supervisor: Doç. Dr. Özgür EKİCİ

September 2022, 66 pages

The rotorcraft CFD literature review showed that efficient, easy to apply, and computationally less expensive methods are still a need for the design and performance predictions of rotorcraft. Immersed Boundary Method (IBM) and Virtual Blade Model (VBM) based simulations have the potential to minimize the user input and computational expense while providing reasonably accurate results. Distinctive property of the IBM method is that, the whole simulation is run on a Cartesian grid that does not conform to the solid boundary. So that, in order to define the boundary conditions, the governing equations near solid boundaries are modified. Virtual Blade Model is a validated and verified way of modeling the aerodynamics of the rotors. It provides the benefit of using Navier-Stokes equations, instead of wake modeling, without being computationally expensive. In this work, an IBM code from a different study and VBM is integrated in the finite volume based Navier-Stokes solver Open Field Operation and Manipulation (OpenFOAM) platform specifically for rotorcraft simulations. The integrated code is verified and validated with the extensively studied wind tunnel experiment of a rotorcraft in forward flight. Finally, the result is compared and discussed against body conformal methods. Although some differences are observed, it is concluded that the results are on par with the body-conformal methods.

Keywords: Immersed Boundary Method, Blade Element Momentum Theory, Virtual Blade Model, Rotorcraft, CFD, OpenFOAM

ÖZET

GÖMÜLÜ SINIR METODU İLE DÖNER KANATLI HAVA ARACI AERODİNAMIĞİNİN OPENFOAM KULLANILARAK İNCELENMESİ

Şevki Çağrı KORALAY

Yüksek Lisans, Makina Mühendisliği

Danışman: Doç. Dr. Özgür EKİCİ

Eylül 2022, 66 sayfa

Döner kanatlı hava araçları üzerine literatürdeki hesaplamalı analiz çalışmaları incelendiğinde, hesaplama maliyeti düşük ve uygulaması kolay metodlara hala ihtiyaç bulunduğu görülmüştür. Gömülü Sınır Metodu (GSM) vs Sanal Pala Modelinin (SPM) beraber kullanılmasının, sonuçların doğruluğundan ödün vermeden kullanıcı girdisini ve hesaplama maliyetlerini azaltabileceği düşünülmektedir. GSM metodunda, sayısal ağ, analizi yapılacak gövdeden bağımsız olarak atılır. Katı sınırlar ise korunum denklemlerinde yapılan değişikliklerle hesaplamalara dahil edilir. SPM ise temellerini Pala Elemanı Momentum Teorisinden alır ve onun OpenFOAM'a uyarlanmış halidir. Bu sayede, rotor etkisini kompleks ve maliyetli bir şekilde modellemeye gerek kalmadan, Navier-Stokes temelli bir çözüm elde edilebilmektedir. Bu çalışmada, başka bir çalışmadan alınan GSM kodu ve SPM, OpenFOAM üzerinde birleştirilmiş ve bir rüzgar tüneli deneyi modellenmiştir. Elde edilen sonuçlar hem deney verileriyle hem de gövdeye uyumlu sayısal ağlarla karşılaştırılmıştır. Sonuçlarda bazı farklar olsa da, GSM ile elde edilen sonuçların doğruluğunun, gövdeye uyumlu sayısal ağlarla elde edilenler kadar iyi olduğu gözlenmiştir.

Keywords: Gml Sınır Metodu, Pala Elemanı Momentum Teorisi, Sanal Pala Modeli,
Dner kanatlı hava aracı, HAD, OpenFOAM

ACKNOWLEDGEMENTS

Firstly, I want to express my gratitude to my supervisor Doç. Dr. Özgür Ekici for his support and guidance throughout this study.

I am also indebted to my friends and supervisors in Lentatek Company, which had been my home for the last 4 years.

I also want to share my appreciation for each and every one who taught me anything, big or small, in my field. They are too many to name but their contributions were invaluable.

Finally, my greatest appreciation goes to my family for always supporting me in their own way. I am glad to have them in my life.

CONTENTS

	<u>Page</u>
ABSTRACT	i
ÖZET	iii
ACKNOWLEDGEMENTS	v
CONTENTS	vi
TABLES	viii
FIGURES	ix
ABBREVIATIONS AND SYMBOLS	xi
1. INTRODUCTION	1
1.1. Scope of The Thesis	1
1.2. Contributions	1
1.3. Organization	2
2. LITERATURE REVIEW	3
2.1. Rotorcraft CFD Simulations	3
2.2. Immersed Boundary Method.....	6
2.2.1. Continuous Forcing Approach	8
2.2.2. Discrete Forcing Approach	9
2.2.2.1. Indirect BC Imposition	10
2.2.2.2. Direct BC Imposition.....	10
3. THEORY AND MODELING	13
3.1. Rotorcraft Aerodynamics	13
3.1.1. Brief Description of Rotor Aerodynamics	13
3.1.2. Description of Some Key Terms	16
3.1.2.1. Figure of Merit and Disk Loading	16
3.1.2.2. Coning	16
3.1.2.3. Flapping.....	17
3.1.2.4. Solidity	18
3.2. The Virtual Blade Model	18

3.2.1. Blade Element Momentum Theory	19
3.3. IBM in OpenFOAM	26
3.4. IBM with y+ Adaptive Wall Function	27
3.4.0.1. Representation of Immersed Boundaries	28
3.4.0.2. y+ Adaptive Immersed Wall Model	29
3.5. Governing Equations	31
4. DEMONSTRATION: GEORGIA INSTITUTE OF TECHNOLOGY AIRFRAME ..	34
4.1. GIT Experimental Setup	34
4.2. Computational Setup	37
4.2.1. Geometry Preparation.....	38
4.2.2. Meshing Strategy.....	39
4.2.2.1. Defining the Nonconformal Boundary Condition.....	41
4.2.3. Computational Setup.....	42
4.2.4. Flow Solver Settings	45
5. RESULTS AND DISCUSSIONS	47
5.1. Pressure Field	49
5.2. Coefficient of Pressure Comparison	50
5.3. Velocity Field Comparison	53
6. CONCLUSION	56
6.1. Conclusion	56
6.2. Future Research.....	57
APPENDIX: Originality Report	66

TABLES

	<u>Page</u>
Table 4.1 Rotor parameters of the GIT experiment.....	36
Table 4.2 Wind tunnel operation and airframe parameters	37
Table 5.1 Pressure coefficient RMS error values for each side	54

FIGURES

	<u>Page</u>
Figure 2.1 CFD simulation of the Blackhawk rotor in which the rotor wake and blade-tip vortices are visualized with isosurfaces of the Q-criterion ...	4
Figure 2.2 Sea King MK50 fuselage mesh.....	5
Figure 2.3 Immersed boundary mesh and body-conforming mesh example	6
Figure 2.4 Linear velocity interpolation scheme used by Fadlun et al.	10
Figure 2.5 Direct BC Imposition interpolation scheme used by Majumdar et al. ...	11
Figure 3.1 Incident velocity of the blade in a) hovering flight and b) forward flight	14
Figure 3.2 Flow field of a helicopter in forward flight	15
Figure 3.3 Effects of the lift and centrifugal force on coning angle	17
Figure 3.4 lead-lag, flapping and feathering motion shown on a rotor blade.....	17
Figure 3.5 Aerodynamic forces and local fluid velocity component on an airfoil .	20
Figure 3.6 Sketch of a generic lift coefficient curve	21
Figure 3.7 Sketch of a generic drag coefficient curve	21
Figure 3.8 Structured hexahedral mesh, suitable to be used with the VBM.....	23
Figure 3.9 Integration of VBM with the simpleFoam solver.....	25
Figure 3.10 Interpolation of the IB cells by Jasak et al.	27
Figure 3.11 Representation of Immersed Boundary, IB cell centers, hit points, image points and immersed interface Γ_{IB} by Xu and Liu	29
Figure 3.12 A simplified representation of the y+ adaptation process by Xu and Liu	30
Figure 4.1 Side view of airframe installation with respect to wind tunnel	35
Figure 4.2 Upstream and top view of airframe and rotor.....	35
Figure 4.3 General view of the domain and the reference axis origin	38
Figure 4.4 A rotor disk that has structured hexahedral mesh and is suitable to be used for VBM	39

Figure 4.5	Cross section of domain mesh and refinement areas. Due to a rendering problem, some irregular lines are visible that are not present in the mesh itself.....	40
Figure 4.6	Nonconformal rotor and domain mesh closeup.....	41
Figure 5.1	Comparison of the pressure values taken from the top of the airframe, obtained from three different meshes	47
Figure 5.2	Comparison of the downward velocity values taken from 12.7mm above from the top of the airframe, obtained from three different meshes	48
Figure 5.3	The front view of the pressure field in Pascal around the airframe	49
Figure 5.4	The side view of the pressure field in Pascal around the airframe	49
Figure 5.5	Comparison of measured and computed C_p on top of the airframe	51
Figure 5.6	Comparison of measured and computed C_p on the bottom of the airframe.....	51
Figure 5.7	Comparison of measured and computed C_p on the port side of the airframe.....	52
Figure 5.8	Comparison of measured and computed C_p on the starboard side of the airframe.....	53
Figure 5.9	Location of the axial and downward velocity measurement	54
Figure 5.10	Comparison of computed and measured axial velocities above the airframe.....	54
Figure 5.11	Comparison of computed and measured downward velocities above the airframe.....	55

ABBREVIATIONS AND SYMBOLS

ABBREVIATIONS

AMI	: Arbitrary Mesh Interface
AMR	: Adaptive Mesh Refinement
AoA	: Angle of Attack
BC	: Body Conformal
BEM	: Blade Element Momentum Theory
IB	: Immersed Boundary
IBM	: Immersed Boundary Method
LRF	: Local Rotor Frame of Reference
RANS	: Reynolds Averaged Navier Stokes
RMS	: Root Mean Square
RPM	: Rotations Per Minute
VBM	: Virtual Blade Model

SYMBOLS

α	Angle of Attack
α_{eff}	Effective Angle of Attack
Γ_b	Immersed Boundary
ϵ	Turbulence Energy Dissipation Rate
μ	Dynamical Viscosity
ν	Kinematic Viscosity
ρ	Density
σ	Rotor Solidity
$\tau_{i,j,turbulent}$	Reynolds Stress Tensor
Φ	Arbitrary Flow Function
Ψ	Blade Azimuth Angle

ω	Angular Velocity
Ω_f	Fluid Domain
c	Blade Chord Length
C_{exp}	Experimental Value Measured for Pressure Coefficient
C_{calc}	Calculated Value Measured for Pressure Coefficient
C_d	Aerodynamic Drag Coefficient
$C_{d\alpha}$	Slope of C_d Curve with Respect to Angle of Attack
C_l	Aerodynamic Lift Coefficient
$C_{l\alpha}$	Slope of C_l Curve with Respect to Angle of Attack
C_μ	k - ϵ Model Constant
D_H	Hydraulic Diameter
f	Blade Section Aerodynamic Force
F	Blade Total Aerodynamic Force
f_b	Boundary Forcing Function Including Momentum and Pressure Forcing
$f_{momentum}$	Momentum Forcing Function
$f_{pressure}$	Pressure Forcing Function
g	Gravitational Acceleration
I_{turb}	Turbulence Intensity
k	Turbulence Energy
M	Mach Speed
n	Sample Size
N	Number of Blades
\vec{u}	Fluid Velocity Vector
NS	Navier Stokes Operations
p	Pressure
P	Power
S	Scaling Factor Used in VBM
T	Thrust
Q	Torque

R	Radius
u	Velocity in x-direction
v	Velocity in y-direction
V	Velocity Scalar
V_{cell}	Volume of the Rotor Mesh Cell Element
w	Velocity in z-direction

Subcripts

b	Boundary
d	Drag
e	Effective
f	Fluid
i	Induced
l	Lift
n	Normal
t	Tangent

1. INTRODUCTION

The literature review showed that, even today, the numerical analysis of rotorcraft aerodynamics is a challenging field. The efforts go back to the 1990s and start with the usage of source panel methods. Since then, a tendency to increase in mesh size and computational expense is observed. The study in this thesis is an effort in the opposite direction. Two methods for modeling rotor and airframe are introduced and explained in detail. It is suggested that using these methods together can decrease preprocessing effort and time without compromising accuracy. An experimental case is modeled and results are shared for supporting these claims. The results are shared in detail and they are found to be satisfactory.

1.1. Scope of The Thesis

This thesis mainly focuses on the integration, application and benchmarking of the Immersed Boundary Method with the Virtual Blade Model in the context of rotorcraft aerodynamical analysis.

1.2. Contributions

In this study, the aim was to propose a novel CFD approach that is computationally inexpensive, easy to apply, and has accuracy at least at the level of body-conformal methods.

This is done by the following steps:

- The IBM code from an existing study and VBM were integrated in OpenFOAM,
- The GIT rotor-airframe experiment was modeled, solved and compared against the experimental results and body-conformal analysis,
- For solving the problems faced during modeling, nonconformal mesh interface (cyclicAMI) and solution mapping (mapFields) were used,

- The simulation results showed good agreement with the experiment, in some cases better than the body-conformal method,
- Although there are limited examples of rotor model integration with IBM, at the time of writing no IBM-VBM study was found on literature and this is believed to be a novel, open source and free to use approach.

1.3. Organization

The organization of the thesis is as follows:

- Chapter 1 presents the motivation, contributions, and the scope of the thesis.
- Chapter 2 provides the literature review on rotorcraft CFD simulations and the immersed boundary methods
- Chapter 3 gives information about rotorcraft aerodynamics, VBM, IBM in OpenFOAM and the IBM used in this study
- Chapter 4 explains the details of the benchmark case GIT rotor-airframe experiment and its modeling in OpenFOAM
- Chapter 5 demonstrates the results and reached conclusion
- Chapter 6 states the summary of the thesis and possible future directions.

2. LITERATURE REVIEW

The literature review of this study is presented in two main sections. The first section is a general review of Rotorcraft CFD Simulations. It includes a short summary of the history of rotorcraft simulations, the up-to-date solution approaches, and remaining challenges.

At the second part, starting from its initial steps, the fundamentals of the Immersed Boundary Method will be introduced. Two main approaches and several subcategories with their advantages and disadvantages will be explained.

2.1. Rotorcraft CFD Simulations

A rotorcraft is a type of aircraft that generates lift by rotary wings which are usually called rotor blades. Its main advantages are the ability to hover and the ability to move with 6 degrees of freedom. These advantages make it an important part of many civilian and military operations.

Although they operate according to the same basic aerodynamical principles, rotorcraft and fixed-wing aircraft have their own unique challenges. A wing that's moving forward leaves the generated tip vortex behind quickly. On the other hand, for a rotorcraft, the vortex wake system remains in the area of the rotor and changes the effective angle of attack seen by the rotor blades [1]. Even in the forward flight, the blade tip vortices interact with the successive blades and creates a phenomenon called blade-vortex interactions [2]. The complex problem of the flow field around a helicopter is still a challenge and it is being attacked by many different approaches.

Many CFD codes have been developed to deal with the complex flow field of the rotorcrafts. These efforts go back to the development of a computational method by Lorber and Egolf [4] to resolve the unsteady flow interactions using source panel methods and the method has been used to simulate the experimental rotor-airframe interaction study of the GIT (Georgia Institute of Technology) [5]. For the same GIT geometry, Zori and Rajagopalan [6] solved

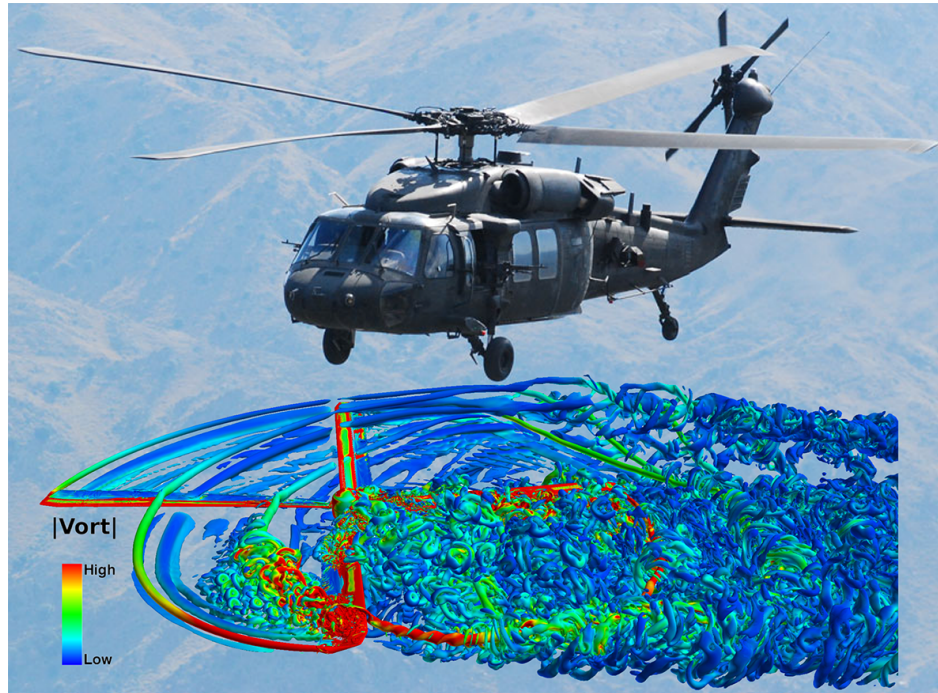


Figure 2.1 CFD simulation of the Blackhawk rotor in which the rotor wake and blade-tip vortices are visualized with isosurfaces of the Q-criterion [3]

3D, incompressible, laminar Navier-Stokes equations. Hariharan and Shankar [7] solved Euler equations with an implicit finite volume scheme and used an overset grid methodology for GIT airframe. On a Dauphin 365N helicopter configuration (which is more elaborate than the GIT airframe) Renaud et al. [8] carried out RANS (Reynolds-Averaged Navier-Stokes) simulations with the elsA solver (Ensemble Logiciel de Simulation en Aérodynamique) [9] which uses multiblock structure meshes and finite volume formulation. DLR (Deutsches Zentrum für Luft- und Raumfahrt e.V.) developed two Navier-Stokes solvers; one is a multiblock structured flow solver FLOWer [10] and the other is the unstructured flow solver Tau (τ) [11].

In the struggle for obtaining a high-fidelity analysis and simulation tool, there have been some international collaboration efforts that used a single geometry for benchmarking. Through a collaboration between AFDD (U.S. Army Aeroflightdynamics Directorate), ONERA (Office National d'Etudes et de Recherches Aérospatiales) and GIT; Renaud et al. [12] and Potsdam et al. [13] used the Dauphin 365N geometry for comparing their multiblock, overset mesh solvers; structured solvers OVERFLOW (overset grid flow), elsA

and unstructured solver FUN3D (Fully Unstructured Navier-Stokes 3D). In their paper, Renaud et al. [12] studied the steady flow conditions and modeled the rotor as an actuator disk. Whereas Potsdam et al. [13] studied unsteady flow conditions and modeled the rotor as individual, discrete moving blades, which is more expensive computationally.

There have been simulations with a very high number of cells, up to 902 million nodes to be specific, by using the Helios (Helicopter overset simulations) code [14]. Helios is a rotorcraft simulation code that's continuously under development and includes different solvers, e.g. OVERFLOW, FUN3D, NSU3D (Navier-Stokes Unstructured 3D) and additionally AMR (adaptive mesh refinement) Cartesian solvers such as; SAMCart, SAMARC and mStrand [14]. This is a highly capable code that's still in use by the U.S. Army. This tool has been used to study different types of rotorcraft including [15]; LRTA (Large Rotor Test Apparatus), Apache AH-64E, UH-60A Black Hawk (Figure 2.1), FLRAA (Future Long-Range Assault Aircraft), ROBIN (Rotor Body Interaction) and Sea King MK50 (Figure 2.2).

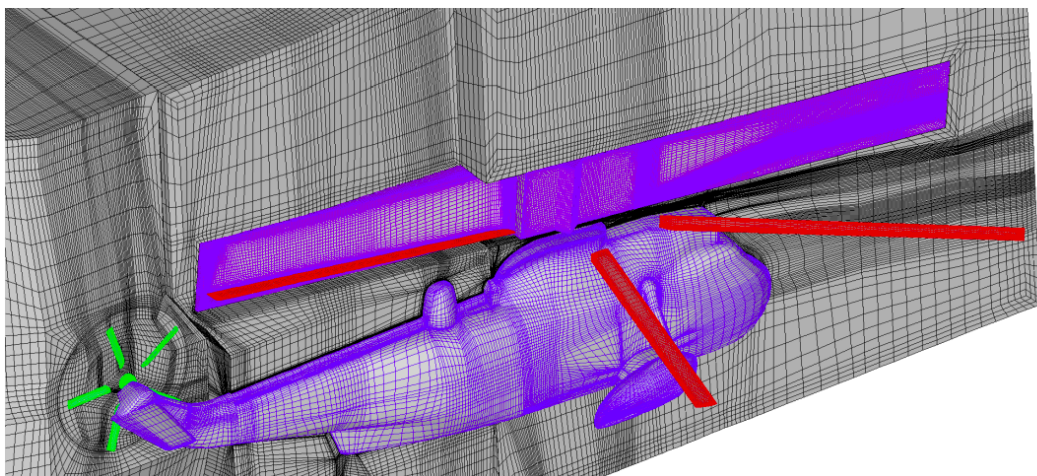


Figure 2.2 Sea King MK50 fuselage mesh from the study [16]

Analyzing the aforementioned studies unveils some commonalities. It can be seen that, for the last 15 years there are some popular codes that's used and improved. These codes mainly incorporate Overset and AMR methods for analysis. Also, due to the complex nature of the problem at hand and with the increase in computational power due to technology, a tendency to increase mesh size is observed. These methods are demanding both computationally and effort-wise. It is clear that less demanding methods with high precision are needed

in the rotorcraft CFD studies. Immersed Boundary Method (IBM) based simulations have the potential to minimize the user input while keeping the mesh size reasonable, without sacrificing the precision of simulation results.

2.2. Immersed Boundary Method

Dealing with complex and moving geometries has been one of the challenges in CFD applications. Part of the problem is that applying a high-quality mesh to these kinds of geometries usually requires a skilled CFD practitioner, a significant amount of time, and effort. IBM has the potential to fasten the CFD preprocessing step without compromising mesh quality.

The IBM is a meshing approach in which the grid is applied without being conformal to the body as in Figure 2.3. The body is immersed in a high-quality Cartesian mesh and the boundary conditions are applied by modifying the governing equations in the region of the boundary.

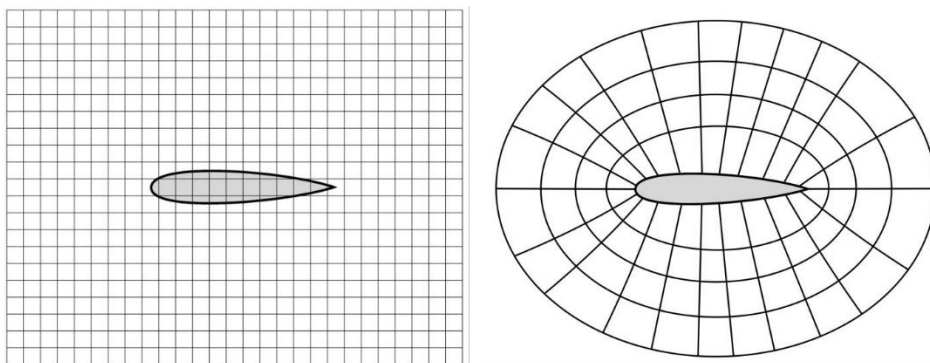


Figure 2.3 Immersed boundary mesh and body-conforming mesh example [15]

It is known that the aim of body conformal methods is to build a grid with sufficiently enough local resolution to correctly resolve flow properties and to have the least amount of total grid elements possible to keep the computational cost as low as possible. These contradictory needs can result in a low grid quality and can negatively impact the accuracy and convergence properties of the solver [17]. Even for simple cases, it usually is an iterative and cumbersome process that requires a high amount of skilled input from the CFD practitioner. Additionally,

for cases with time-dependent moving boundaries, the meshing process is done repeatedly for each time step and a method is required to project the solution to the grid for the next time step [18]. The IBM method requires a less demanding and easier-to-produce grid which is a high-quality orthogonal Cartesian mesh that's fixed for each time step.

This non-conformal grid approach has two significant advantages [15]. The first one is that it allows automated grid generation with minimal input from the user while at the same time maintaining a low level of non-orthogonality and skewness. The second advantage is that it may allow a better convergence to be achieved due to uncompromised mesh quality. These advantages make the method especially suitable for objects in motion because once a high-quality mesh is produced, it can be used for every time step without any change.

Although it has clear advantages, the application of IBM is not straightforward. The boundary conditions are applied by modifying the governing equations and reproducing the effect of the boundary near the edge of the solid. Additionally, body conformal methods allow better control of grid resolution around the body due to flow alignment. That results in a higher increase in IBM mesh size due to Re increase, with respect to body conformal grids [19].

IBM term was first used for the method developed by Peskin [20] in 1972. The goal of the work was to develop a method to find a numerical solution to the flow patterns around heart valves. This particular problem involves the interaction of an elastic solid with a fluid in which the solid boundary is changing at each time step. Since then, a considerable number of modifications and improvements have been offered about the method. The distinguishing quality of these methods is the different approaches to how and when the effect of boundary is defined. The details of imposing the boundary conditions on the Immersed Boundary also determine the classification of the IB method [19].

The boundary condition of immersed geometry is applied implicitly by adding a forcing function or a source term to the corresponding governing equation. The differences in the application of boundary conditions between IB methods can be summed up under two different ways: Continuous Forcing Approach and Discrete Forcing Approach.

2.2.1. Continuous Forcing Approach

Continuous forcing approach is the original IB method used by Peskin [20] for flows with elastic boundaries, such as muscle contraction of the beating heart. The flow is governed by incompressible Navier-Stokes equations. The heart is simulated as a set of elastic fibers and massless points that are tracked in a Lagrangian way. The deformation of elastic fibers due to blood flow is modeled by Hooke's law.

For a hypothetical, 2D, incompressible flow past the body, assume that the following governing equations define the flow:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p - \frac{\mu}{\rho} \nabla^2 \vec{u} = 0 \quad (1)$$

$$\nabla \cdot \vec{u} = 0 \quad \text{in } \Omega_f$$

$$\vec{u} = \vec{u}_\Gamma \quad \text{on } \Gamma_b \quad (2)$$

$$NS(\underline{U}) = 0 \quad \text{in } \Omega_f \quad (3)$$

$$\underline{U} = (\vec{u}, p)$$

$$\underline{U} = \underline{U}_\Gamma \quad \text{on } \Gamma_b \quad (4)$$

Where \vec{u} is the flow velocity, p is pressure, μ is dynamic viscosity, Ω_f defines the fluid domain, \underline{U} is equal to (\vec{u}, p) , Γ_b defines the immersed boundary and NS operator represents the Navier-Stokes equations.

The continuous forcing approach is applied by modifying continuous equation (3) and introducing the forcing function $\underline{f}_b = (\vec{f}_{momentum}, f_{pressure})$ and then discretizing this equation over the grid to get the system of equations (6) [19]. Then this system of equations is solved in the entire domain.

$$NS(\underline{U}) = \underline{f}_b \quad (5)$$

$$[NS]\{\underline{U}\} = \left\{ \underline{f}_b \right\} \quad (6)$$

This approach has been used for many different kinds of problems in various areas. Such as modeling cochlea [21] and sperm motility [22].

Continuous forcing method can be modified to be used with rigid boundaries. One approach is to assume that the body is an elastic body with extreme stiffness [19]. Another one is that the solid is attached to a specific location with a spring, that has a large value of spring constant [23]. Although this forcing approach is very preferable for biological phenomena with elastic boundaries, it is observed that it leads numerical accuracy and stability problems for rigid boundaries [19].

2.2.2. Discrete Forcing Approach

In this approach, first the continuous governing equations of fluid flow are discretized through the whole grid as if no solid boundary exists. Then, the forcing term is applied to the discretized equations in the vicinity of the immersed boundary.

Continuous Navier-Stokes equation (3) will be discretized to get the equation (7):

$$[NS]\{\underline{U}\} = 0 \quad (7)$$

Then, the discretization terms near the IB cells will be adjusted for the presence of the boundary:

$$[NS]\{\underline{U}\} = \left\{ \underline{f}'_b \right\} \quad (8)$$

The preference of discretization method is an important aspect of this approach because it directly affects the numerical accuracy and stability. Unlike Continuous Forcing Approach, this method does not need user-defined parameters in the forcing term [19], which is one of the attractive features of it. There are two types of boundary condition imposition for this approach.

2.2.2.1. Indirect BC Imposition Navier-Stokes equations cannot be analytically integrated and that characteristic makes it a challenge to derive a forcing term that defines a specific boundary condition. The Continuous Forcing Approach avoids this problem by using simplified versions of the required forcing.

Mohd-Yusof [24] proposed another method that determines an a priori estimate of the forcing extracted from the numerical solution. Verzicco et al. [25] applied this method to the combustion engine and Balaras [26] applied it to flow on a 2D bluff body. On the other hand, Fadlun et al. [27] proposed a linear velocity reconstruction. By which the first grid velocity is linearly interpolated between the second grid point and the desired velocity in the boundary that's immersed.

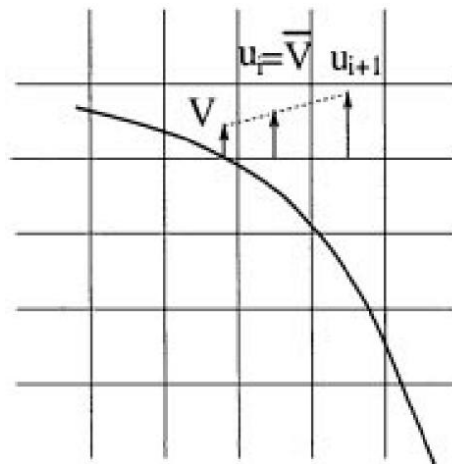


Figure 2.4 Linear velocity interpolation scheme used by Fadlun et al. [27]

2.2.2.2. Direct BC Imposition The direct BC imposition aims to define the solid boundary “sharply” unlike the aforementioned methods that define the immersed boundary spreaded due to the smooth force distribution function [19]. This imposition is especially preferred for the type of flows for which correctly resolving the local boundary layer is important. The method uses a high order interpolation scheme for the IB to obtain a better local accuracy.

For example, Majumdar et al. [28] proposed defining a ghost cell G inside the solid body as shown in Figure 2.4 and assuming a linear variation of Φ (an arbitrary flow variable) along

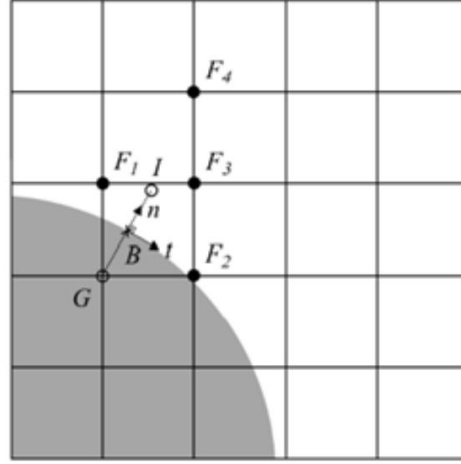


Figure 2.5 Direct BC Imposition interpolation scheme used by Majumdar et al. [28]

the line up to point I. Φ_G , value of the ghost cell, is expressed as:

$$\Phi_G = 2\Phi_B - \Phi_I \quad (9)$$

Then, by using the following quadratic interpolation to find Φ_I :

$$\Phi_I = C_1 n^2 + C_2 n t + C_3 n + C_4 t + C_5 \quad (10)$$

where n is normal and t is tangential wall coordinates. The coefficients C_1 to C_5 are calculated from F_1, \dots, F_4 . Flow past over an airfoil and a circular cylinder has been studied by Ghias et al. [29] via this method. Another method that is specifically designed to satisfy the conservation laws for the cells near the immersed boundary is the cut-cell finite volume method. To apply this method, Cartesian mesh cells in the immersed body are truncated to get a body conformal mesh. This results in irregularly shaped cells and a suitable numerical scheme should be employed to solve for these cells. It has been used by Mittal et al. to simulate flow induced vibrations [30] and flapping foils [31].

In this section, two main approaches of IB methods and their examples in the literature are presented. The literature review shows that using IB methods for rotorcraft aerodynamics

analysis is an uncommon approach. This study is believed to make a unique contribution in this context.

3. THEORY AND MODELING

This chapter starts by introducing the basics of rotorcraft aerodynamics and moves forward by describing some key terms. After the related theoretical information was laid down, details of the Virtual Blade Model and its relation with the Blade Element Momentum Theory are explained. The final parts are reserved for the IBM applications in OpenFOAM. The available applications are shared and the algorithm used in this study is explained in detail.

3.1. Rotorcraft Aerodynamics

The modern helicopter is a truly capable aircraft. Unlike any other aircraft, it can hover stationary relative to ground or fly forward, backward or sideways. Additionally, it has vertical landing and takeoff capabilities. Helicopter rotor has three functions:

1. Vertical lift force generation opposite to helicopter's weight,
2. Horizontal propulsive force generation for forward flight
3. Source of forces and moments to control attitude and position in the 3D space [32]

Thanks to these capabilities, helicopter is a unique aircraft that has its own challenges.

In the next part, rotor aerodynamics will be introduced briefly and after that, some key terms of the rotor will be explained. This information will help the reader to comprehend the upcoming chapters more easily.

3.1.1. Brief Description of Rotor Aerodynamics

To describe the arbitrary positions inside a rotor disk, a specific cylindrical coordinate system is used. As expected, origin is located at the disk center. The radial and azimuthal positions are represented by the variables r and ψ respectively. When the rotor blade is directed downstream, ψ is defined as zero. Additionally, rotor flap angle is represented by β .

The convention for the rotor rotation direction is positive in counterclockwise direction when viewed from above. This direction of rotation is said to have positive angular velocity. While the helicopter is flying forward and there is positive angular velocity in the rotor, the right half of the rotor is called the advancing side because the rotor blades are advancing to the freestream. By the same logic, the left half of the rotor is called the retreating side. Due to rotation of the rotor, advancing and retreating sides experience different local velocities. This creates unequal lift for both sides and is called dissymmetry of lift. From

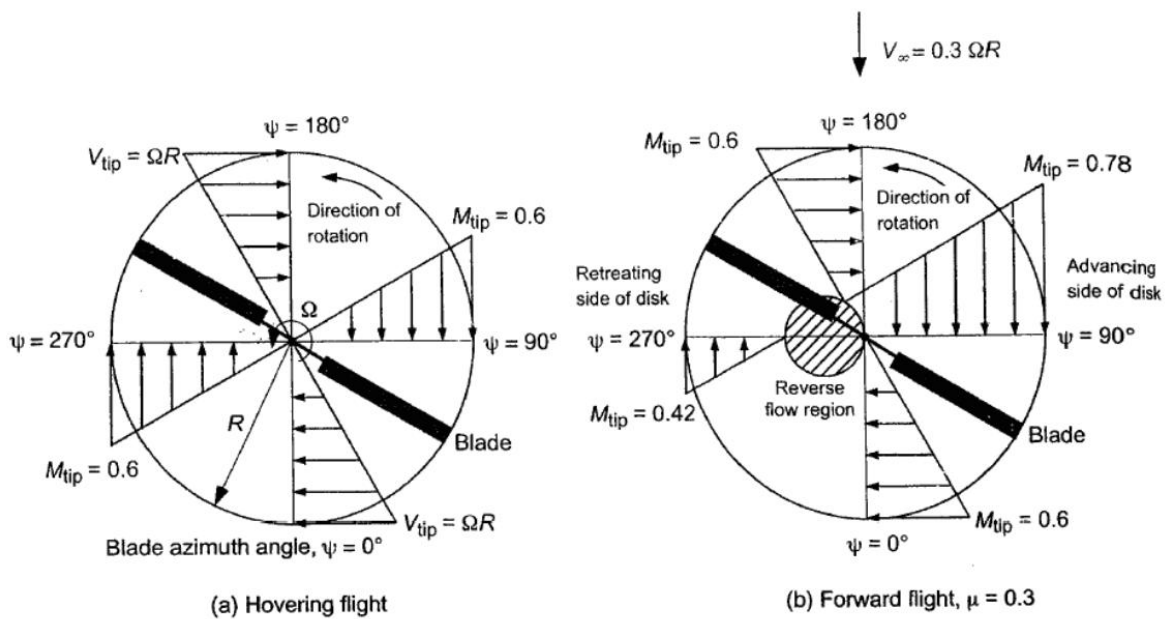


Figure 3.1 Incident velocity of the blade in a) hovering flight and b) forward flight [32]

the aerodynamics perspective, the helicopter rotor possesses complex flow. In forward flight, the free stream velocity component is added or subtracted from the rotational velocity at each part of the blade. Even if the velocity distribution along the blade remains linear, the velocity distribution is no longer axisymmetric. There are additional factors determining the AoA (angle of attack) distribution of the blade sections, such as; blade pitch angle, blade flapping, and induced flow through the rotor. As it is known that the AoA of blade sections determines the lift distribution of the blades which in turn determines rotor thrust and rotor power consumption. Due to the high dynamic pressure that is present at the rotor blade tips, strong vortices are formed. Depending on the forward speed of the helicopter, tips on the advancing side of the blade (Figure 3.1) can reach transonic or even supersonic speeds. On

the other hand, the tips of blades that are retreating away from the relative wind due to the forward speed of the helicopter, need to operate at a higher AoA due to lower local velocity. When the AoA becomes too high, the blade section fails to provide lift and stall occurs. These are the reasons that make the aerodynamic analysis of the helicopter rotor, at hover, climb and forward flight, intrinsically difficult [32]. Adding the helicopter fuselage, hub, tail and tail rotor to that flow field makes the aerodynamics of the helicopter even a harder problem to solve. A visual representation of the flow field can be seen in Figure 3.2. It is also

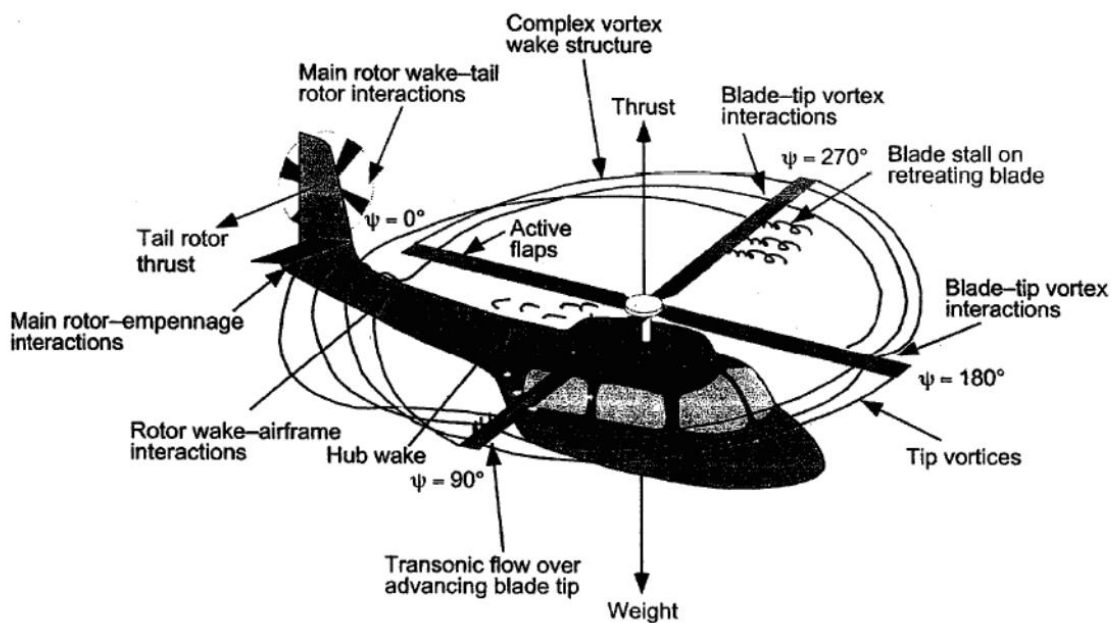


Figure 3.2 Flow field of a helicopter in forward flight [32]

important to understand the aerodynamic sources of consumed power. The engine power is consumed mostly by the main rotor and there are two main components of the rotor power. These are induced power and profile power.

The work done by the rotor on surrounding air results in a kinetic energy increase in the rotor slipstream. That increase in momentum creates the thrust for the rotorcraft. This inescapable power loss is named induced power. Similar to fixed-wing aircraft, to overcome the aerodynamic drag of the blades, another power component is required and this type of power loss is called profile power. For the hover, approximate distribution of power used by

the rotor is [33] %60 due to induced power and %30 due to profile power. The remaining %10 is due to nonuniform inflow, swirl in the wake, and tip losses.

3.1.2. Description of Some Key Terms

3.1.2.1. Figure of Merit and Disk Loading To measure the efficiency of a rotor in hover, a useful and important parameter is the Figure of Merit. It is defined as the ratio of the minimum possible power required to hover to the actual power required [32]. The minimum power required is calculated according to the idealized case where the only power loss is due to the induced power. The induced power loss can be calculated by the momentum theory.

The figure of merit is similar to the propulsive efficiency but its use is limited to only constant disk loading. Disk loading is a term that's used repeatedly in rotor analysis. It is, thrust divided by the rotor area. By only increasing the disk loading, induced power can be increased relative to the profile power [33]. That results in an increase in the figure of merit. Although this can not be considered as an efficiency increase of the rotor. This is an important observation that leads to the conclusion that the use of Figure of Merit is meaningful only when comparing rotors with the same Disk Loading [32]. It is an important measurement tool for evaluating rotor aerodynamic efficiency within its limitations. It is especially useful for rotor design optimization while comparing the effect of airfoil choice, twist amount, and blade planform geometry [33].

3.1.2.2. Coning Rotor blades are basically very high aspect ratio wings that are rotating at a high RPM. Due to this movement, two major forces act on the blades simultaneously. These are the lift force and the centrifugal force. Lift force depends on the local flow velocity and local angle of attack. Centrifugal force acts outward while pulling the blade from the hub and it increases with the RPM. This force provides rigidity to the rotor and it is useful while it is carrying the weight of the helicopter. As the rotorcraft develops lift during takeoff or flight, the result of these two forces makes the blades rotate in a cone shape which is

unsurprisingly called coning. Excessive coning results in a decrease in lift due to lowering of the effective disk area and can result in even structural damage.

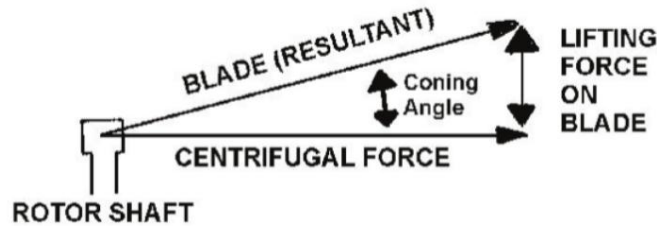


Figure 3.3 Effects of the lift and centrifugal force on coning angle [34]

3.1.2.3. Flapping The up and down movement of rotor blades about a hinge is called flapping. In the hovering rotor, each rotor blade experiences a similar aerodynamic field. That results in the lift force on the rotor disk being azimuthally axisymmetric. The blades will form a steady state balance between centrifugal forces and aerodynamic forces by coning up. When the rotorcraft starts to move forward, the dissymmetry of lift in rotor disk occurs. The flapping is the primary means of compensating the dissymmetry of lift [35]. Letting the blade freely move up and down allows local changes in the angle of attack. The azimuthal change in aerodynamic loads is compensated by the flapping motion and local AoA changes [32]. The flapping becomes periodic in any steady state of the rotor, and therefore can be expressed in Fourier series [36].

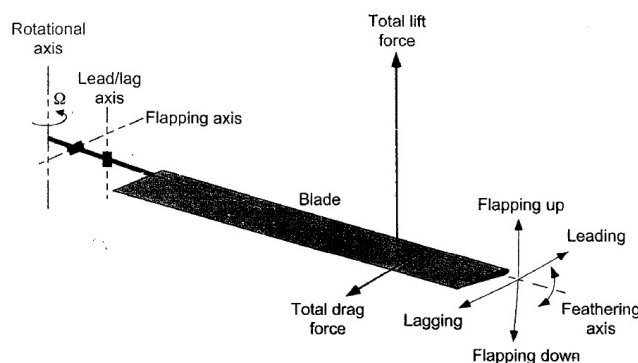


Figure 3.4 lead-lag, flapping and feathering motion shown on a rotor blade [32]

3.1.2.4. Solidity Another key term for rotor aerodynamics is solidity. Solidity is defined as the ratio of the area of the blades to the area of the whole rotor. For a constant chord blade, N being the number of blades, the solidity is calculated by the following formula:

$$\sigma = \frac{\text{blade area}}{\text{disc area}} = \frac{NcR}{\pi R^2} = \frac{Nc}{\pi R} \quad (11)$$

3.2. The Virtual Blade Model

The complexity of helicopter aerodynamics became a stimulus for the advanced computational techniques in the fluid dynamics domain. The virtual blade model is one of the tools developed for simulating and understanding this complex phenomenon. It was first proposed by Rajagopalan et al. [37]. In this method, the geometry of the rotors is not directly modeled in the computational mesh. Its propulsive influence that changes the flow's vector momentum is modeled. This is done implicitly by adding source terms in the governing momentum equation at discrete points through which the rotor blade passes. Therefore, the aerodynamic effects of the rotor are accounted for without the rotor being present in the discretized mesh domain.

For a reliable simulation of helicopter aerodynamics, the flowfield around the rotor also should be correctly simulated. Especially the vortical wake is important because of its influence on the helicopter vibration, aeroelastic stability, noise, gust response, handling qualities, and flight dynamics [38]. Because the body-fitted methods require immense computational expense, using prescribed wake models have been used. Using wake models is highly restrictive because the nature of the wake would significantly differ for other rotor configurations, flight conditions, advance ratio, etc. The need for a first-principles calculation of wake formation is identified by Johnson [39]. Performing a Navier-Stokes solution with the VBM eliminates these disadvantages. Wake is properly taken into account by the governing equations and develops as part of the solution without any modeling near or the far wake [6]. This allows the simulation of body-rotor interactions for different rotor geometries and flight conditions.

No assumptions about the wake structures are needed with this method. Also, a body-fitted mesh for the rotor is not needed which results in a lower computational expense. Although it has many advantages, there are some inherent limitations present for this method. For example, some aerodynamic aspects of the flow, like boundary layer separation, can be assessed indirectly because the chordwise flow over the blade is not directly simulated [38]. The source terms are unknown at the start of the iterations. They evolve with the flowfield, as a part of the total solution. Magnitudes of the source terms are calculated according to rotor geometry, cross-sectional blade aerodynamic characteristics, and the local flow properties [6]. Source terms are calculated from the rotor force and moments, which are calculated according to the Blade Element Momentum Theory.

3.2.1. Blade Element Momentum Theory

Blade Element Momentum (BEM) theory was first introduced by C.N.H. Lock et al. [40] in 1925. Originally it was presented as a tool to study the performance of propellers. Later in 1926, its modern form is formulated by H. Glauert [41]. Later in 1946, it was proposed to be used in helicopter rotor by Gustafon Gessow [32] and since then it has been used to assess the performance of propellers, rotors, and turbines.

BEM is a hybrid model and formed by integrating two theories; the Blade Element Theory (BET) and the Momentum Theory.

William Froude [42] introduced BET in 1878. In his work, he cut the turbine blade into sections, blade elements, and used them to calculate local forces according to the flow characteristics and blade shape. For each blade element, the local angle of attack is calculated according to the flow and linear velocity of the element. This information is used with lift and drag coefficients to account for the sectional thrust and torque of the element. Then all results are integrated to calculate global values. The BET ignores the velocity contribution in the radial direction [43].

The momentum theory (also known as actuator disc theory) is the oldest mathematical model of a propeller and dates back to 1865. This theory is not interested in the local flow through blades but adopts a macroscopic view of the rotor by modeling the behavior of the fluid column passing through the rotor [44]. The rotor is replaced with an infinitely thin, permeable disc in which the load is replaced by a pressure distribution [45] and adopts a more macroscopic approach to model the behavior of a column of fluid passing through a turbine. This method was introduced by Rankine [46] and later it is used by Joukowsky [45] and Betz [47] to formulate the Betz-Joukowsky limit. The disadvantages of this method are, its inability to account for the flow separation of complex 3D phenomena (tip vortices, shocks, etc.) [43]. Although it is computationally more affordable than simulating whole rotor geometry, only time-averaged effects of the entire rotor will be captured. The aerodynamic

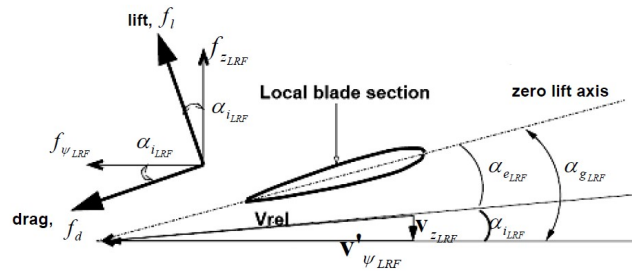


Figure 3.5 Aerodynamic forces and local fluid velocity component on an airfoil [32]

forces on a 2D blade element can be seen from the Figure 3.5. In the figure, V'_ψ is the linear velocity formed due to the rotation of the blades and the movement direction of the helicopter. V_z is the induced velocity, that's created by the pressure difference between the rotor upstream and downstream (wake).

The angle of attack α_g is formed between the V_z and the blade cord. The induced AoA is formed due to the induced axial flow of V_ψ and V_z . It is calculated by the following formula:

$$\alpha_{iLRF} = \tan^{-1} \left(\frac{V_{zLRF}}{V'_{\psi LRf}} \right) \quad (12)$$

The aerodynamic forces on the blade element are calculated according to the effective AoA which is calculated by:

$$\alpha_{eLRF} = \alpha_{gLRF} - \alpha_{iLRF} \quad (13)$$

For each blade element, the effective angle of attack (AoA) and a predefined lookup table of lift and drag coefficients is used to calculate the aerodynamic forces. The lookup table is prepared by the user and it can be organized according to 2-D lifting line theory or airfoil experimental data. The lookup table is a numerical representation of the airfoil lift (Figure

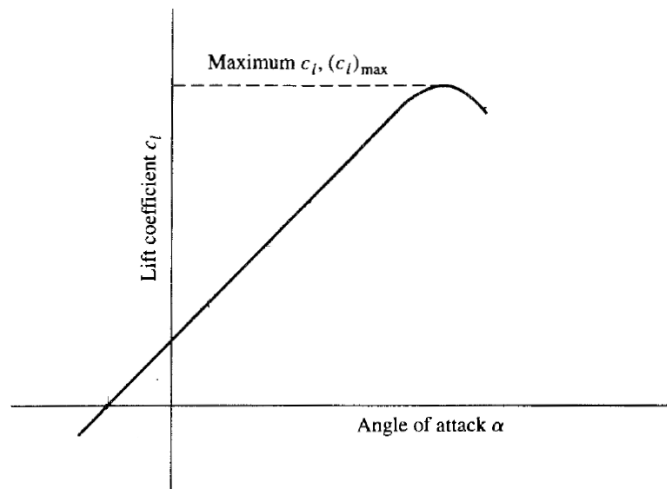


Figure 3.6 Sketch of a generic lift coefficient curve taken from [48]

3.6) and drag (Figure 3.7) coefficients. In which the corresponding coefficients are calculated according to the blade section angle of attack which is shown by α on the curves.

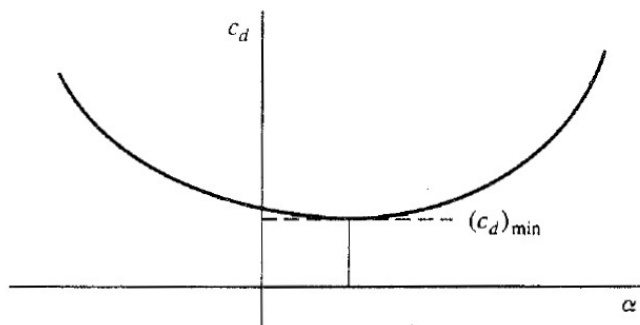


Figure 3.7 Sketch of a generic drag coefficient curve taken from [48]

As can be understood from the process, forces depend on effective AoA and this method neglects compressibility effects that are present at high Reynolds numbers.

Following equations describe the local velocity and the resulting aerodynamic forces for each blade element:

$$U_{local} = \sqrt{(V_{z_{LRF}})^2 + (V'_{\psi_{LRF}})^2} \quad (14)$$

$$f_l = \frac{1}{2} \rho (U_{local})^2 c C_l \quad (15)$$

$$f_d = \frac{1}{2} \rho (U_{local})^2 c C_d \quad (16)$$

where;

C_l and C_d are the aerodynamic force coefficients for lift and drag,

c is the chord length of the specific blade element,

U_{local} is the local velocity of fluid felt by the blade element.

As discussed before, BEMT does not take the 3D flow effects into account. There is one exception to that, the loss of lift near the tip of the rotor due to tip vortices. This phenomena is often called *tip loss* and represents a deviation from the ideal case, in which no tip vortex is present. VBM accounts for this effect by applying a correction factor for the elements near the blade tip. For a non-dimensionalized ratio between 0 and 1, where 0 represents the rotor center and 1 represents the blade tip, after a threshold value the lift forces (Equation 15) for the blade elements does not taken into consideration. Only the drag (Equation 16) is included in the calculations. For helicopter rotors, this factor is found to be between 0.95 to 0.98 [32].

Now that sectional forces are obtained for each element, these forces are converted to forces normal and parallel to the rotor disk plane. After that, all elemental thrust, torque and power on each blade element are calculated using the following equations:

$$dT = N_b f_{Z_{LRF}} dr \quad (17)$$

$$dQ = N_b f_{\psi_{LRF}} r dr \quad (18)$$

$$dP = \omega dQ = N_b f_{\psi_{LRF}} \omega . r dr \quad (19)$$

where;

T , Q and P represents thrust, torque and power respectively,

ω represents angular velocity,

N_b represents the number of rotor blades.

Total thrust, torque and power are found by integrating elemental values starting from the end of the hub to the blade tip.

Although there is no direct relation between the blade element and the rotor disk cell, because the rotor disk is modeled with structured hexahedral meshes, the same approach is applied to calculate forces in each cell [49]. After the forces are calculated, they can be introduced to the flow as momentum sources after a modification in rotor computational cells.

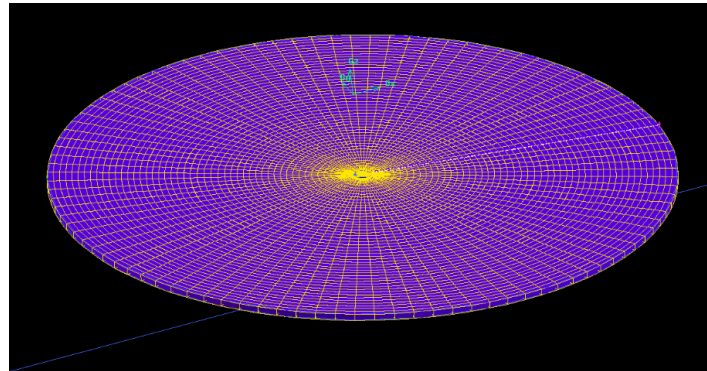


Figure 3.8 Structured hexahedral mesh, suitable to be used with the VBM

Assuming a steady-state analysis such as RANS, and assuming a constant rotor rotational speed, the time-averaged force for each cell becomes only a fraction of instantaneous forces and can be calculated by geometric averaging over an angle of 2π [49]. The scaling factor S is needed for this and it is basically used to spread out the instantaneous forces over the

entire rotor diameter:

$$S = N_b \delta r \frac{r \delta \psi}{2\pi r} \quad (20)$$

$\frac{r \delta \psi}{2\pi r}$ defines the ratio between the arc length of a blade element to the circumference of the rotor disk.

When this factor is applied to the instantaneous forces, the time-averaged forces for rotor cells are found:

$$F_{z_{LRF}} = f_{z_{LRF}} \cdot S = f_{z_{LRF}} N_b \frac{\delta r \cdot r \delta \psi}{2\pi r} \quad (21)$$

$$F_{\psi_{LRF}} = f_{\psi_{LRF}} \cdot S = f_{\psi_{LRF}} N_b \frac{\delta r \cdot r \delta \psi}{2\pi r} \quad (22)$$

The term $\delta r \cdot r \delta \psi$ is the equivalent of the sectional blade element area dA when a structured hexahedral mesh is used. This term is the main reason for limiting the cell type of rotor disk only to the structured hexahedral meshes [49].

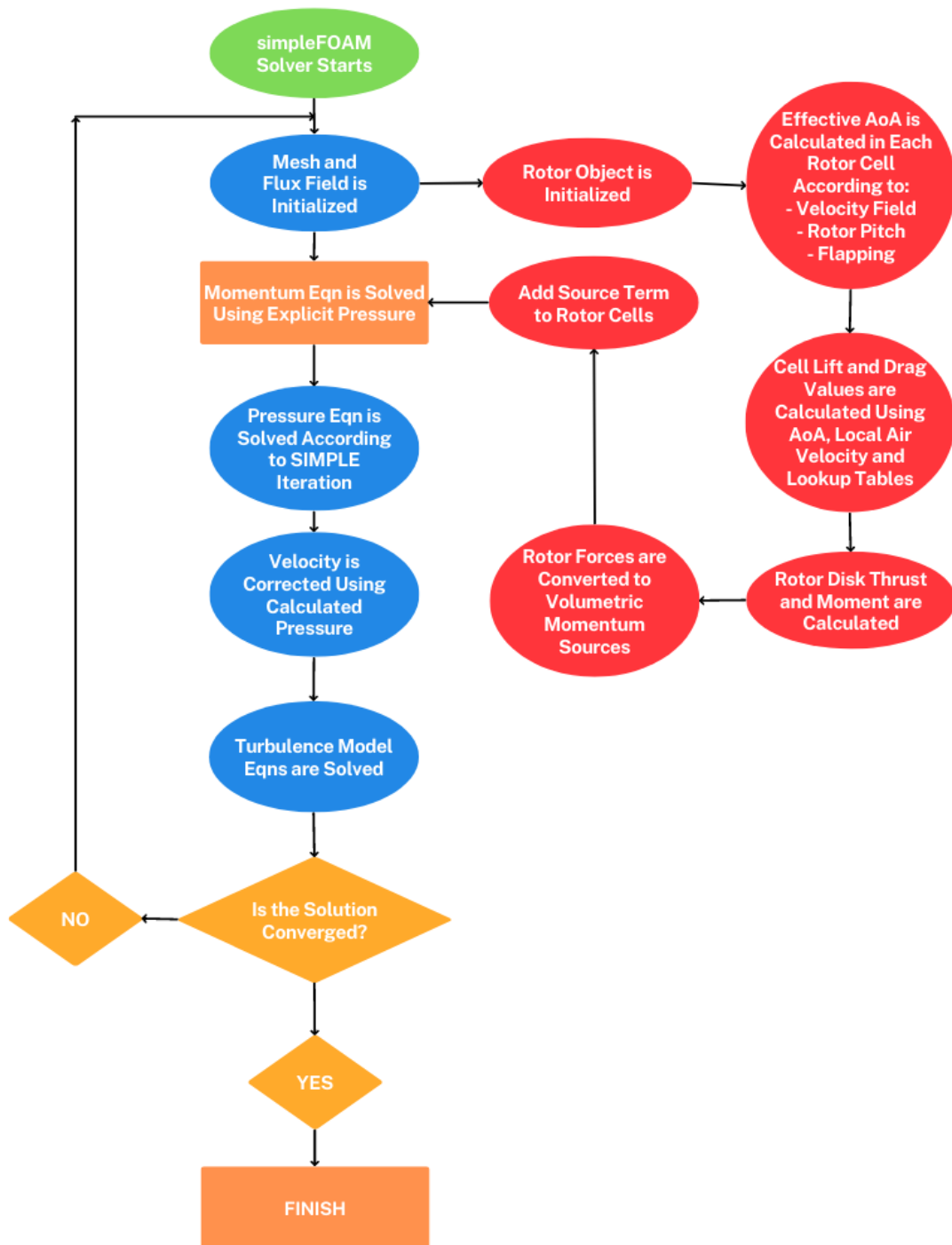


Figure 3.9 Integration of VBM with the simpleFoam solver

The final step is to convert the cell forces into volumetric momentum sources by:

$$(\vec{S}_U)_{cell} = \frac{1}{V_{cell}} x(\vec{F})_{cell} \quad (23)$$

Where the V_{cell} is the volume of the cell and F_{cell} is the calculated force for each cell in previous steps.

The interaction of the VBM with the solver simpleFoam can be seen in Figure 3.9.

This concludes the calculation of the rotor forces and momentum source values.

3.3. IBM in OpenFOAM

OpenFOAM, short for "Open-source Field Operation And Manipulation", is a collection of C++ CFD codes consisting of numerical solvers and pre/post-processing utilities. It is an open-source software that's being released since 2004. Although it is a mature and capable platform its source code is easily modifiable. These qualities make it an attractive platform for researchers. It has three main variants; ESI Group variant, foam-extend variant and OpenFOAM Foundation variant. In this study, the last one was used.

At the date of writing, there are three applications of IBM within OpenFOAM. All of them work according to the discrete forcing approach which was explained in Chapter 2. They only differ in the way they define the boundaries; direct versus indirect.

The first one is an indirect boundary imposition discrete forcing approach by Constant et al. [50] which is a modified version of the work by Pinelli et al. [51]. The goal of this method is to offer numerical stability, accuracy, and computational efficiency, especially for the analysis of deformable and/or moving geometries. For this purpose, IBM is integrated into the Pressure-Implicit Split-Operator (PISO) solver. Besides, a new technique called Reproducing Kernel Particle Method was used to interpolate the velocity field onto a boundary and the spreading of the force field from the IB to neighboring mesh points. The details of this method can be found in references [50–52]. Constant et al. released the code online [53], which is compatible with OpenFOAM version 2.2 (ESI-OpenCFD).

The second IBM within OpenFOAM is a direct boundary discrete forcing approach and is developed by Jasak et al. [54]. This code is a part of foam-extend since version 4.0. This method is designed to allow the boundaries to be represented sharply and also allow high Re flow analysis. The first goal is achieved by direct imposition of the boundaries. In which the values inside the IB are not calculated and values near the IB are calculated by quadratic interpolation. Values for the individual IB cells are interpolated from the wall boundary condition and the neighboring cells. The second goal is achieved by hosting an IB specific wall function. The sampling point is determined as 1.5 times farther from the wall compared to the first cell center as seen in Figure 3.10, and the values at this point are evaluated according to near non-IB cell values and distance from the wall. Finally, the wall function sets the velocity gradient at the wall, using the velocity component tangential to the wall [54].

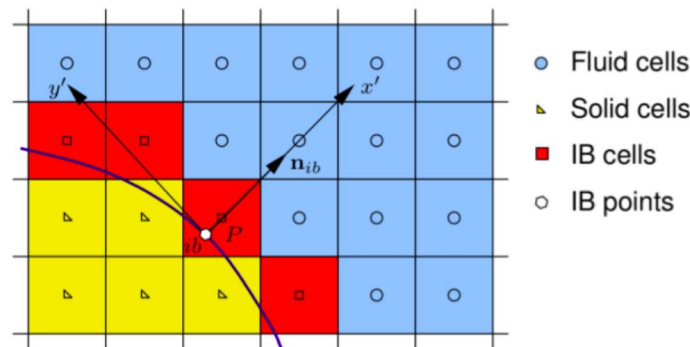


Figure 3.10 Interpolation of the IB cells by Jasak et al. [54]

The third IBM within OpenFOAM is the one that's used in this study and will be explained thoroughly in the next subsection.

3.4. IBM with y^+ Adaptive Wall Function

Xu and Liu [55] observed that for most of the IB studies the focus is on the correct estimation of pressure forces and the correct calculation of wall shear or viscous forces is given less importance. They also observed that the IBM implementation of Jasak et al. [54] results in nonuniform and nonphysical wall shear stress distribution while failing to produce smooth

wall shear stress with existing wall functions. This is caused by two reasons. Some of the Reynolds-averaged Navier Stokes turbulence models like $k-\epsilon$ are sensitive to the placement of cells next to the wall. For correct modeling of the near wall effects, the first layer of cells should be placed at the log-law layer ($y^+ > 30$), not at the viscous sub-layer or buffer layer ($y^+ < 30$) because all of these layers has fundamental differences and the $k-\epsilon$ turbulence model is able to realistically model the near wall phenomena only if the first layer is placed in the log-law region. While using the IBM, wall and cell distances may dynamically change and disobey the y^+ requirement. If the wall distance requirement has not been met, it is observed that [55] the wall shear values diverge significantly from the real value.

The work of Xu and Liu is a discrete forcing approach with direct boundary imposition and it is based on the IBM applied by Jasak et al. [54]. It differs from the other codes by focusing on the smooth and correct modeling of wall shear stress. The correct modeling of wall shear also means the correct modeling of boundary layer and viscous forces.

3.4.0.1. Representation of Immersed Boundaries Immersed boundary representation is similar to the other IB methods in the literature. The computational cells are classified as; IB cells, live cells, or dead cells and can be seen in Figure 3.11. The cells that are in contact with the immersed boundary are called IB cells and are used to enforce the boundary conditions. A “hit point” which is the normal projection of the IB cell center to the IB is calculated and marked for each cell. Then, the “image point” for each cell is determined according to the distance between the hit point and IB cell center.

These points are used to reconstruct the flow variables for imposing the boundary conditions. The reconstruction is made according to an interpolation stencil and stencil building is done according to the following steps:

1. The cell hosting the image point, called the image cell, is searched and selected.
2. The IB and live cells that share a face with the image cell are selected. These cells are called first-row cells.

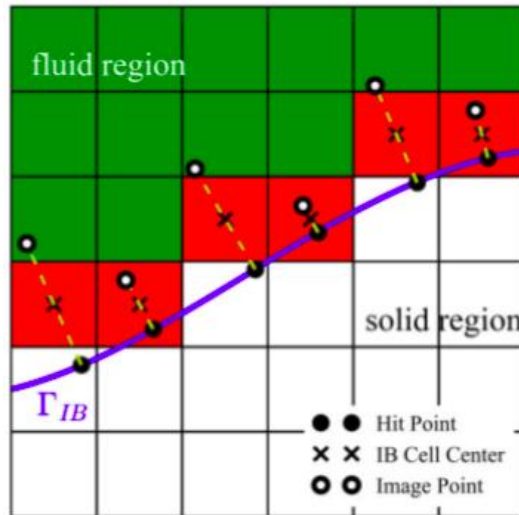


Figure 3.11 Representation of Immersed Boundary, IB cell centers, hit points, image points and immersed interface Γ_{IB} by Xu and Liu [55]

3. The same neighbor selecting process is applied for each of the first-row cells. The newly selected cells are called second-row cells. Second-row cells can not include cells that are already in the first-row cells.
4. The second step is repeated until the total number of the neighboring cells for each IB cell is above the interpolation scheme-dependent threshold value.
5. All the neighboring cells are sorted according to their distance to the image point.

The reconstruction is made according to the image point instead of immersed cell center because of the immensely lower computational cost, according to Xu and Liu [55].

3.4.0.2. y^+ Adaptive Immersed Wall Model For the body conformal meshes, the first cell height from the wall can be held constant with prism elements. For the IB meshes, the non-dimensional distance from the wall y^+ value can not be controlled with the conventional methods. It is very likely that some of the IB cells will be placed at the buffer layer or viscous sublayer, resulting in irregular wall shear stress values.

The wall adaptive algorithm of Xu and Liu depends on the idea of choosing a new IB cell when the wall distance is unacceptably small. In that case, the initial IB cell is marked as a

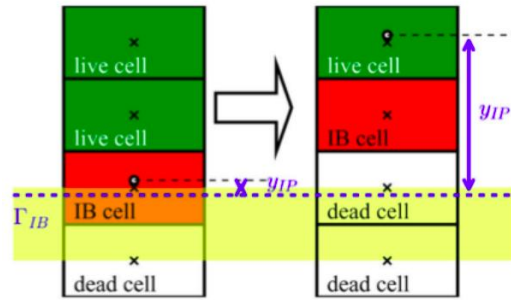


Figure 3.12 A simplified representation of the y^+ adaptation process by Xu and Liu [55]

dead cell and a new neighboring cell with a suitable wall distance is found and set as the new IB cell as seen in Figure 3.12. This is repeated at every iteration step until all elements are located in the log-law layer.

The verification and validation of this approach were made with the 2-D Pitz-Daily case [56] and turbulent flat plate case from the NPARC [57] CFD verification and validation resources. The method works as intended and creates a more realistic wall shear stress distribution along the boundary. Additionally, in the 2D Pitz-Daily case, the adaptive wall function outperformed the previous IB wall function and produced results very close to the body-fitted mesh.

Even though it has apparent advantages while modeling the near wall phenomena, there is a slight compromise in the sharp representation of the immersed boundary. According to the authors [55], this disadvantage can be minimalized with the sufficient refinement of mesh near the boundary.

The y^+ adaptive IBM is advantageous when the near wall modeling and viscous forces are important. For example, aerospace applications. The VBM does not take into account the y^+ or the viscous forces but for the airframe, correct modeling of the wall shear is important while computing the drag force.

The authors released the code online [58] and it is compatible with OpenFOAM 3.0.1 (OpenFOAM Foundation). This is the code that's used in the CFD analysis of the verification and validation case included in this study.

3.5. Governing Equations

The flow modeled in this study can be summarized as; steady, incompressible and turbulent. In this subsection, the governing equations related with these flow characteristics will be shared.

It wouldn't be wrong if it is said that the Navier-Stokes Equations are the cornerstone equations of aerodynamics. Although it doesn't look intimidating, it is a nonlinear, second-order, partial differential equation and an analytical solution is only obtainable for a handful of simple cases [59].

For a Newtonian, incompressible, isothermal flow, the equations in Cartesian coordinates are written in the following way.

Incompressible continuity equation:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (24)$$

x-component of the incompressible Navier–Stokes equation:

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) = -\frac{\partial P}{\partial x} + \rho g_x + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \quad (25)$$

y-component of the incompressible Navier–Stokes equation:

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) = -\frac{\partial P}{\partial y} + \rho g_y + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) \quad (26)$$

z-component of the incompressible Navier–Stokes equation:

$$\rho\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) = -\frac{\partial P}{\partial z} + \rho g_z + \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) \quad (27)$$

where x , y and z represents the directions in the Cartesian coordinate system. The u , v and w are the components of the velocity vector in x , y and z directions respectively. P represents pressure, μ represents viscosity, ρ represents density of the fluid, and g represents gravitational acceleration.

Even for the steady flow, turbulence is difficult to simulate. The reason is that turbulent flow is always inherently unstable and includes swirling random vortical structures called turbulent eddies [59]. These structures operate in a wide range of scales and in all directions. The size and time scale difference is so huge, solving these equations on a grid that fully resolves the eddies requires immense CPU time. It is highly unpractical and unfeasible. Therefore, some simplifying assumptions are used while keeping in mind their limitations.

Turbulence models are used to remove the need of resolving all the different scale turbulent eddies while being able to model turbulent flows. The $k - \epsilon$ is such a model.

Reynolds-averaged Navier-Stokes (RANS) equation is used instead of Equations 25-27 when a turbulence model is used. For a steady, incompressible, turbulent flow RANS is:

$$(\vec{V} \cdot \vec{\nabla}) \vec{V} = -\frac{1}{\rho} \vec{\nabla} P' + \nu \nabla^2 \vec{V} + \vec{\nabla} \cdot (\tau_{ij,turbulent}) \quad (28)$$

The additional term $\tau_{ij,turbulent}$ is called the specific Reynolds stress tensor,

$$\tau_{ij,turbulent} = - \begin{pmatrix} \overline{u'^2} & \overline{u'v'} & \overline{u'w'} \\ \overline{u'v'} & \overline{v'^2} & \overline{v'w'} \\ \overline{u'w'} & \overline{v'w'} & \overline{w'^2} \end{pmatrix} \quad (29)$$

where the prime sign denotes the instantaneous (fluctuating) velocity components and the overline indicates the time average of these components. This matrix introduces additional unknowns and to be able to solve these additional turbulence components, some closure equations are needed. The $k - \epsilon$ model is used at this point.

According to Launder and Spalding [60], k represents the turbulence energy and is defined as in Equation 30,

$$k \equiv \frac{1}{2} \overline{u_i u_i} \quad (30)$$

where the i subscript denotes the Cartesian coordinate directions. The other component, ϵ represents turbulence energy dissipation rate and is formulated as in Equation 31.

$$\epsilon = \nu \overline{\frac{\partial u_i \partial u_i}{\partial x_k \partial x_k}} \quad (31)$$

The ν is the kinematic viscosity and the k subscript denotes the Cartesian coordinate directions.

This chapter introduced the application details of the VBM and $y+$ adaptive IBM. After that, the governing equations of the flow are shared. The next chapter introduces the details of the demonstration case.

4. DEMONSTRATION: GEORGIA INSTITUTE OF TECHNOLOGY AIRFRAME

In the literature, there are a number of CFD codes that have been developed to address complicated rotorcraft aerodynamics problems. Some extensively studied rotorcraft experiments have been used for the verification and validation of these codes. GIT rotor-airframe interaction study [5] is one of the popular ones. According to Park [15], out of the 24 rotorcraft CFD code studies in the literature, 11 have used the GIT rotor-airframe interaction study. This popularity can be attributed to the easy-to-model plain geometry and to the extent of the experimental data available. Steady and unsteady measurements as well as particle-image velocimetry data are available.

The GIT experiment has been chosen for the verification and validation of the VBM-IBM application because it is one of the well-understood and extensively studied cases on rotor-airframe interactions. It also allows comparing the IB method against the body-conformal approaches.

This chapter provides detailed information about the GIT study and the modeling approach. The original experiment was done on the John J. Harper wind tunnel by Brand [5] which is located at the Georgia Institute of Technology. The experiment aims to understand the interaction of a rotor in forward flight with a relatively simple airframe.

The computational setup is designed to be consistent with the tunnel experiment. In the following sections, details of each will be explained.

4.1. GIT Experimental Setup

The experimental setup includes the placement of a simple airframe in the low-speed wind tunnel under a uniform freestream velocity. The wind tunnel cross section was 2.74 m x 2.13 m. To assure the flow uniformity and low turbulence intensity, a honeycomb, and a

turbulence damping screen were integrated into the tunnel [5] before conducting the tests. The turbulence intensity was subsequently measured to be below %1.

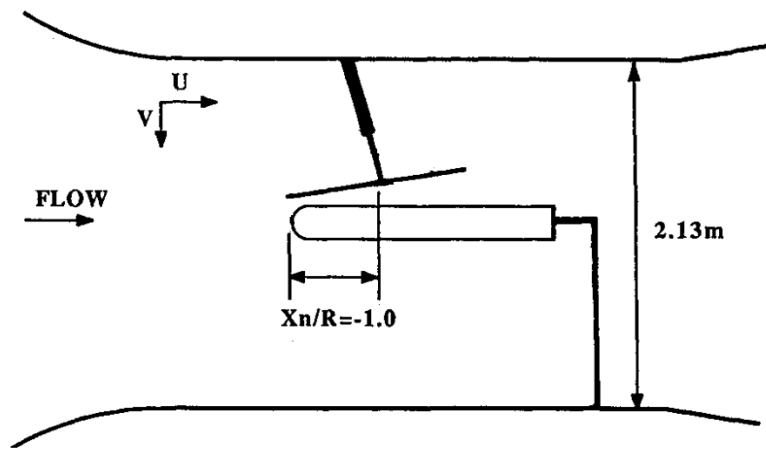


Figure 4.1 Side view of airframe installation with respect to wind tunnel [61]

The airframe used in the tests was a 1350 mm long cylinder with a hemispherical nose of 134 mm diameter. Its placement in the tunnel can be seen in Figure 4.1 and Figure 4.2. It was built from hollow segments of aluminum that allowed the pressure taps to be inserted along the frame. There were a total of 53 static pressure taps along a longitudinal line on the fuselage. The body geometry was axisymmetric and was free to be rotated along the longitudinal axis when needed. That allowed taking measurements from top, bottom and sides of the airframe without increasing the number of pressure taps. The airframe had zero angle of attack and was aligned with the freestream.

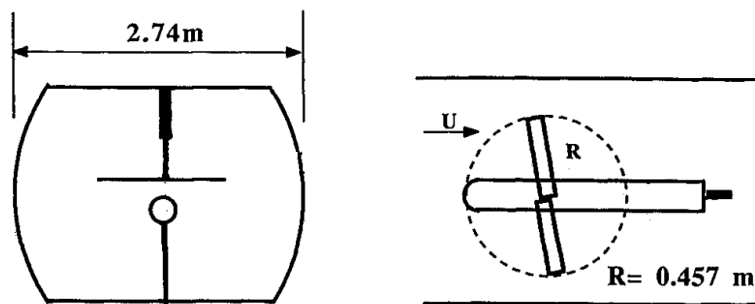


Figure 4.2 Upstream and top view of airframe and rotor [61]

The airframe and the rotor had no physical connection. The rotor had a teetering design that allowed the rotor shaft to be tilted as necessary. According to Brand, the teetering design

also allowed to minimize the radius of the rotor hub, which has been observed [62] to be able to generate considerable interference between the rotor and the airframe. The cyclic and collective pitch control mechanism was eliminated to reduce the size of the rotor hub further. These design choices led to a compact hub size of only 2.7% of the rotor radius. The GIT rotor consists of two untwisted rectangular blades with the NACA0015 symmetric airfoil. They were designed with a constant collective pitch of 10 degrees.

Table 4.1 Rotor parameters of the GIT experiment

Parameter	Value
Number of Blades	2
Rotor Diameter	917 mm
Hub Diameter	44.5 mm
Twist	0°
Taper Ratio	1
Shaft Angle	-6°
Collective Pitch Angle	10°
Long. Flapping Angle	1.94°
Lat. Flapping Angle	2.03°

The blades were produced from rigid materials and the hub was designed to be hingeless to prevent elastic blade motions or coning. Yet, blades were unrestricted to flap laterally and longitudinally. For simulating the forward flight, the rotor shaft was tilted 6 degrees forward. The rotor diameter was 914 mm and it was operated at a fixed value of 2100 rpm, regardless of the freestream velocity. The freestream velocity was chosen as 10 m/s according to the 0.1 *advance ratio* (which is defined as the ratio of freestream velocity to the tip speed). The rotor parameters are shared in Table 4.1.

The hub center was located 457 mm downstream and 137.1 mm above the nose of the airframe. A summary of the airframe geometry and placement parameters for the experiment can be seen in Table 4.2

Table 4.2 Wind tunnel operation and airframe parameters

Parameter	Value
Inlet velocity	10 m/s
Rotor RPM	2100
Advance ratio	0.1
Airframe length	1350 mm
Airframe diameter	134 mm
Airframe angle of attack	0°
Airframe sideslip angle	0°
Rotor center location, downstream	457 mm
Rotor vertical location	137.1 mm

4.2. Computational Setup

For the CFD analysis of the GIT rotor-airframe study, CAD geometry was prepared using Gambit and Tgrid. For the meshing, both Gambit and snappyHexMesh were used. Finally, solving the case and post-processing were done using OpenFOAM 3.0.1 and ParaView, respectively. The computational setup is prepared to be in line with the experimental setup. In this part, starting from the geometry preparation, the method followed for the case will be shared.

4.2.1. Geometry Preparation

The computational domain was chosen to be consistent with the wind tunnel experiment. The cross-section has the dimensions of $[-1.37 \text{ m}, 1.37 \text{ m}] \times [-1.065 \text{ m}, 1.065 \text{ m}]$ in the y and z directions. A general view of the domain can be seen in Figure 4.3. There

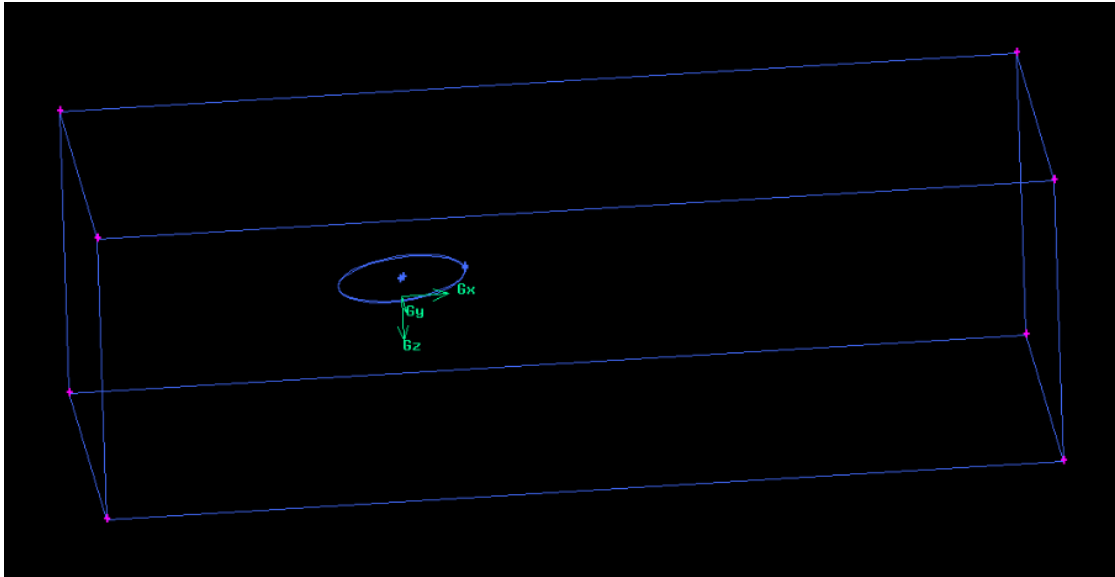


Figure 4.3 General view of the domain and the reference axis origin

is no information provided about the length of the test section in the experiment but the computational dimensions were chosen according to best practice guidelines. The length of the domain in the x -axis was chosen as $[-5R, 10R]$ to have adequate upstream and downstream length. The longitudinal dimensions are $[-2.285 \text{ m}, 4.57 \text{ m}]$ accordingly. Unlike the experiment, the domain shape was selected as a rectangular prism. This choice was made for a few reasons. The wall effect of the tunnel was negligible [5] and the geometric details of the test section shape were absent in the original experiment. Additionally, previous studies in literature [15] indicated no negative effects related to this choice.

The rotor was modeled with a radius of 457 mm. The hub was left as a hole in the center with a radius of 122.5 mm. This choice was made according to two reasons. The center of the rotor is a place where no aerodynamic force exists and the VBM has no inherent ability to distinguish the hub and blades. If the disk was modeled as a whole disk, the center mesh

elements would have been low-quality elements influencing the numerical stability of the analysis. Also, no negative effect of this choice was observed in the previous studies [49].

The domain geometry and the rotor geometry were drawn and meshed separately.

4.2.2. Meshing Strategy

The rotor and the domain were modeled in Gambit. As it was explained before, correct rotor modeling requires one cell thick structured hexahedral mesh. This structured mesh should have an adequate number of elements both in radial and azimuthal directions to model blade section forces accurately. Such requirements result in varying sizes of high aspect ratio mesh elements. This limitation also affects the mesh number, size, and shape around the rotor while proving to be hard to create a high-quality domain mesh. Furthermore, the vicinity of immersed geometry should be refined for representing the boundaries sharply. The conflicting requirements of the domain and rotor mesh led this study to use nonconformal mesh. By using nonconformal mesh, each region was meshed according to its specific needs.

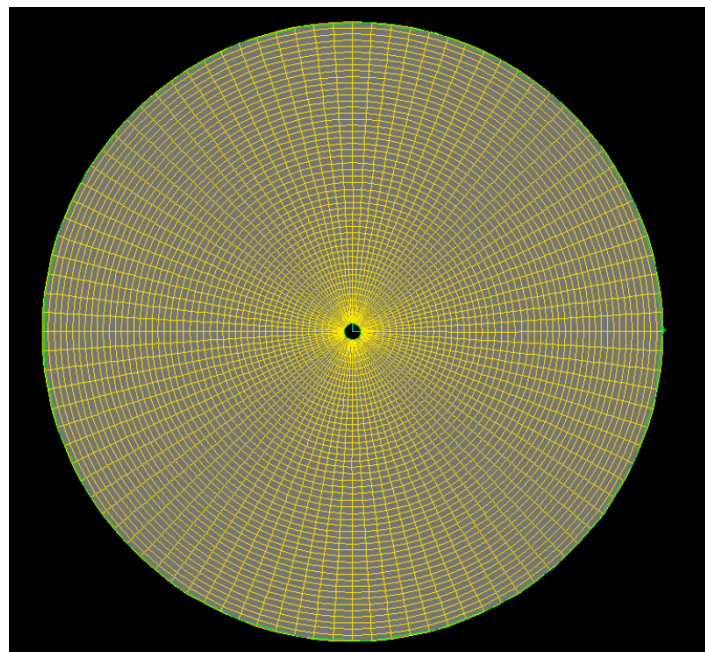


Figure 4.4 A rotor disk that has structured hexahedral mesh and is suitable to be used for VBM

The rotor was meshed by Gambit and it has 29400 elements. A sample rotor mesh can be seen in Figure 4.4.

The domain was meshed using the standard OpenFOAM meshing tool that's called snappyHexMesh. This tool generates hybrid mesh (hexahedral and split-hexahedral) for domain meshing and also allows refinement of the mesh. Refinement can be done according to the CAD model in the domain or by selecting a specific region.

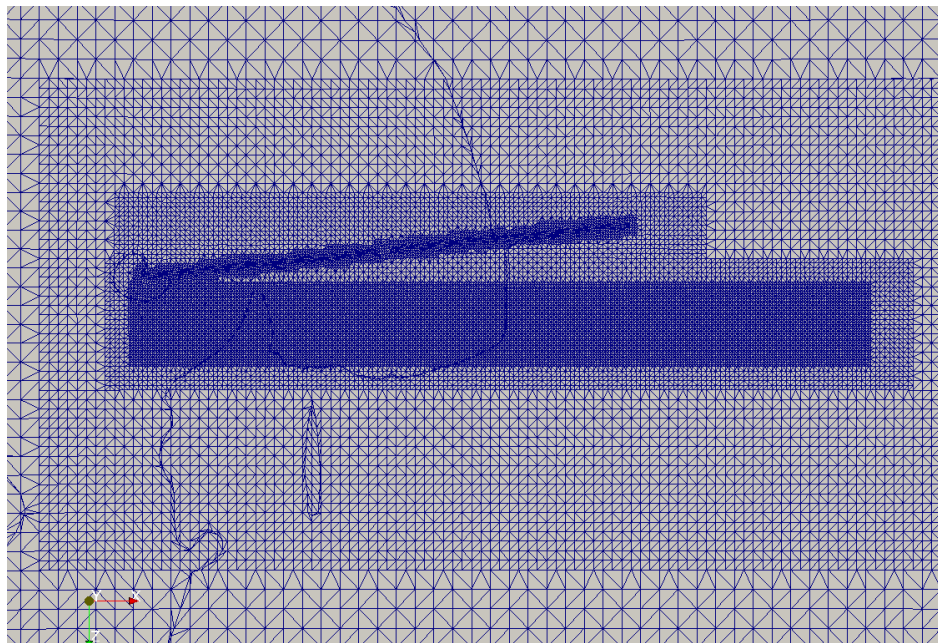


Figure 4.5 Cross section of domain mesh and refinement areas. Due to a rendering problem, some irregular lines are visible that are not present in the mesh itself.

For this study, four levels of refinement were applied to the mesh. The finest refinement was applied to two regions. The first region is the boundaries of the airframe with the aim of correct representation of the immersed edges. The second region is the vicinity around the rotor. The reason was to decrease the numerical diffusion that may happen due to the nonconformal mesh face between rotor and domain. A lower number of elements than the rotor elements in the domain side would have failed to capture the momentum source gradients between different blade elements and would have caused an unphysical averaging effect. The refinement was done to have an adequate number of elements around the rotor for resolving the rotor aerodynamical forces correctly.

The rotor upstream and downstream were refined to capture pressure gradients. The front and back of the airframe were refined to capture stagnation and flow separation points. Final result can be seen in Figure 4.5. When the meshing process of both regions was completed individually and mesh quality was found to be satisfactory, meshes were combined with the `mergeMeshes` command.

4.2.2.1. Defining the Nonconformal Boundary Condition Initially, a conformal unstructured tetrahedral mesh was prepared for the domain in Gambit. The pairing between the structured hex-mesh rotor and the unstructured tetrahedral domain was made possible by pyramidal elements. Although this process was faster, it is observed that as the number of rotor mesh elements increased, the mesh element quality in the vicinity of the rotor got lower. The result was, that even though the nonconformal correction was applied in the solver, the case got stability problems. Another observed disadvantage was that flow alignment of mesh elements was not possible to obtain. Nonconformal meshing is free from all of these disadvantages and allows the use of a high-quality structured Cartesian grid. The Arbitrary

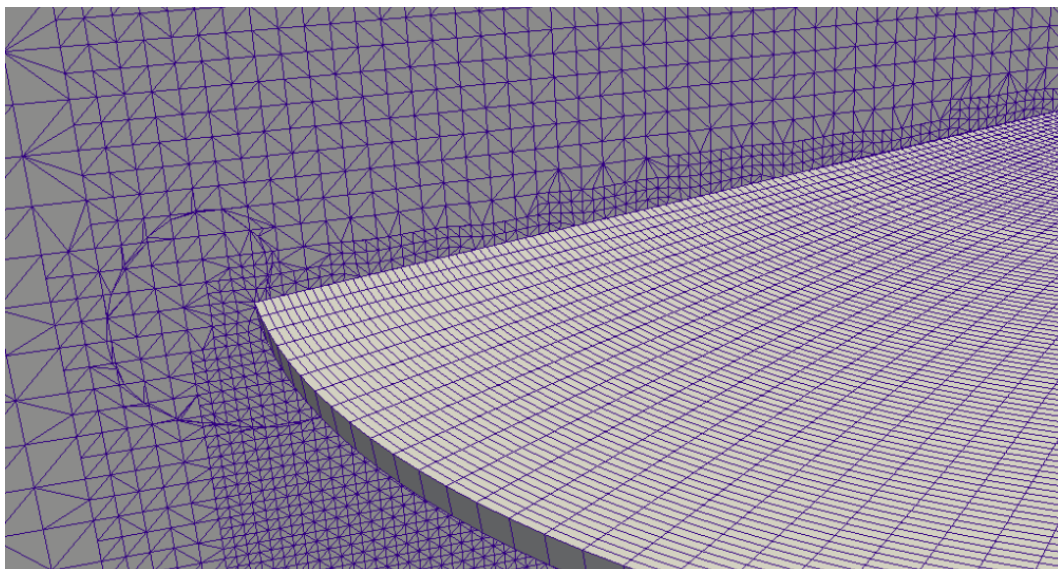


Figure 4.6 Nonconformal rotor and domain mesh closeup

Mesh Interface (AMI) is the name of the nonconformal mesh application in OpenFOAM. The `cyclicAMI` boundary condition in OpenFOAM allows adjacent and disconnected mesh

domains to be used together. It was presented in OpenFOAM v2.1.0 and since then, found to be especially useful for simulating rotating geometries [63]. The VBM method does not require the rotor mesh to be rotated and in this study, AMI was used to connect two fixed mesh regions by a nonconformal interface.

4.2.3. Computational Setup

The closed source CFD codes aim to simplify the process and decrease the user input to eliminate human error. The curse and blessing of an open source CFD code is that the user has an overwhelming amount of control and should make an elevated number of decisions while preparing the case. With great power comes great responsibility. These educated choices could make or break the case, easily.

For this case, inlet and outlet boundary conditions were chosen as; velocity inlet and pressure outlet, respectively. The advance ratio in the experiment was 0.1 and the corresponding inlet velocity is 10 m/s. Walls were defined as no-slip. The airframe was defined as an immersed boundary.

Reynolds number is the ratio of the inertial forces to viscous forces and is a dimensionless quantity. It is used to understand if the flow is laminar or turbulent [59]. The threshold value from laminar flow to turbulent flow is called the critical Reynolds number and is 500,000 for external flows.

The Reynolds number for this case is found to be 905,000, which is estimated according to rotor tip speed and blade chord. Although it is higher than the critical Reynolds number, the transition from laminar to turbulent is not worrisome for the rotor. Let us not forget that the VBM does not solve turbulence or viscous forces on the blades. It merely models the time averaged blade effect as a volumetric source. Therefore, the transition on the blade is not modeled. What about the airframe? Airframe is under the rotor and being exposed to the disturbed flow by the rotor. The flow on it can be accepted as fully turbulent.

The turbulence model was chosen as the standard k - ϵ model. This model was first proposed in 1973 by Launder and Spalding [64]. Since then, it demonstrated its worth in a wide range of cases and became one of the most popular turbulence models. It comprises two turbulence quantities as explained in the previous chapter, the turbulence kinetic energy k , and the turbulence dissipation rate ϵ .

Boundary and initial conditions for inlet k and ϵ values are calculated according to the following formulas [65]:

$$k = 1.5(U_{inlet}I_{turb})^2 \quad (32)$$

where;

U is the mean velocity,

I is the turbulence intensity.

$$\epsilon = \frac{C_\mu^2 k^{1.5}}{0.07 D_H} \quad (33)$$

where;

D_H is the tunnel hydraulic diameter,

C_μ is a k - ϵ model constant that's commonly accepted as 0.09.

Standard k - ϵ wall functions in OpenFOAM, *kqRWallFunction* and *epsilonWallFunction*, were applied on all boundaries, including the immersed boundary.

The fvOptions file residing in the system folder can be used to define user-selected finite volume options, such as; sources and sinks. For rotor modeling, rotorDiskSource class is implemented via this dictionary as a momentum source. Furthermore, all the specifications related to the rotor (number of blades, airfoil lift and drag data, rpm, flap and pitch angles, etc.) should be defined here.

The rotorDisk source class can be used in two ways. The first way is to set a fixed trim by inputting collective and cyclic pitch values and is called *fixedTrim*. The second way is to set

target force and moments and let the code iteratively determine the trim values. It is called *targetForcetrिम*. The aerodynamic prediction performance of the VBM is out of context for this study, the effect of these options will not be discussed here. During the experiment, there was no mechanism to control the cyclic or collective pitch, because of that the rotorDisk is operated at *fixedTrim* mode. The rotor blade pitch was fixed to 10 degrees and it was input to fvOptions file as is. There was no pitching or rolling moment measurement or correction and likewise, none was applied in the model. The results of this choice and the applied changes will be discussed in Chapter 5..

In their computational analysis of the GIT case, Steijl and Barakos [66] observed that, when the rotor flapping was not included, the peak pressure values on top of the airframe were subdued. To be able to model the pressure fields correctly, the flapping values obtained from the experiment were included in this case.

The rotor geometry was defined according to the GIT experiment; 2-bladed rotor with zero twist, having constant chord of 0.086 m along the radius and operating at constant 2100 RPM with NACA0015 symmetrical airfoil. The tip loss correction factor was input as 0.96.

In fvOptions file, airfoil aerodynamic data can be defined in two ways. Lift and drag curves are applied either using “series” mode or “table lookup” mode. In the first mode, lift and drag curves are represented by sine and cosine series (Equation 34 and 35), respectively.

$$C_D = \sum_{i=1}^n (C_{d_\alpha})_i \times \cos(i \times \alpha_{eff}) \quad (34)$$

$$C_L = \sum_{i=1}^n (C_{l_\alpha})_i \times \sin(i \times \alpha_{eff}) \quad (35)$$

where;

C_{d_α} is the slope of C_d curve with respect to angle of attack α ,

C_{l_α} is the slope of C_l curve with respect to angle of attack α ,

and i subscript represents the i_{th} coefficient which is obtained by curve fitting to the C_l and C_d graphs.

On the other hand, the “table lookup” mode works by representing the aerodynamic coefficient curves with point data. Specific values are obtained by linear interpolation. For each mode, airfoil data should be defined from $-\pi$ to $+\pi$.

The airfoil aerodynamic data for the study of Reference [49] is obtained from XFOil. XFOil is a 2D panel method code developed by Mark Drela [67] for the design and analysis of subsonic airfoils. The aerodynamic data was benchmarked in the study of Wahono [49]. Therefore, the same lookup table for NACA0015 is used here.

4.2.4. Flow Solver Settings

For flow speeds up to 0.3 Mach, the compressibility effects are assumed to be negligible [59]. The rotor tip speed was 100.5 m/s and when the freestream velocity is factored in, the maximum velocity for the case becomes 0.32 Mach at the advancing side of the rotor. Even though the Mach speed is higher than the incompressibility limit, it was shown [49] that the solutions made with the incompressibility assumption produce better results in the GIT airframe CFD analysis. The flow solver was chosen accordingly.

OpenFOAM 3.0.1 includes a variety of solvers and is capable of dealing with different types of flow physics. The solvers are differentiated according to the type of flow being solved. A review of these solvers can be found in Reference [68]. The flow type for the GIT airframe can be summarized as; incompressible, steady state, and turbulent. A robust solver that's suitable for that type of flow is *simpleFOAM*. This steady-state RANS (Reynolds Averaged Navier-Stokes) solver is based on the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm.

The SIMPLE method was first introduced by Patankar and Spalding [64] as a way of solving the fundamental governing equations of fluid mechanics. It is an iterative algorithm that acts in a successive guess and correct fashion. The iteration cycle starts with the calculation

of an intermediate velocity field that satisfies the estimated pressure field according to the linearised momentum equations. Then, these values are corrected to also satisfy the continuity equation and the corrected pressure solution is used to correct the initial estimation in the momentum equation. The iterations continue until the pressure and velocity values converge into a solution. Although it is computationally cost-effective, the algorithm is inherently unstable [49] and it can lead to oscillations and divergence unless countermeasures have been taken with the appropriate choice of solver settings.

The strategy to counteract the oscillations and divergence applied in this study can be summarized as follows;

- Starting iterations with first-order divergence schemes like Gauss upwind and Gauss linear,
- Starting iterations with limited gradient schemes of k and ϵ ,
- Starting iterations of pressure with the matrix solver *smoothSolver*,
- Keeping solver tolerances high at initial steps and as the solution stabilizes, tolerances were decreased,
- Keeping under-relaxation factor for p at 0.3 and for the others at 0.4 even after the solution stabilizes.

These settings removed the oscillatory behavior of the case. After the case converged with these settings; second-order divergence schemes were applied, the limits of the gradient schemes were decreased, and matrix solver tolerances were lowered step by step.

For optimizing the case settings (mesh, solvers, schemes), many cases with small differences were prepared. For each of these cases, iterating until first-order convergence would have taken a lot of time. Therefore, a sample first-order solution was prepared and its converged iteration was used as the initial condition for all other cases. In OpenFOAM, this is done by mapping a solution to another case with *mapFields* command. This approach shortened the computational time needed for the final solution.

5. RESULTS AND DISCUSSIONS

The GIT rotor-airframe experiment was simulated in three different meshes and two different rotor settings. Although initially four different meshes were prepared, it's seen that the mesh convergence was reached and no simulation was run on the mesh with the highest elements, 3.2 million. The other meshes had 1.3 million, 1.9 million and 2.6 million elements and were run with exactly the same settings. It can be seen from the Figure 5.1 that, although the computed pressure value on top of the body for 1.3 million is very different, it is practically the same for both 1.9 million and 2.6 million meshes.

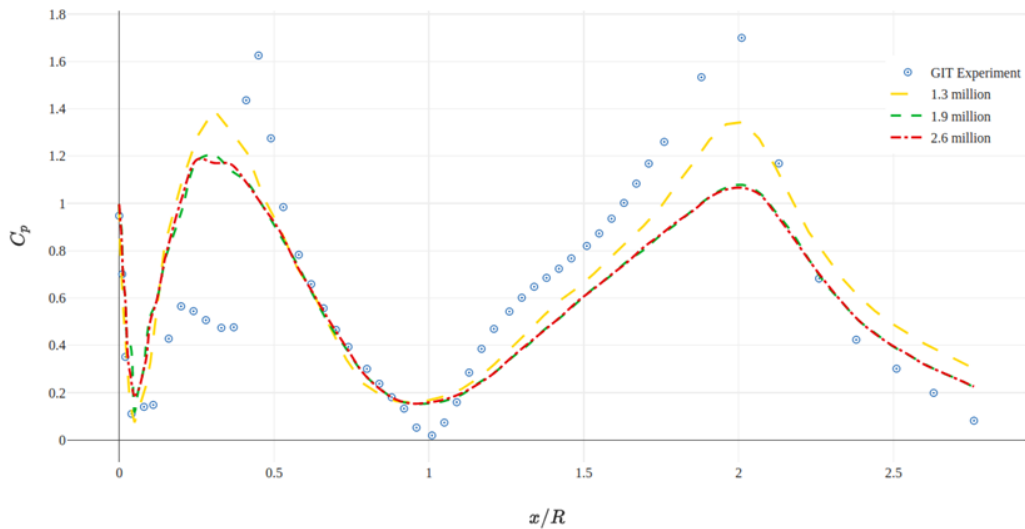


Figure 5.1 Comparison of the pressure values taken from the top of the airframe, obtained from three different meshes

One thing that catches attention is that the result of the 1.3 million mesh in Figure 5.1 seems closer to the experimental values. For removing all doubt, computed downward velocity data taken from the top of the airframe is also visualized in Figure 5.2 for all three meshes. It is very clear that 1.3 million mesh data has more errors than the other two meshes. This leads to the conclusion that the 1.3 million elements are not enough for an accurate numerical analysis of the GIT experiment.

Therefore, it was assumed that the mesh convergence is reached and the other simulations were run on the mesh with 1.9 million elements.

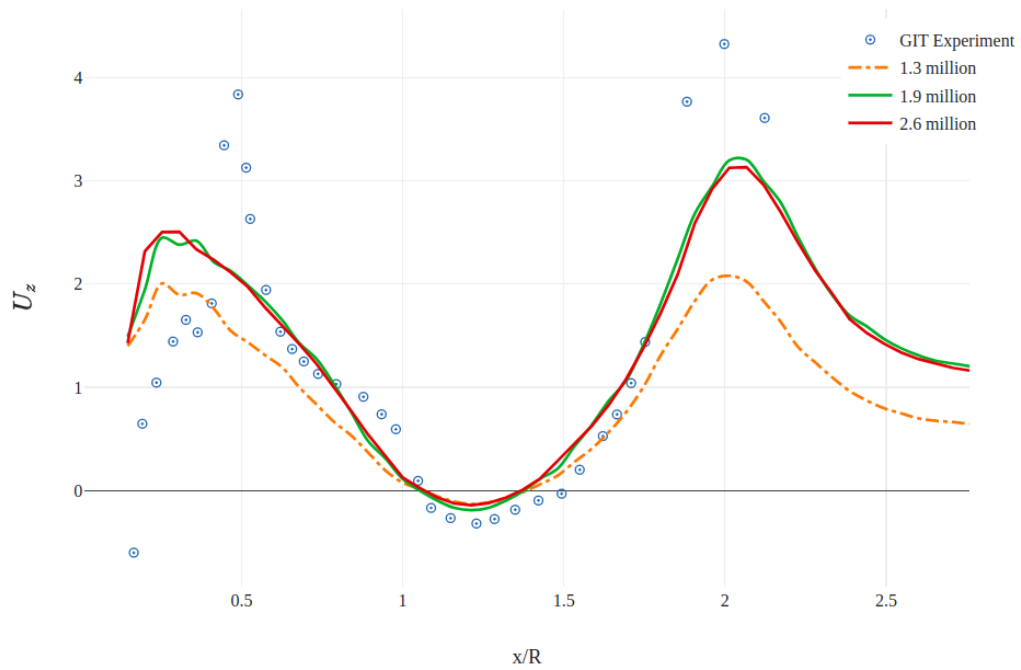


Figure 5.2 Comparison of the downward velocity values taken from 12.7mm above from the top of the airframe, obtained from three different meshes

The above analyses were made with the collective pitch and flapping values of the experiment. With these settings, it is observed that rotor thrust and the pressure values on top of the airframe were underpredicted. As Ruith [69] and Wahano [49] pointed out, using the same pitch as the experiment yields faulty results. The Virtual Blade Method makes accurate aerodynamic predictions only when the rotor is operated at the correct thrust value and with zero rotor moment about the hub. For complying with these restrictions, Wahano modeled the rotor with a 12 degrees collective pitch. In this study, the findings will be compared against the experiment but also against a body-conformal analysis that used the same CFD and rotor settings. Wahano’s research is the ideal candidate since the case specifications are the same; steady-state simpleFoam solver with k-epsilon turbulence model. Hence, the analysis was run with the same rotor settings.

5.1. Pressure Field

To give a general sense of the flowfield, mean pressure contour plots of x-z and y-z plane were shown in the Figure 5.3 and 5.4. The airframe was shown only for easy viewing.

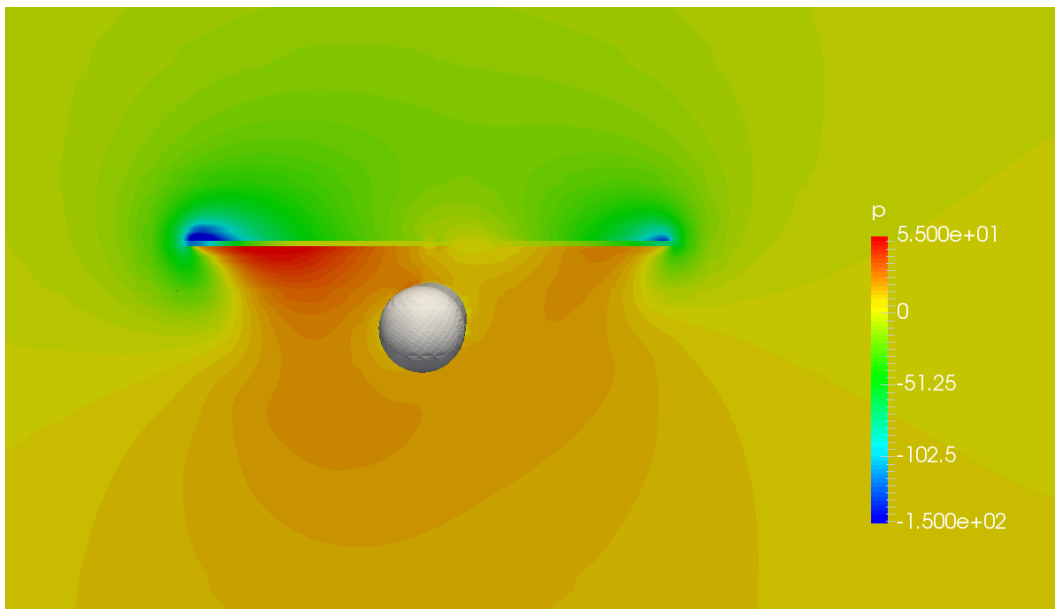


Figure 5.3 The front view of the pressure field in Pascal around the airframe

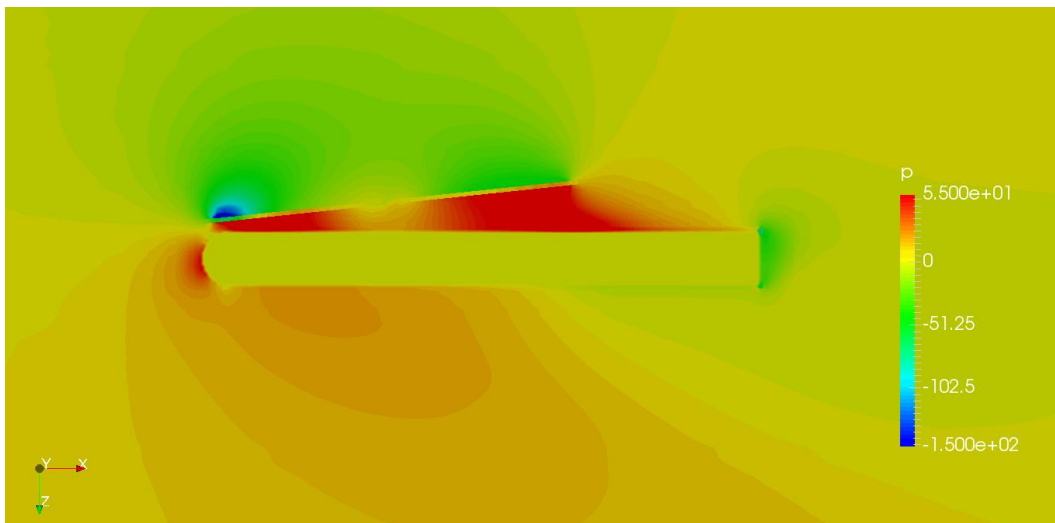


Figure 5.4 The side view of the pressure field in Pascal around the airframe

In the Figure 5.3, there is an asymmetry between the high-pressure areas below the rotor and the low-pressure areas above the rotor. This is expected due to anti-clockwise rotation

of the blades. The relative speed of the advancing blade is higher than the retreating side. Therefore, dissymmetry of lift occurs, and the pressure field forms accordingly. Around the low-pressure areas, the effect of tip vortices can be seen.

Figure 5.4 shows the side view of the airframe. The immersed boundary can be seen clearly from that view. The stagnation point seems to have formed just below the axisymmetric center, due to the downwash of the rotor. The pressure loss due to separation can be seen at downstream of the airframe. The high-pressure areas under the rotor and the pressure loss due to the hub seem to have formed as expected.

5.2. Coefficient of Pressure Comparison

In the experiment, measured mean surface pressure values are nondimensionalized with respect to the dynamic pressure, as in the equation below:

$$C_p = \frac{P - P_\infty}{\frac{1}{2}\rho V^2} \quad (36)$$

The mean pressure distribution is measured on the top and bottom surfaces, and on the port and starboard sides of the airframe. Port and starboard are terms that refer to the left and right of the aircraft respectively. The following figures provide a comparison between the experiment, the body-conformal (shortened as BC in the figures) method, and the IB method. The body-conformal method chosen for comparison is described in detail at Reference [49].

As can be seen from the Figure 5.5 both body-conformal and immersed boundary methods successfully resolved pressure drop near the nose due to velocity increase but IB method seems to capture the lowest Pressure value better. Body-conformal methods [49, 66, 69] seem to fail to resolve the first pressure peak on top. This may have been due to the rotor modeling technique but the studies that use different modeling techniques [15, 66] also show the same behavior. This study seems to capture the first peak but fails to resolve the rapid pressure fall. Park [15] commented that further mesh refinement in the vicinity of the airframe may improve their results. No similar tendency is observed in the mesh convergence examination

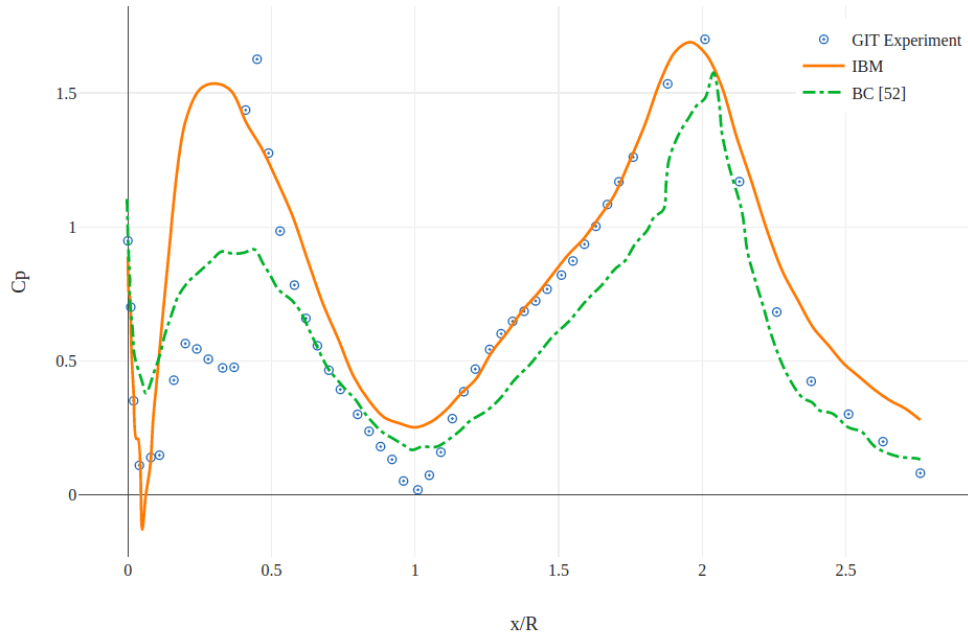


Figure 5.5 Comparison of measured and computed C_p on top of the airframe

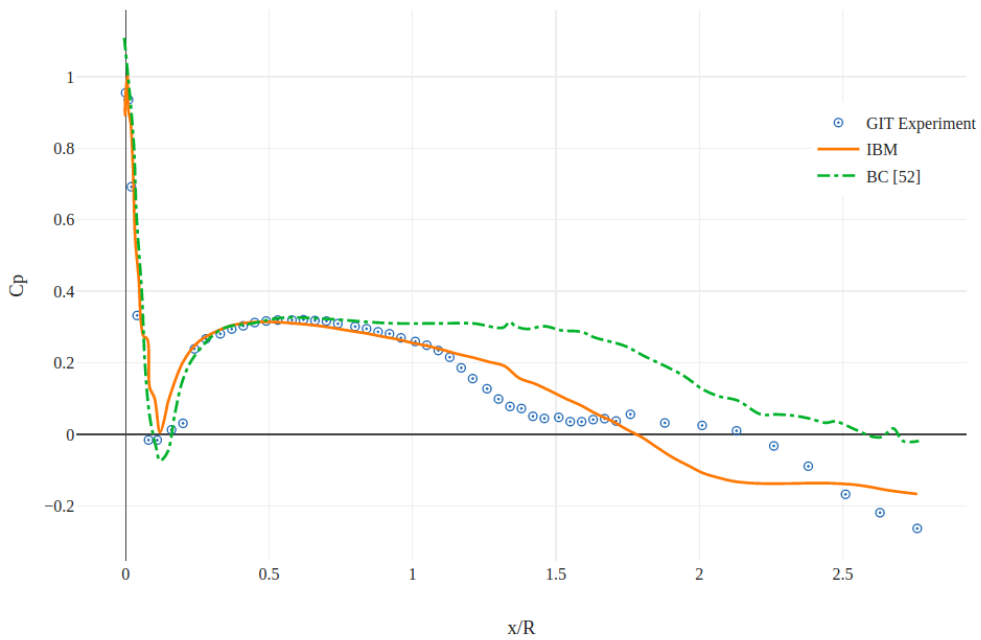


Figure 5.6 Comparison of measured and computed C_p on the bottom of the airframe

of this work (Figure 5.1). The other possible reason is the use of hexahedral mesh against the tetrahedral mesh [49, 69]. This possibility is also eliminated by the studies that used [15, 66] hexahedral mesh. One possible cause is the use of nonconformal mesh. Although

the domain face that's in contact with the rotor surface was carefully meshed and was refined to have more elements than the rotor, it may have introduced numerical diffusion to the case. This deduction will not be investigated further due to the reason of being out of the scope of this study, but it can be examined in follow-up researches.

Figure 5.6 shows the Pressure measurement on the bottom side. Both approaches failed to model the rapid decrease between $x/R=1$ to 1.5 but IB method seems to produce closer values to the experiment.

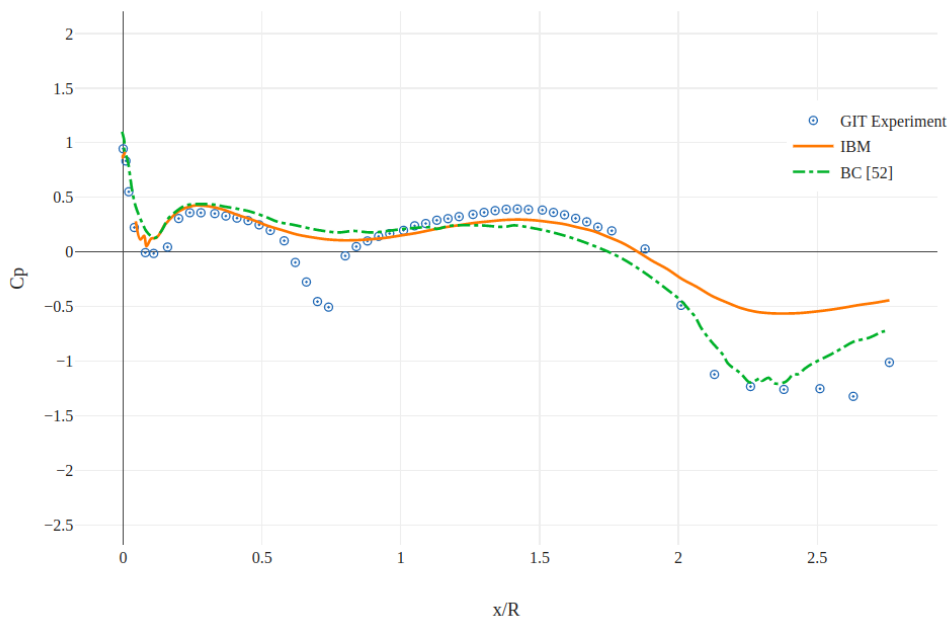


Figure 5.7 Comparison of measured and computed C_p on the port side of the airframe

The port (Figure 5.7) and starboard (Figure 5.8) side comparisons show that although the IB method and BC method has some differences, the end result of the IB method is at an acceptable level and at least as good as the body-conformal method.

The difference between the experimental and computed values are quantified by using the Root Mean Square Error (RMSE) in Table 5.1. RMSE is a way of quantifying the difference between a calculated and a real value.

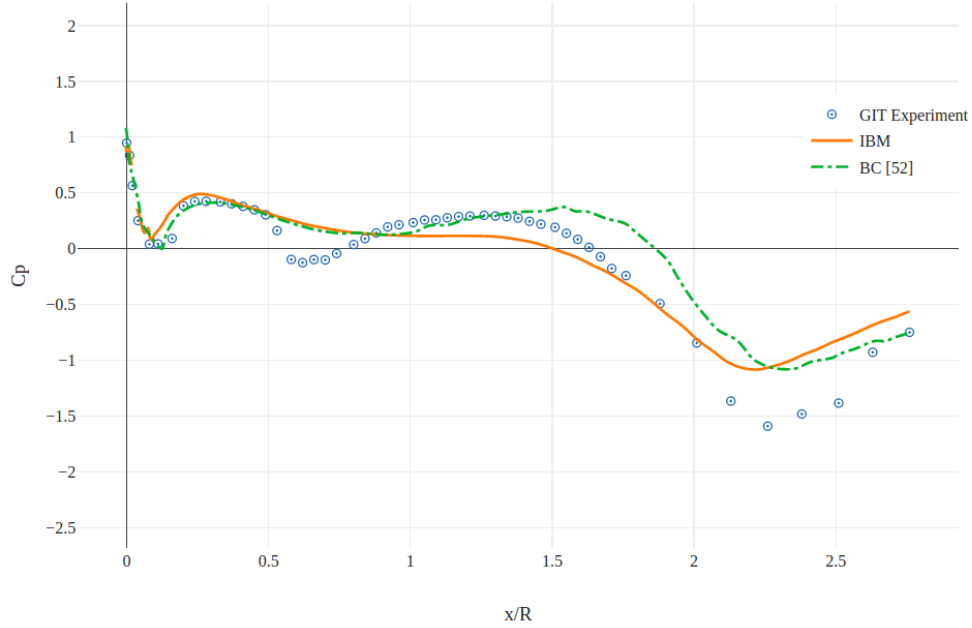


Figure 5.8 Comparison of measured and computed C_p on the starboard side of the airframe

RMSE is calculated according to following formula,

$$RMSE = \sqrt{\sum_{i=1}^n \left(\frac{C_{exp} - C_{calc}}{n} \right)^2} \quad (37)$$

where n represents the sample size, C_{exp} is the experimental value measured for pressure coefficient, C_{calc} is the calculated value of the pressure coefficient. The values at Table 5.1 shows that the difference is at an acceptable level.

5.3. Velocity Field Comparison

One of the advantages of the IB method by Xu and Liu [55] is the absence of irregularities in the near-wall values. This effect can be observed by the velocity near the airframe. Liou et al. [70] performed mean velocity measurements 12.7 mm above the body surface (Figure 5.9) for the GIT experiment. The measurements were done both for the axial direction and the downward direction.

In Figure 5.10 and Figure 5.11 the velocity values are shown with respect to each other. The

Table 5.1 Pressure coefficient RMS error values for each side

Side	RMSE
Top	0.36
Bottom	0.06
Port	0.30
Starboard	0.21

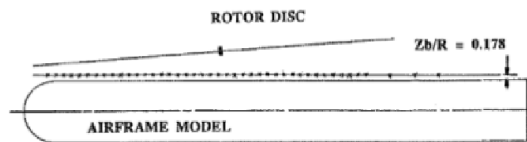


Figure 5.9 Location of the axial and downward velocity measurement made by [61]

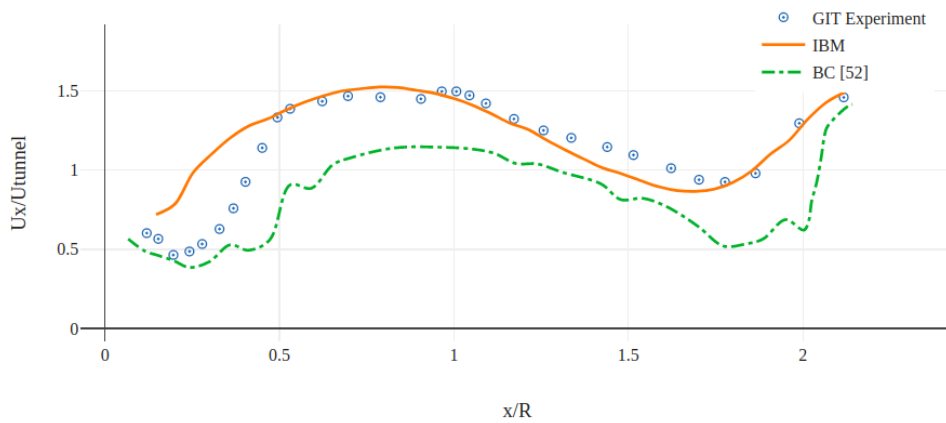


Figure 5.10 Comparison of computed and measured axial velocities above the airframe

IB method seems to have captured the general characteristics of the axial velocity better than the body conformal method. No irregularities associated with bad immersed cell selection or bad boundary forming are observed. The IB method seems to have captured the axial near-wall velocity better than the body conformal method. This may be due to mesh cell

type. The body-conformal computational study used unstructured tetrahedral meshes. There is no indication of the use of flow-aligned prismatic meshes on the airframe surface. If that's the case, this difference may be due to the better flow and mesh cell alignment in the IB application.

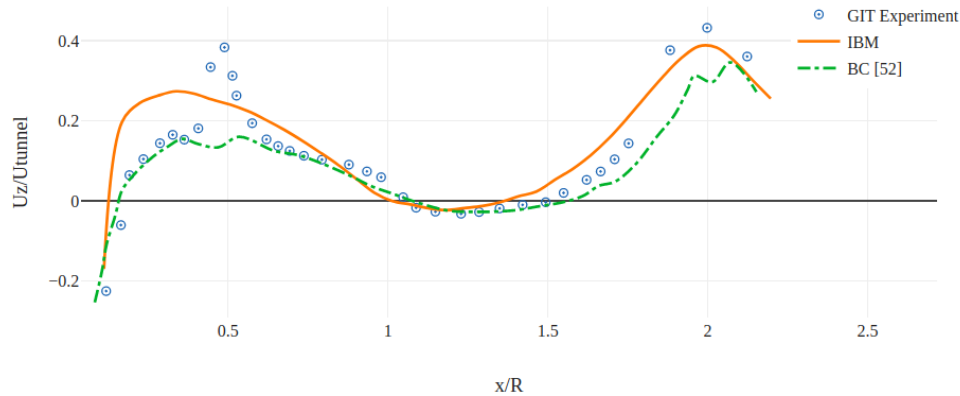


Figure 5.11 Comparison of computed and measured downward velocities above the airframe

Both pressure field comparison and velocity field comparison show great agreement with the experiment. The IB method seems to have performed on the top and bottom sides better than the BC method. The velocity field seems to be formed very close to the experiment. Although, it is clear that both body-conformal and immersed boundary models failed to capture the velocity gradients and the pressure gradients but this may have another reason. Steijl [66] commented that this insensitivity to rapid changes might be a general characteristic of the RANS turbulence models.

It can be concluded that; for the GIT rotor-airframe experiment, the IB method is on par with body conformal methods.

6. CONCLUSION

6.1. Conclusion

The remaining challenges of the rotorcraft aerodynamics can be summarized in two topics. Complex cases that demand high setup effort and accordingly high computational expenses. The aim of this thesis was to suggest a rotorcraft aerodynamics analysis tool that's free of these challenges. The suggested method combines IBM, VBM, and AMI to allow fast meshing that is independent of the airframe geometry. Immersed Boundary Method allows the use of the same domain mesh with different geometries. Virtual Blade Method allows changing the number of blades, airfoil type, rpm, twist, pitch, and taper without needing any change in the rotor mesh. It can be a very valuable tool for the design stage of a rotorcraft. Although the method is flexible, it doesn't compromise accuracy and its results are on par with the body-conformal methods.

In this thesis, basics of the helicopter aerodynamics have been introduced and the VBM method has been explained. Up-to-date IB methods and their applications in OpenFOAM have been reviewed and the one suitable for aerospace applications has been selected. The y^+ adaptive IBM method of Xu and Liu [55] was coupled with the VBM method [49] in OpenFOAM. The GIT rotor-airframe interaction experiment has been chosen due to its extensive dataset and its popularity among verification and validation studies. This experiment allowed comparing the IB method results against body-conformal methods and experiment results.

The rotor and domain were meshed with Gambit and snappyHexMesh. The separate meshing allowed individual refinement of both zones. Nonconformal mesh faces were connected with the cyclicAMI library of OpenFOAM. The numerical setup and boundary conditions were determined according to experiment. Running the case showed that it is prone to oscillations and divergence. Therefore; under-relaxation factors, linear solvers and discretization schemes were chosen accordingly to dampen the numerical oscillations.

Examining the results has shown that, the discrepancies between the experiment and the analyses were mainly due to limitations of the VBM. It can be concluded that, with the correct mesh parameters, the IB method produces results that are on par with body-conformal methods.

6.2. Future Research

- **Performing the same analysis with a better tip effect modeling** the tip loss modeling of the VBM is an oversimplification. There are other rotor tip loss models in the literature. The analysis can be repeated with a better tip loss correction and see if the results improve, especially in the port and starboard sides.
- **Checking the performance of IBM in transient cases** The performance of IBM in the steady-state analysis was found to be satisfactory. A similar study should be done to evaluate the IBM in a time-dependent setting.
- **IBM with other rotor modeling techniques** The modeling of the rotor with VBM only allows steady-state analysis. The study can be extended by using other rotor modeling techniques that allow time dependent operation.
- **A study to better understand the effect of AMI** The cyclicAMI application may have been introduced numerical diffusion to the case. The effect of cyclicAMI can be analyzed on simpler cases.
- **IBM on a more complicated geometry** In this study, most of the airframe was in line with the flow and mesh elements. The performance of IBM on more complex geometries should be evaluated.

REFERENCES

- [1] Robin B Gray. On the motion of the helical vortex shed from a single-bladed hovering model helicopter rotor and its application to the calculation of the spanwise aerodynamic loading. Technical report, PRINCETON UNIV NJ JAMES FORRESTAL RESEARCH CENTER, 1955.
- [2] Nathan Hariharan and Lakshmi Sankar. A review of computational techniques for rotor wake modeling. In *38th Aerospace Sciences Meeting and Exhibit*, page 114, 2000.
- [3] Nasa@sc18: Advanced computational fluid dynamics tools for accurate rotorcraft analysis and design. <https://www.nas.nasa.gov/SC18/demos/demo1.html>.
- [4] Peter F Lorber and T Alan Egolf. An unsteady helicopter rotor-fuselage aerodynamic interaction analysis. *Journal of the American Helicopter Society*, 35(3):32–42, 1990.
- [5] Albert Gerard Brand. *An experimental investigation of the interaction between a model rotor and airframe in forward flight*. Georgia Institute of Technology, 1989.
- [6] Laith AJ Zori and R Ganesh Rajagopalan. Navier—stokes calculations of rotor—airframe interaction in forward flight. *Journal of the American Helicopter Society*, 40(2):57–67, 1995.
- [7] Nathan Hariharan and Lakshmi N Sankar. Unsteady overset simulation of rotor-airframe interaction. *Journal of Aircraft*, 40(4):662–674, 2003.
- [8] T Renaud, C Benoit, J-C Boniface, and P Gardarein. Navier-stokes computations of a complete helicopter configuration accounting for main and tail rotors effects. *ONERA, TP no. 2003-142*, 2003(142), 2003.

- [9] Onera. <http://elsa.onera.fr/>. Accessed: 2021-06-50.
- [10] Jochen Raddatz and Jens K Fassbender. Block structured navier-stokes solver flower. In *MEGAFLOW-Numerical Flow Simulation for Aircraft Design*, pages 27–44. Springer, 2005.
- [11] Thomas Gerhold, M Galle, O Friedrich, J Evans, T Gerhold, M Galle, O Friedrich, and J Evans. Calculation of complex three-dimensional configurations employing the dlr-tau-code. In *35th Aerospace Sciences Meeting and Exhibit*, page 167, 1997.
- [12] Thomas Renaud, David O’Brien, Marilyn Smith, and Mark Potsdam. Evaluation of isolated fuselage and rotor-fuselage interaction using cfd. Technical report, OFFICE NATIONAL D’ETUDES ET DE RECHERCHES AEROSPATIALES CHATILLON (FRANCE), 2004.
- [13] M Potsdam, Marilyn Smith, and Thomas Renaud. Unsteady computations of rotor-fuselage interaction. volume 1, 01 2009.
- [14] Andrew M Wissink, Jayanarayanan Sitaraman, Buvaneswari Jayaraman, Beatrice Roget, Vinod K Lakshminarayan, Mark A Potsdam, Rohit Jain, Andrew Bauer, and Roger Strawn. Recent advancements in the helios rotorcraft simulation code. In *54th AIAA Aerospace Sciences Meeting*, page 0563, 2016.
- [15] Hee Sung Park. *Immersed Boundary Method for High Reynolds Number Computation of Rotorcraft Aerodynamics*. PhD thesis, University of Sydney, 2020.
- [16] Clément Crozon. *Coupling flight mechanics and CFD: numerical simulation of shipborne rotors*. PhD thesis, University of Liverpool, 2015.
- [17] Joel H. Ferziger, Milovan Perić, and Robert L. Street. *Computational Methods for Fluid Dynamics*. Springer International Publishing, 2020.

- [18] Tayfun E. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8(2):83–130, June 2001.
- [19] Rajat Mittal and Gianluca Iaccarino. IMMERSED BOUNDARY METHODS. *Annual Review of Fluid Mechanics*, 37(1):239–261, January 2005.
- [20] Charles S Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, October 1972.
- [21] Richard P Beyer. A computational model of the cochlea using the immersed boundary method. *Journal of Computational Physics*, 98(1):145–162, January 1992.
- [22] Lisa J. Fauci and Amy McDonald. Sperm motility in the presence of boundaries. *Bulletin of Mathematical Biology*, 57(5):679–699, September 1995.
- [23] Ming-Chih Lai and Charles S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160(2):705–719, May 2000.
- [24] Mohd-Yusof J. Combined immersed-boundary/b-spline methods for simulations of flow in complex geometries. *CTR Annual Research Briefs*, pages 317–327, 1997.
- [25] Roberto Verzicco, Jamaludin Mohd-Yusof, Paolo Orlandi, and Daniel Haworth. Large eddy simulation in complex geometric configurations using boundary body forces. *AIAA Journal*, 38(3):427–433, March 2000.
- [26] Elias Balaras. Modeling complex boundaries using an external force field on fixed cartesian grids in large-eddy simulations. *Computers & Fluids*, 33(3):375–404, March 2004.

- [27] 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*, 161(1):35–60, June 2000.
- [28] Sekhar Majumdar, Gianluca Iaccarino, and Paul A. Durbin. Rans solvers with adaptive structured boundary non-conforming grids. 2002.
- [29] Reza Ghias, Rajat Mittal, and Thomas Lund. A non-body conformal grid method for simulation of compressible flows with complex immersed boundaries. In *42nd AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, January 2004.
- [30] Rajat Mittal, C. Bonilla, and H. Udaykumar. Cartesian grid methods for simulating flows with moving boundaries. *Computational Engineering*, 4:557–566, 01 2003.
- [31] R. Mittal, Y. Utturkar, and H. Udaykumar. Computational modeling and analysis of biomimetic flight mechanisms. In *40th AIAA Aerospace Sciences Meeting & Exhibit*. American Institute of Aeronautics and Astronautics, January 2002.
- [32] Gordon J Leishman. *Principles of helicopter aerodynamics with CD extra*. Cambridge university press, 2006.
- [33] Wayne Johnson. *Helicopter theory*. Courier Corporation, 2012.
- [34] Department of Army Headquarters. *Fundamentals of Flight*. US Army, 2012.
- [35] Helicopter Flying Handbook. Faa-h-8083-21a. *United States Department of Transportation, Federal Aviation Administration, Airman Testing Branch*, 2019.
- [36] John M Seddon and Simon Newman. *Basic helicopter aerodynamics*. John Wiley & Sons, 2011.
- [37] R Ganesh Rajagopalan and Chin K Lim. Laminar flow analysis of a rotor in hover. *Journal of the American Helicopter Society*, 36(1):12–23, 1991.

- [38] R Ganesh Rajagopalan and Sanjay R Mathur. Three dimensional analysis of a rotor in forward flight. *Journal of the American Helicopter Society*, 38(3):14–25, 1993.
- [39] Wayne Johnson. Airloads, wakes, and aeroelasticity. Technical report, 1990.
- [40] H. Bateman C. Lock and H. Townend. *An extension of the Vortex Theory of Airscrews with applications to airscrews of small pitch, including experimental results*. A.R.C. Research Reports and Memoranda, 1925.
- [41] H. Glauert. *The Elements of Aerofoil and Airscrew Theory*. Cambridge University Press, June 1983.
- [42] William Froude. On the elementary relation between pitch, slip, and propulsive efficiency. Technical report, 1920.
- [43] Håkan Nilsson. Description and validation of the rotordisksource class for propeller performance estimation, 2017.
- [44] Jeremy Ledoux, Sebastián Riffo, and Julien Salomon. Analysis of the blade element momentum theory. *SIAM Journal on Applied Mathematics*, 81(6):2596–2621, 2021.
- [45] GAM Van Kuik, Jens Nørkær Sørensen, and VL Okulov. Rotor theories by professor joukowsky: momentum theories. *Progress in Aerospace Sciences*, 73:1–18, 2015.
- [46] W. J. Rankine. On the mechanical principles of the action of propellers. *Trans. Roy. Inst. Naval Arch.*, 6:13–30, 1865.
- [47] A. Betz. Das maximum der theoretisch möglichen ausnützung des windes durch windmotoren. *Zeitschrift für das gesamte Turbinenwesen*, 26:307–309, 1920.
- [48] John David Anderson. *Aircraft performance and design*, volume 1. WCB/McGraw-Hill Boston, 1999.

- [49] Stefano Wahono. Development of virtual blade model for modelling helicopter rotor downwash in openfoam. Technical report, DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION FISHERMANS BEND, 2013.
- [50] E. Constant, J. Favier, M. Meldi, P. Meliga, and E. Serre. An immersed boundary method in OpenFOAM : Verification and validation. *Computers & Fluids*, 157:55–72, November 2017.
- [51] A. Pinelli, I.Z. Naqavi, U. Piomelli, and J. Favier. Immersed-boundary methods for general finite-difference and finite-volume navier–stokes solvers. *Journal of Computational Physics*, 229(24):9073–9091, December 2010.
- [52] Markus Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476, November 2005.
- [53] Favier, J., Meliga, P., Serre, E., Constant, E., Meldi, M., Goncalves, E., and Hamza, R. Immersed boundary in openfoam. <https://www.researchgate.net/project/Immersed-boundary-in-Openfoam>, 2018. "Accessed: 2022-08-10".
- [54] Hrvoje Jasak, Damir Rigler, and Željko Tuković. Design and implementation of immersed boundary method with discrete forcing approach for boundary conditions. 2014.
- [55] Yuncheng Xu and Xiaofeng Liu. An immersed boundary method with y+ adaptive wall function for smooth wall shear. *International Journal for Numerical Methods in Fluids*, 93(6):1929–1946, February 2021.
- [56] R. PITZ and J. DAILY. Experimental study of combustion in a turbulent free shear layer formed at a rearward facing step. In *19th Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, January 1981.
- [57] Nparc verification and validation web site archives page. <https://www.grc.nasa.gov/WWW/wind/valid/archive.html>. Accessed: 2022-08-15.

- [58] An immersed boundary method with y^+ -adaptive wall function for smooth wall shear. https://github.com/psu-efd/ibwallfunction_OpenFOAM. Accessed: 2022-07-15.
- [59] Yunus Cengel and John Cimbala. *EBOOK: Fluid Mechanics Fundamentals and Applications (SI units)*. McGraw Hill, 2013.
- [60] Brian Edward Launder and Dudley Brian Spalding. The numerical computation of turbulent flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 96–116. Elsevier, 1983.
- [61] SG Liou, NM Komerath, and HM McMahan. Measurement of the interaction between a rotor tip vortex and a cylinder. *AIAA journal*, 28(6):975–981, 1990.
- [62] Brian G Allan, Norman W Schaeffler, Luther N Jenkins, Chung-Sheng Yao, Oliver D Wong, and Philip E Tanner. Active aerodynamic load reduction on a rotorcraft fuselage with rotor effects: A cfd validation effort. In *American Helicopter Society International Annual Forum & Technology Display*, number NF1676L-19647, 2015.
- [63] Chris Greenshields. Openfoam 2.3.0: Arbitrary mesh interface. <https://openfoam.org/release/2-3-0/non-conforming-ami/>, Apr 2016. "Accessed: 2022-08-12".
- [64] LS Caretto, AD Gosman, SV Patankar, and DB Spalding. Two calculation procedures for steady, three-dimensional flows with recirculation. In *Proceedings of the third international conference on numerical methods in fluid mechanics*, pages 60–68. Springer, 1973.
- [65] Turbulence free-stream boundary conditions. "Accessed: 2022-07-05".
- [66] R. Steijl and G. N. Barakos. Computational study of helicopter rotor-fuselage aerodynamic interactions. *AIAA Journal*, 47(9):2143–2157, 2009.

- [67] Mark Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In *Low Reynolds number aerodynamics*, pages 1–12. Springer, 1989.
- [68] Chris Greenshields. Openfoam v10 user guide, Jul 2022.
- [69] Michael Ruith. Unstructured, multiplex rotor source model with thrust and moment trimming-fluent’s vbm model. In *23rd AIAA applied aerodynamics conference*, page 5217, 2005.
- [70] SG Liou, NM Komerath, and HM McMahon. Velocity field of a cylinder in the wake of a rotor in forward flight. *Journal of aircraft*, 27(9):804–809, 1990.