UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING

GENERATION OF AN AI DEMONSTRATOR FOR AIRCRAFT AERODYNAMICS

by

ALEXIA KALATHA

Supervisors
Dr Konstantinos Ritos, University of Thessaly
Dr Paul Hutcheson, Ansys Hellas

A thesis submitted to the University of Thessaly in partial fulfillment of the requirements for the degree of Integrated M.Sc., in Mechanical Engineering.

Volos, 2025

UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING

GENERATION OF AN AI DEMONSTRATOR FOR AIRCRAFT AERODYNAMICS

by

ALEXIA KALATHA

Supervisors
Dr Konstantinos Ritos, University of Thessaly
Dr Paul Hutcheson, Ansys Hellas

A thesis submitted to the University of Thessaly in partial fulfillment of the requirements for the degree of Integrated M.Sc., in Mechanical Engineering.

Volos, 2025

Approved by Final Examination Committee:

Supervisor:	Assistant Prof. Konstantinos Ritos, Department of Mechanical Engineering, University of Thessaly
Member	Associate Prof. Georgios Charalampous, Department of Mechanical Engineering, University of Thessaly
Member	Prof. Konstantinos Ampountolas, Department of Mechanical Engineering, University of Thessaly

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Dr. Konstantinos Ritos, for all his advice and guidance during the work for my diploma thesis. I would also like to thank Prof. K. Ampountolas and Assistant Prof. G. Charalampous for serving in my thesis committee.

I am deeply indebted to my manager, Dr. Paul Hutcheson, for his constant support and insightful guidance during my internship at Ansys Hellas. My heartfelt thanks also go to my mentors, Marco Coderoni and Valerio Viti, for their practical assistance during the research phase of my thesis. I would furthermore like to express my appreciation to all my colleagues at the Ansys Hellas office, as well as to those across Ansys worldwide. It is truly a privilege to collaborate with such talented people, and I look forward to continuing this journey together.

Θα ήθελα να ευχαριστήσω από καρδιάς τους γονείς μου, Αθανάσιο και Βασιλική, για όλη την αγάπη, την υπομονή και τη στήριξή τους, καθώς και τα αδέρφια μου, Νίκο και Κώστα. Χωρίς αυτούς δεν θα είχα καταφέρει να υλοποιήσω τους έως τώρα στόχους μου.

Δεν μπορώ, φυσικά, να παραλείψω μερικά από τα αγαπημένα μου πρόσωπα, για όλες τις όμορφες στιγμές που μοιραστήκαμε κατά τη διάρκεια των σπουδών μου στον Βόλο. Μαρία, Ελπίδα, Δωροθέα, Παντελή, Μαριάννα, Δέσποινα, Νάσια, Ειρήνη, Ελίζα, Νότα και Σμαράγδα, σας ευχαριστώ.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην ομάδα «Κένταυρος» του Πανεπιστημίου Θεσσαλίας και στα μέλη της, με τα οποία συνεργαστήκαμε και ζήσαμε μοναδικές εμπειρίες και σπουδαίες νίκες.

GENERATION OF AN AI DEMONSTRATOR FOR AIRCRAFT AERODYNAMICS

ALEXIA KALATHA
Department of Mechanical Engineering, University of Thessaly

Supervisors:
Dr Konstantinos Ritos
Dr Paul Hutceson, Application Engineering Manager

Abstract

The increasing demand for accurate yet computationally efficient aerodynamic analysis has motivated the integration of Artificial Intelligence (AI) into conventional Computational Fluid Dynamics (CFD) workflows. The current thesis, entitled "Generation of an AI Demonstrator for Aircraft Aerodynamics", presents the development of AI models in SimAI, a proof-of-concept system that combines high-fidelity CFD data with machine learning models for aerodynamic prediction.

High-quality datasets were generated with the HB-2 and AGARD standard models by detailed CFD processes, including mesh generation, solver settings, and mesh-independence studies. These datasets served as the foundation for AI demonstrator training and validation, in both parametric and non – parametric cases..

Results show that AI models can recreate aerodynamic behavior with encouraging accuracy at much less computational cost than traditional CFD. Sensitivity analyses illustrate the effect of data density, boundary conditions, and model architecture on predictive performance.

The findings confirm the effectiveness of AI-accelerated aerodynamics tools and demonstrate their promise towards accelerating design processes, enabling real-time applications, and supporting control strategies. Research opportunities are indicated in hybrid CFD–AI methods, enhanced turbulence modeling, and large-scale surrogate modeling.

Key words: Artificial Intelligence (AI), Machine Learning (ML), Aerodynamics, Computational Fluid Dynamics (CFD), Training Data, HB2, AGARD

ΑΝΑΠΤΥΞΗ ΕΠΙΔΕΙΚΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΓΙΑ ΑΕΡΟΔΥΝΑΜΙΚΗ ΑΕΡΟΣΚΑΦΩΝ

ΑΛΕΞΙΑ ΚΑΛΑΘΑ

Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας, 2025

Επιβλέπων Καθηγητής: Δρ. Κωνσταντίνος Ρήτος

Επιβλέπων: Δρ. Paul Hutcheson, Ansys Hellas

Περίληψη

Η αυξανόμενη ζήτηση για ακριβή αλλά και υπολογιστικά αποδοτική αεροδυναμική ανάλυση έχει οδηγήσει στην ενσωμάτωση της Τεχνητής Νοημοσύνης σε συμβατικές ροές υπολογιστικής ρευστοδυναμικής . Η παρούσα διπλωματική εργασία, με τίτλο *«Ανάπτυξη ενός Επιδεικτικού Συστήματος Τεχνητής Νοημοσύνης για Αεροδυναμική Αεροσκαφών»*, παρουσιάζει την ανάπτυξη μοντέλων τεχνητής νοημοσύνης στο SimAI, ένα σύστημα το οποίο συνδυάζει δεδομένα υψηλής πιστότητας CFD με μοντέλα μηχανικής μάθησης για την πρόβλεψη αεροδυναμικής συμπεριφοράς.

Σύνολα δεδομένων υψηλής ποιότητας παρήχθησαν με τα πρότυπα μοντέλα HB-2 και AGARD μέσω λεπτομερών αναλύσεων, που περιλάμβαναν παραγωγή πλέγματος, παραμετροποίηση του επιλύτη και μελέτες ανεξαρτησίας πλέγματος. Τα δεδομένα αυτά αποτέλεσαν τη βάση για την εκπαίδευση και επικύρωση των ανεπτυγμένων μοντέλων τόσο σε παραμετρικές όσο και σε μη παραμετρικές περιπτώσεις.

Τα αποτελέσματα δείχνουν ότι τα μοντέλα AI μπορούν και αναπαράγουν την αεροδυναμική συμπεριφορά με ενθαρρυντική ακρίβεια, με σημαντικά μικρότερο υπολογιστικό κόστος σε σχέση με τις παραδοσιακές αναλύσεις υπολογιστικής ρευστοδυναμικής. Οι αναλύσεις ευαισθησίας αναδεικνύουν την επίδραση της πυκνότητας δεδομένων, των οριακών συνθηκών και της αρχιτεκτονικής του μοντέλου στην προβλεπτική απόδοση.

Τα ευρήματα επιβεβαιώνουν την αποτελεσματικότητα των εργαλείων αεροδυναμικής επιταχυνόμενων από AI και καταδεικνύουν την προοπτική τους στην επιτάχυνση των διαδικασιών σχεδιασμού, στην υλοποίηση εφαρμογών σε πραγματικό χρόνο και στην υποστήριξη στρατηγικών ελέγχου. Προοπτικές περαιτέρω έρευνας εντοπίζονται σε υβριδικές μεθόδους CFD–AI, σε βελτιωμένη μοντελοποίηση τύρβης και σε μεγάλης κλίμακας υποκατάστατα μοντέλα.

Λέξεις-κλειδιά: Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Αεροδυναμική, Υπολογιστική Ρευστοδυναμική (CFD), Δεδομένα Εκπαίδευσης, HB2, AGARD

FUNDING: This thesis has been funded by Ansys Inc. and was conducted during my internship in the Greek offices.

Contents

LIST OF TABLES

LIST OF FIGURES

1. Introduction

1.1 General Information

As the world of technology accelerates at an unstoppable pace, industries are continuously looking for faster and more efficient solutions to complex engineering problems. Aerospace engineering, particularly, is a sector that demands rapid and accurate results in all the fields of research that are involved in it, including aerodynamics, while also keeping low development costs. Aerodynamic analysis methods such as wind tunnel testing and computational fluid dynamics (CFD) simulations are remarkably accurate and have marked high levels of evolution through the years, however they can be time consuming and computationally expensive. Artificial intelligence (AI) and machine learning (ML) are two very significant engineering tools that are emerging as solutions to this problem, since they can perform reliable aerodynamic predictions in multiple real – life applications.

The purpose of the current thesis is to focus on the generation of high-quality training data for an AI demonstrator designed to make valid predictions of specified quantities, for applications in aircraft aerodynamics. A structured methodology must be followed to finally generate solid, reliable datasets, using model design, meshing workflows and CFD simulations over various aircraft geometries and flight conditions. By the end of the project, the goal is to have an AI model that, through testing and validation, has proved that it can be used for predicting aerodynamic behaviors in intermediate angles of attack, different Mach numbers and different unseen configurations.

Three models were used for this study. The first model is the HB – 2 generic ballistic model, an industry standard for validation cases, on which a mesh independence study was conducted to define the optimal computational mesh. The results of this study were used on the second, custom model of a high-speed maneuverable vehicle, used for transonic flights of trajectory training and data sampling. The purpose of this model training is to test the behavior of the AI predictions in intermediate geometric angles of the vehicle as the angle of attack remains the same. The third model that is used is the AGARD – B standard wind tunnel model, which is again used for a mesh independence study and extended AI model training. Intermediate, custom configurations of this model, in various operating conditions were used for the training of the AI model that is finally used for predicting the unseen geometry of the AGARD – C generic missile.

This thesis was conducted during an internship program at Ansys Hellas. The main tools that were used for simulations and the AI model training were the Ansys Fluent Meshing, Ansys Fluent Solver, Ansys Discovery and the Ansys SimAI platform.

1.2 HB – 2 Model Description

The HB – 2 generic ballistic missile is a well-known blunt-cone-cylinder-flare body that has been widely used as a validation case for external aerodynamic studies. The model has been used in numerous wind tunnel facilities all over the world and through the decades, an extensive amount of reliable experimental data has been obtained for both supersonic and hypersonic test cases.

While a lot of experimental investigations have been conducted, there are limited CFD studies on the HB-2 configuration, especially for hypersonic flows. For the first part of this study, the existing transonic and supersonic validation cases are going to be considered sufficient.

The HB-2 model is axisymmetric, with a 25º nose cone half angle and a 10º tail flare, used to limit the impact of the viscous effects of the model. Smooth radius fairings are connecting the cylindrical part with both the nose and the flare. The total model length is defined as 4.9D, while the momentum center is located at 1.95D from the nose. For this application, a model diameter of 100mm was selected. The schematic of the HB-2 configuration is presented in Figure 1: Schematic of the HB-2 Body and the final design with the coordinate system is in Figure 2: Coordinate System of the HB-2 Body.

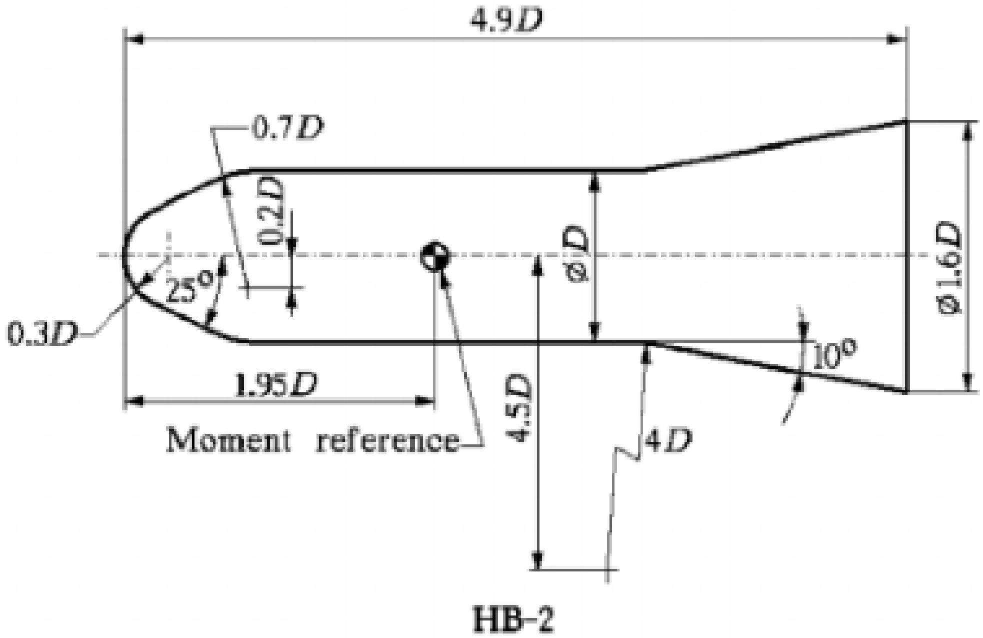


Figure 1: Schematic of the HB-2 Body

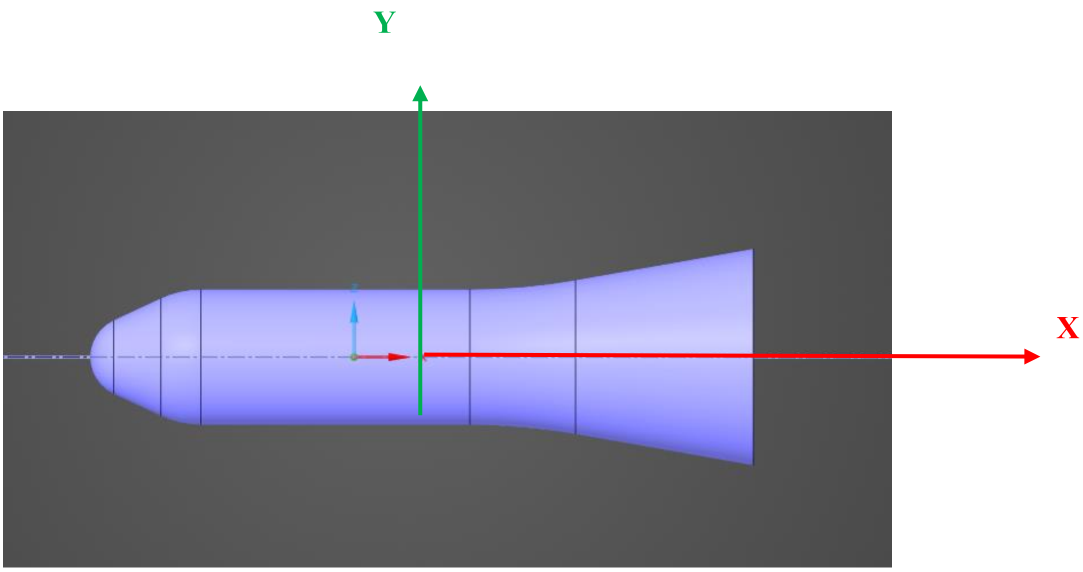


Figure 2: Coordinate System of the HB-2 Body

In the first part of this study, the HB-2 model is employed to develop a meshing workflow and finally conduct a mesh independence study in different angles of attack (AoA). The experimental values of the aerodynamic coefficients serve as reference values against which the computational fluid dynamics (CFD) results are compared.

Proceeding with this step, the forebody of the HB-2 model, as described later, is integrated into the design of a custom maneuverable configuration in ten different geometric angles.

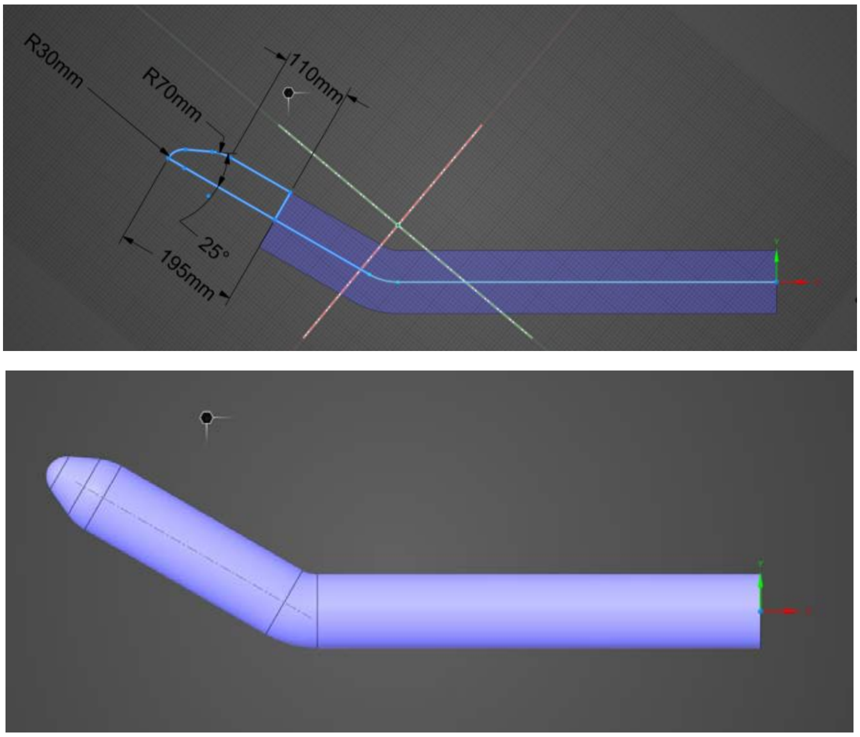


Figure 3: Schematic of the Maneuverable Vehicle in 15 Deg

This configuration represents a maneuverable vehicle, which can perform a high – rate pivoting of the forebody, because of its articular – jointed neck. The results of the HB-2 mesh independence study are applicated to the meshing of the geometric variations of this model, establishing reliable results.

1.3 AGARD Models Description

The AGARD B and C Models are a group of aerodynamic models that were developed by the Advisory Group for Aerospace Research and Development, which is a segment of NATO (North Atlantic Treaty Organization). These models have been widely used in the aerospace industry for many decades, and they have been included in numerous tests for predicting the aerodynamic behavior of aircraft designs. They provide a significant amount of experimental data that can be instrumental in aerodynamic studies in aerospace engineering, especially when computational fluid dynamics (CFD) may not be practical, usually in early design stages.

The AGARD B calibration model is one of the most popular standard models used in aerodynamic – validation cases. It is a configuration with a cylindrical body and an ogival nose and a delta wing, representing a high-speed aircraft. The reference area that was used for the calculation of the aerodynamic coefficients is considered to be the theoretical delta wing, while the reference length for the pitching moment coefficient is the mean aerodynamic chord. The aerodynamic center is located at a point 50% of the mean aerodynamic chord. The model length is defined as 8.5D.

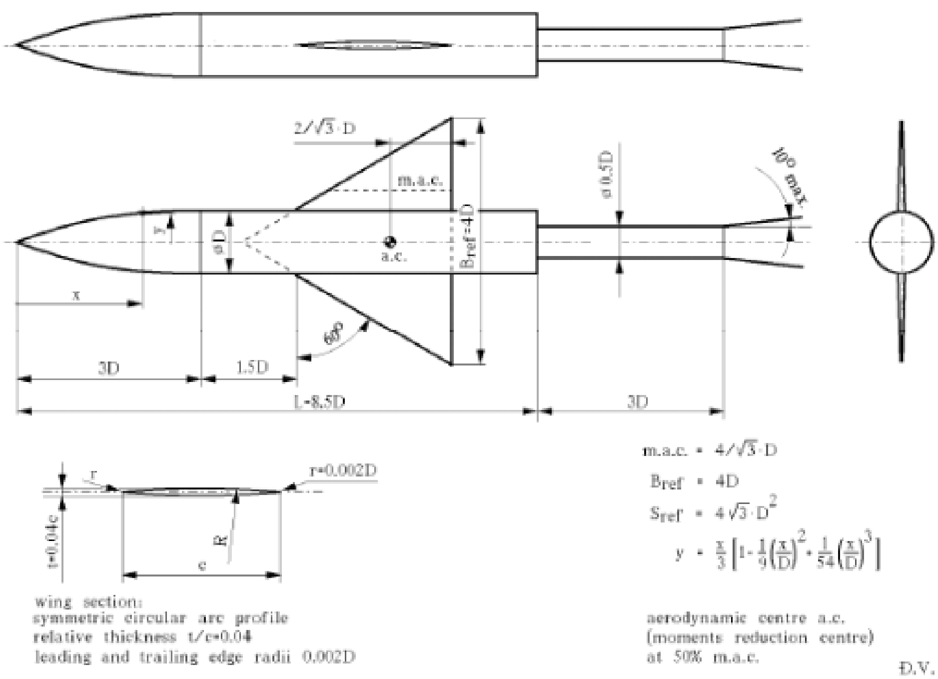


Figure 4: Schematic of the AGARD B Model

Two diameters, 115,8 mm and 178 mm, have been used for the testing of the AGARD model in the T-38 and T-35 wind tunnels. The 115.8 mm diameter model has been used for the calibration of VTI's T-38 wind tunnel in subsonic, transonic and supersonic speed ranges, up to Mach 2, while the 178 mm diameter model has only been used in the T-35 wind tunnel at VTI, up to Mach 0.5. The model diameter that was selected for this study is the 115.8 mm model.
The AGARD C standard calibration model has been designed primarily for testing and calibration in the transonic speed range. It is identical to the AGARD B model, with the only differences being the body extending over 1.5D aft and the addition of two tail surfaces, a vertical and a horizontal one. The AGARD C geometry can be found in the 1961 book "A Review of Measurements on AGARD Calibration Models".

In the second part of the current study the AGARD B model is going to be used for the validation study of the meshing workflow that will be used. On the next step, a combination of custom variations, between AGARD B and AGARD C, is going to be designed and simulated in different aerodynamic conditions, aiming to be used for the generation of an AI demonstrator for aircraft aerodynamics. Finally, the AGARD C model will be used as a testing point for this demonstrator, being introduced to the AI model as an unseen geometry, giving us the opportunity to evaluate the performance of the platform by comparing the results to both CFD and experimental data.

1.4 Thesis Structure

Chapter 1 lies in providing an introduction of the content of the thesis. A general review of the underlying context is presented, while also the objectives and the scope of the work are being discussed thoroughly. An overview of the thesis structure completes this chapter.
Chapter 2 includes a brief overview of the aerodynamics theory for aircraft cases and the CFD theory that applies to external aerodynamics flow simulations.
Chapter 3 describes the practical setup of CFD simulations for the HB-2 and AGARD models. It details meshing strategies, metrics, and workflow steps, including input geometry, solver setup, and mesh independence studies. The procedures for both model types are outlined, ensuring reproducibility and accuracy of the computational results.
The next chapter presents SimAI, an AI-based platform for aerodynamic prediction. It covers machine learning fundamentals, data generation, model construction, and training procedures for parametric and non-parametric models. Results for maneuverable vehicles and AGARD

models are compared, highlighting the effectiveness of AI in aerodynamic analysis and discussing performance variations with different data sets.

Chapter 5 summarizes the key findings of the thesis, emphasizing insights gained from CFD simulations and AI modeling. It also suggests directions for future research, including potential improvements in AI approaches, model fidelity, and aerodynamic prediction methodologies.

The appendix provides supporting mathematical derivations and additional technical details, including the continuity equation, Navier–Stokes derivation, Reynolds decomposition, and turbulence model formulations.

2. Aerodynamics and CFD Theory

In this chapter the general theory that is being applied in this study is going to be presented in two sections. The first section includes the basic aircraft aerodynamics theory, the definition of the aerodynamic forces and coefficients that will be used in this study. A short reference to the boundary layer theory and the key characteristics of shock waves in transonic and supersonic applications will complete the first category. The second refers to the Computational Fluid Dynamics (CFD) theory and methodology of this research. It begins with the definition of the RANS (Reynolds - Averaged Navier – Stokes) equations and proceeds with an analysis of the fundamental turbulence models and numerical methods that applied in the Ansys Fluent solver for the simulations of this current research.

2.1 Aircraft and Missile Aerodynamics

Aerodynamics has a significant role in the design and performance optimization stages of aviation systems nowadays. As a vehicle moves through the air, forces act on its surfaces, affecting some key parameters, such as drag, lift, pitching moment and motion stability, which subsequently determine a part of the overall performance. Drag force opposes the forward motion of the vehicle, impacting acceleration and deceleration. Lift force on the other hand, acting perpendicularly to the upcoming airflow, is the force that causes the vehicle to stay in the air.

In both aircraft and missile aerodynamic studies, it is important to understand the common principle of relative motion. Both in wind tunnel testing and computational fluid dynamics (CFD) simulations, rather than considering that the aircraft is moving through stationary air, it is assumed that the air flows over the stationary vehicle, having the same effect as in the opposite case.

2.1.1 Aerodynamic Forces

When a vehicle moves relatively to the surrounding air, an aerodynamic force is generated, acting on its center of pressure in the rearward direction and at an angle specified by the properties of the motion. This aerodynamic force can be described as a combination of two major force components, the drag force and the lift force.

Drag acts in the direction that is parallel to the relative motion of the vehicle and can be broken down to:

- Viscous Drag, or skin friction drag, which is caused by the friction between the air and the surface of the vehicle.
- Pressure Drag, or form drag, which is a result of the boundary layer separation and the wake that is created afterwards.

The general formula of drag force is defined as follows:

$$Drag = C_D \frac{1}{2} \rho U^2 A$$

Where:

- C_D is the coefficient of drag force
- ρ is the air density
- U is the speed of the vehicle
- A is the frontal area of the vehicle

Lift is the aerodynamic force component that acts perpendicularly to the oncoming airflow. It is defined as:

$$Lift = C_l \frac{1}{2} \rho U^2 A$$

Where:

- C_l is the coefficient of lift force

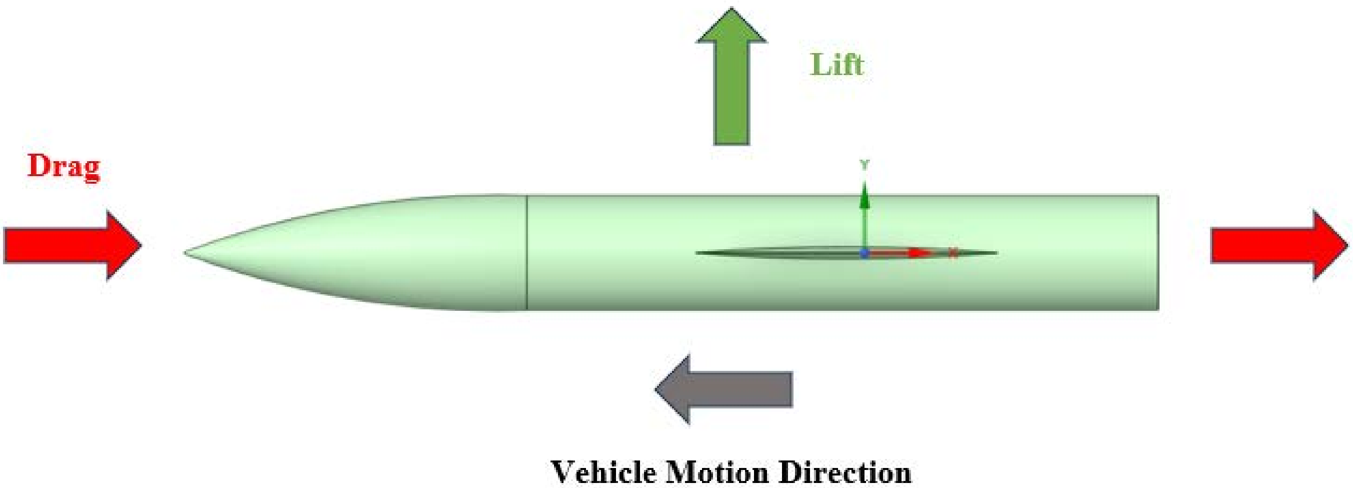


Figure 5: Aerodynamic Forces acting on the AGARD – B vehicle

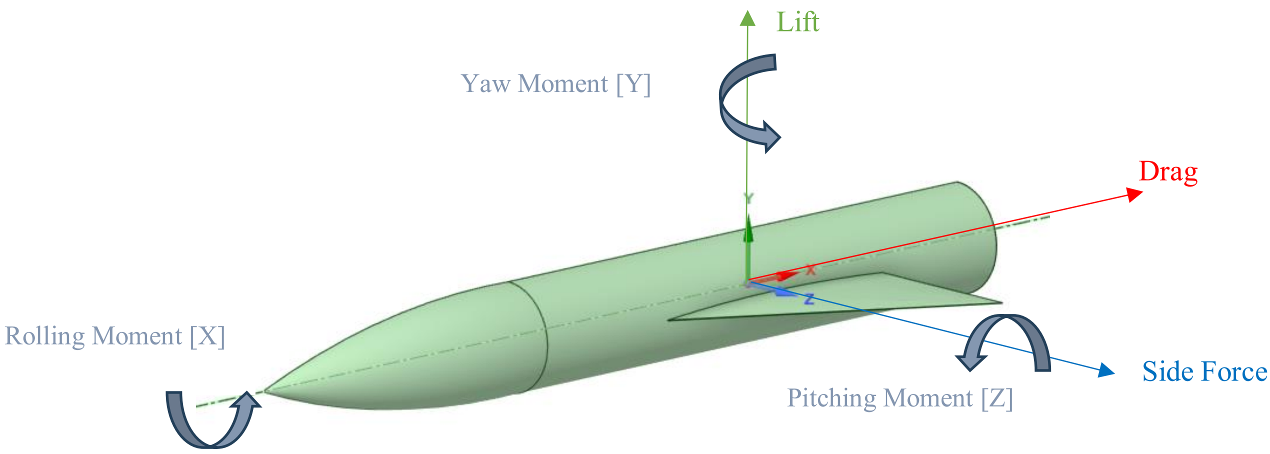


Figure 6: Aerodynamic Forces acting on the AGARD – C vehicle

2.1.2 Boundary Layer Theory

Flow separation is one of the main contributors to aerodynamic drag in moving vehicles and it requires basic understanding of boundary layer.

As fluid flows over an object, air molecules in the immediate vicinity of the object surface stick to it due to molecular viscosity, an action that causes the velocity to be zero at the surface, forming a thin boundary layer. This deceleration of the flow is then propagated subsequently to successive neighboring layers of flow, next above the first one, by viscous shear stresses between them. The shear stresses are weakened since the layers are progressing farther from the surface and the velocity is increasing, until it finally reaches the value of freestream velocity. This thickness, δ, of this set of developed layers normal to the surface, from the surface to the point where the velocity is equal to 99% of the freestream velocity, depicts the profile of a boundary layer.

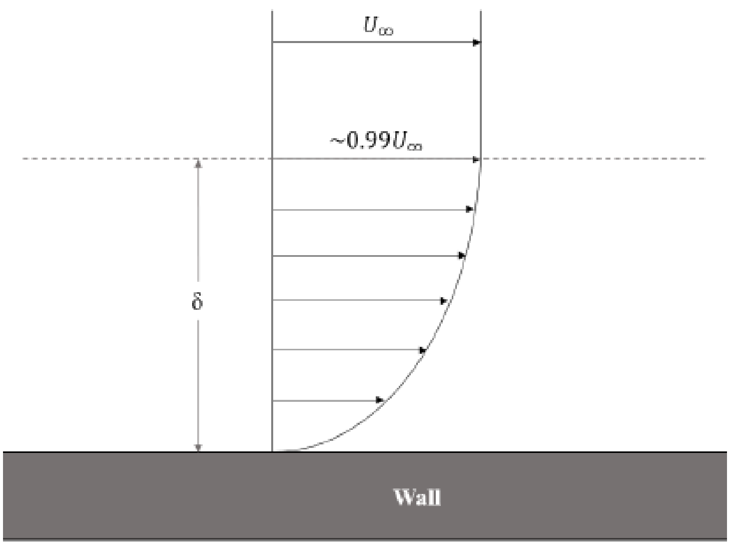


Figure 7: Velocity profile of a laminar boundary layer.

The boundary layer is a thin region near a surface dominated by viscous effects. At low Reynolds numbers (Rex<500,000), the flow is laminar with smooth, orderly velocity profiles. As the Reynolds number increases further down the pipe, the flow becomes unstable and forms a turbulent flow with unsteady, three-dimensional motion.
Pressure gradients exert a major influence on the boundary layer: positive (adverse) gradients slow down or reverse the flow, leading to separation and drag increase, while negative (favorable) gradients accelerate the flow and maintain attachment. Turbulent boundary layers are less prone to separation than laminar boundary layers, and separated laminar flows will reattach when they turn turbulent or when they encounter a favorable pressure gradient.

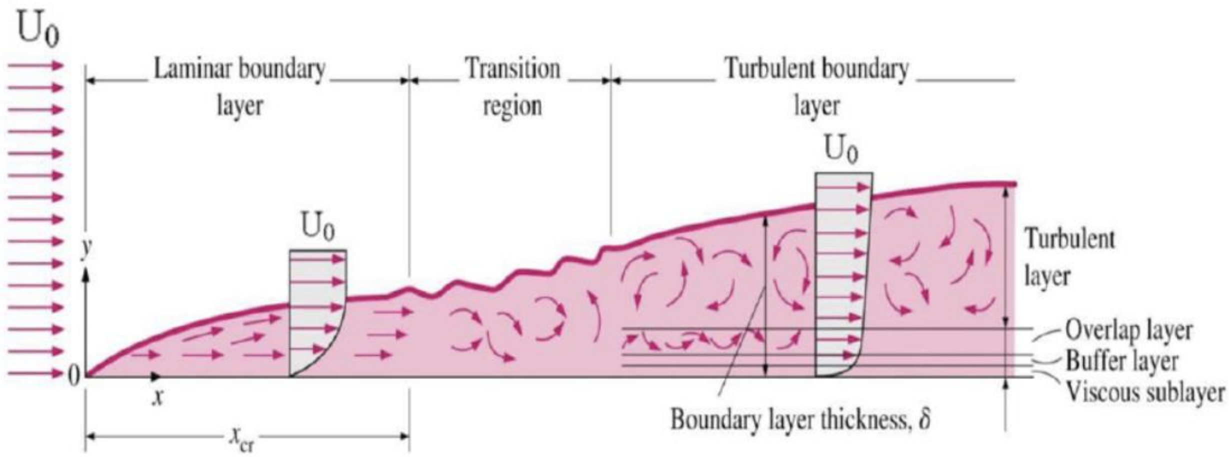


Figure 8: *Boundary layer transition from laminar to turbulent.*

In high-speed (transonic and supersonic) regimes, boundary layer behavior interacts with shock waves, further complicating the aerodynamic environment. Shock–boundary layer interactions can cause abrupt separation, significantly altering drag and lift forces. These effects are critical in both aircraft and missile configurations, such as those modeled in the AGARD–C case.

2.1.3 Shock Waves

Shock waves are a fundamental phenomenon of compressible fluid dynamics, arising whenever a body travels through a medium, such as air, at a speed equal to or greater than the speed of sound. In these circumstances, the airflow around the body is no longer able to flow smoothly out of the way, and a sharp discontinuity is formed in the flow field. Across this shock front, the flow properties, i.e., pressure, temperature, density, and velocity, change almost instantaneously. In aircraft aerodynamics, shock waves typically develop on wings, fuselage surfaces, or engine inlet lips in transonic and supersonic flight regimes. Their presence is associated with localized regions where the flow changes from supersonic to subsonic speeds, often generating additional drag, known as wave drag, and potentially adverse effect on stability and control. Detection and analysis of shock waves are extremely significant for aerodynamic shape and performance optimization. Common detection methods include experimental techniques like schlieren or shadowgraph imaging that visualize density gradients, and pressure-sensitive paint and embedded sensors that measure rapid surface pressure changes. Computational fluid dynamics (CFD) simulations are also an important method for detecting and characterizing shock structures so that engineers can anticipate their impact and implement design features to negate their effect.

In this thesis the shock wave detection techniques that were used are Schlieren visualization contours and static pressure contours and plots of the 3D flowfield.

2.2 CFD Methodology

2.2.1 Navier - Stokes Equations

The Navier – Stokes equations are a set of governing partial differential equations that describe the motion of fluids. They represent conservation of mass, momentum and energy to mathematically model fluid flow phenomena.

The continuity equation for a finite control volume, fixed in space, in partial differential form is described as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho V) = 0 \tag{2.1}$$

For the momentum equation, the x, y, z components are given respectively as:

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u V) = -\frac{\partial p}{\partial x} + \rho f_x + (F_x)_{viscous} \tag{2.2}$$

$$\frac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho v V) = -\frac{\partial p}{\partial y} + \rho f_y + \left(F_y\right)_{viscous} \tag{2.3}$$

$$\frac{\partial(\rho w)}{\partial t} + \nabla \cdot (\rho w V) = -\frac{\partial p}{\partial z} + \rho f_z + (F_z)_{viscous} \tag{2.4}$$

In each of these momentum equations, the first term on the left - hand side represents the rate of change of momentum, due to unsteady, time – dependent flow behavior inside the control volume. The second term is the net flow of momentum through the control surface. On the right

- hand side, the first term represents the pressure forces, the second term the body forces and the third term corresponds to the total viscous forces.

For incompressible flow, the continuity equation takes the following form:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{2.5}$$

And the Navier – Stokes equations can be written as:

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)=-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right) \tag{2.6}$$

$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\right)=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right) \tag{2.7}$$

$$\rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)=-\frac{\partial p}{\partial z}+\mu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right) \tag{2.8}$$

2.2.2 Reynolds - Averaged Navier - Stokes (RANS) Equations

The Reynolds-Averaged Navier–Stokes (RANS) equations are derived by time-averaging the fundamental equations of fluid motion. These are particularly useful in analyzing turbulent flows, where small-scale fluctuations are averaged out to focus on the behavior of the larger-scale flow. This process also involves modeling how these small fluctuations influence the overall motion, as their nonlinear effects can significantly impact the larger flow structures.

To illustrate, take a velocity component u_i that contains both mean and fluctuating parts. For statistically steady flows, a time-averaging process can separate the fluctuating components from the mean. This method, known as Reynolds decomposition (explained in [A3. Reynolds decomposition]), leads to the RANS equations.

After applying the Reynolds decomposition technique (described in [A3. Reynolds decomposition]), the RANS equations are derived as:

$$\underbrace{\rho\left(\frac{\partial\bar{u}}{\partial t}+\bar{u}\frac{\partial(\bar{u})}{\partial x}+\bar{v}\frac{\partial(\bar{u})}{\partial y}+\bar{w}\frac{\partial(\bar{u})}{\partial z}\right)}_{\textbf{A}}+\underbrace{\rho\left(\frac{\partial\overline{u'u'}}{\partial x}+\frac{\partial(\overline{v'u'})}{\partial y}+\frac{\partial(\overline{w'u'})}{\partial z}\right)}_{\textbf{B}}$$

$$= \underbrace{-\frac{d\bar{p}}{dx}}_{\mathbf{C}} + \underbrace{\mu\left(\frac{\overline{\partial^2\overline{(u)}}}{\partial x^2} + \frac{\overline{\partial^2\overline{(u)}}}{\partial y^2} + \frac{\overline{\partial^2\overline{(u)}}}{\partial z^2}\right)}_{\mathbf{D}} \tag{2.9}$$

Each term has a physical interpretation:

- A: Change in momentum over time.
- B: Contribution from Reynolds stresses caused by turbulent fluctuations.
- C: Pressure gradient (isotropic stress).
- D: Viscous stresses due to fluid viscosity.

In a more general, tensor form, the RANS equations can be written as:

$$\rho\frac{\partial\bar{u}_i}{\partial t} + \rho\bar{u}_j\frac{\partial\bar{u}_i}{\partial x_j} = \left(\rho\bar{f}_i\right) + \frac{\partial}{\partial x_j}\left[-\bar{p}\delta_{ij} + 2\mu\overline{S_{ij}} - \rho\left(\overline{u_i'u_j'}\right)\right] \tag{2.10}$$

where:

- $\overline{S_{ij}} = \frac{1}{2}(\frac{\partial\overline{u_i}}{\partial x_j} + \frac{\partial\overline{u_j}}{\partial x_i})$ is the mean rate of strain tensor.
- $\rho(\overline{u_i'u_j'})$ represents the Reynolds stress tensor.

Solving the RANS equations requires knowledge of the Reynolds stress terms, which are not directly available from the averaging process. Determining these stresses introduces what is known as the closure problem, a major challenge in turbulence modeling that requires additional modeling assumptions or equations to close the system and solve for the mean flow.

2.2.3 Businessq Approach

The Boussinesq hypothesis is a common method of modeling Reynolds stress in the RANS equations. The hypothesis assumes that the turbulence momentum transport, because of the turbulent, eddies within the flow, can be modeled similarly to molecular viscosity but in terms of an effective turbulent or eddy viscosity (μ_t). The procedure asserts that the Reynolds stress tensor is proportional to the deviatoric part of the mean strain rate tensor. This leads to the equation:

$$\tau_{ij} = 2\mu_t S_{ij}^* - \frac{2}{3}\rho k \delta_{ij} \qquad (2.11)$$

where Sij is the trace-free strain rate tensor, k is the turbulent kinetic energy, and δij is the Kronecker delta. In more practical terms, the Reynolds stress is modeled as:

$$-\rho\left(\overline{u_i' u_j'}\right) = \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho k \delta_{ij} \qquad (2.12)$$

This formulation enables closure of the RANS equations by linking turbulence-induced stresses to mean flow gradients, forming the foundation for many eddy viscosity models such as k-ε and k-ω.

2.2.4 Turbulence Models

Turbulence models are used to simulate the influence of turbulent flow without necessarily solving all the details of the turbulence itself. Because turbulence is a nonlinear process, the fluctuating components of the flow interact with the mean flow in complex ways. Upon averaging of the governing equations, these couplings show up as additional stress terms, termed often as turbulent stresses. Turbulence models aim to simulate these stresses. In fluid flow, the large-scale vortices are typically created by the geometric walls themselves, while the small ones tend to be more regular, which can be isotropic and primarily responsible for energy loss. Since the large-scale vortical structure significantly affects the mean flow behavior, they need to be well-resolved. Nevertheless, all the scales of turbulent motion, most importantly the small ones, are not usually needed to be resolved in most engineering problems.

The selection of an appropriate turbulence model depends upon several parameters: the specific nature of the flow, standard modeling practices for similar problems, the accuracy desired, available computing capability, and available computation time.

2.2.4.1 k-epsilon (k-ε) model

The k-ε models are widely used two-equation turbulence models that introduce transport equations for the turbulent kinetic energy (k) and its dissipation rate (ε). These equations allow for the calculation of eddy viscosity, which models the effects of turbulence on the mean flow. The transport equation for k includes terms representing unsteady flow (time evolution - A), convection (B), diffusion (C), and various sources or sinks (D), such as turbulence production from velocity gradients (G_k), buoyancy effects (G_b), and a compressibility-related term (Y_M).

Equation for k:

$$\underbrace{\frac{\partial(\rho k)}{\partial t}}_{\mathbf{A}} + \underbrace{\nabla \cdot (\rho u k)}_{\mathbf{B}} = \underbrace{\nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right]}_{\mathbf{C}} + \underbrace{G_k + G_b - \rho\varepsilon - Y_M + S_k}_{\mathbf{D}} \tag{2.13}$$

The ε equation follows a similar structure but includes additional modeling constants (C_1, C_2, C_3) and accounts for the rate at which turbulent energy dissipates. Both equations also contain user-defined source terms (S_k and S_ε), and the diffusion terms are influenced by the turbulent Prandtl numbers (σ_k and σ_ε).

Equation for ε:

$$\underbrace{\frac{\partial(\rho\varepsilon)}{\partial t}}_{\mathbf{A}} + \underbrace{\nabla \cdot (\rho u \varepsilon)}_{\mathbf{B}} = \underbrace{\nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\nabla \varepsilon\right]}_{\mathbf{C}} + \underbrace{C_1 \frac{\varepsilon}{k}(G_k + C_3 G_b) - C_2 \rho \frac{\varepsilon^2}{\kappa} + S_\varepsilon}_{\mathbf{D}} \tag{2.14}$$

The eddy viscosity (μ_t), which links turbulence to momentum transfer, is calculated using:

$$\mu_t = C_\mu \frac{k^2}{\varepsilon} \tag{2.15}$$

where $C\mu$ is a model constant. This formulation makes efficient prediction of wall-bounded flows, external aerodynamics, and jet or wake flows possible where turbulence is predominantly responsible for performance. Although it may not simulate intricate separation as efficiently as more advanced models like SST k-ω, it remains a robust and reliable choice for steady-state aerospace calculations with high-Reynolds-number flows.

2.2.4.2 Standard k-omega (k-ω) model

The standard k-ω model is a two-equation turbulence model that is formulated on the basis of solving transport equations for the specific dissipation rate (ω) and for the turbulent kinetic energy (k), where ω is obtained as the ε to k ratio scaled by some constant:

$\omega = \frac{\varepsilon}{C_\mu k}$, where $C_\mu = 0.09$

Equation for k:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho k u_i)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + G_k - Y_M + S_k \tag{2.16}$$

Equation for ω:

$$\frac{\partial(\rho\omega)}{\partial t} + \frac{\partial(\rho\omega u_i)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\frac{\partial k}{\partial x_j}\right] + G_\omega - Y_\omega + S_\omega \tag{2.17}$$

The turbulent viscosity is calculated by the model from the relation:

$$\mu_t = \alpha^* \rho \frac{k}{\omega} \tag{2.18}$$

With "a*" being a model constant. There are terms for unsteady and convective transport, diffusion, generation due to velocity gradients (G_k), and dissipation in the k equation. There are terms for all of these except generation in the ω equation, but for ω it is G_ω and Y_ω its dissipation. Source terms (S_k, S_ω) can be included if needed. This model is particularly robust near solid walls, hence highly relevant in aerospace CFD simulations of boundary layers, shockwave – boundary layer interactions and attached or mildly separated flows. Its sensitivity to near-wall behavior provides high accuracy in regions where accuracy in turbulence modeling is critically important, e.g., turbine blades, nozzles, or high-lift airfoils.

2.2.4.3 Shear – Stress Transport (SST) k – omega model

Shear-Stress Transport (SST) k-ω is a hybrid model that takes the benefits of both the k-ω and k-ε models. It employs the standard k-ω formulation in the vicinity of walls, with its ability to yield accurate solutions to boundary layer flows and separation, and transfers to a redefined k-ε model in the free stream, with improved performance away from surfaces. This switch is regulated by a blending function that constantly enables the proper model for the flow regime. The SST model further introduces a cross-diffusion term in the ω equation as well as alters the definition of turbulent viscosity to enhance shear stress transport representation. As a result of these advancements, SST k-ω is utilized widely in wall-bounded turbulence, separation, and flows with adverse pressure gradients and thus is the most popular and most widely used model in aerospace and automotive simulations were robustness and accuracy are equally important. In this thesis the SST k-omega was selected as the most suitable.

2.2.5 Solver Theory

2.2.5.1 Overview

ANSYS Fluent provides two main numerical methods for solving fluid flow problems: the pressure-based solver and the density-based solver. In both approaches, the velocity field is obtained by solving the momentum equations. In the density-based method, density is computed using the continuity equation, and pressure is determined through the equation of state. The pressure-based method, on the other hand, determines pressure by solving a pressure or pressure correction equation derived from a combination of the continuity and momentum equations.

Although the density-based solver is typically favored for compressible aerodynamic flows, especially in the transonic regime, the pressure-based solver was chosen for this work due to its superior convergence behavior during simulation at Mach 0.8. This decision reflects practical consideration, as numerical stability and solution robustness were prioritized over strict adherence to the conventional solver-selection guidelines.

Fluent uses a control-volume-based technique to solve the integral forms of the conservation equations for mass and momentum. This involves:

- Discretizing the computational domain into control volumes using a mesh.
- Integrating the governing equations over each control volume to produce algebraic equations for variables such as velocity and pressure.
- Linearizing these equations and solving the resulting system to iteratively update the flow variables until convergence is achieved.

Given the performance and stability advantages observed, the pressure-based solver was selected for all simulations in this study.

2.2.5.2 Pressure – Based Solver

In the pressure-based solver, the continuity equation is enforced by solving a pressure equation derived from the continuity and momentum equations. This approach ensures that the pressure-corrected velocity field satisfies mass conservation.

Due to the nonlinear and coupled nature of the governing equations, the solution process is inherently iterative. Each iteration involves solving the full set of equations until residuals fall below a specified threshold, indicating convergence.

ANSYS Fluent provides two algorithmic options within this solver framework:

- The segregated algorithm, which solves equations sequentially.
- The coupled algorithm, which solves pressure and momentum equations simultaneously, often offering improved convergence in compressible or complex flow conditions.

2.2.5.3 Pressure – Based Segregated Algorithm

In the segregated algorithm, each governing equation—such as those for the velocity components (*u*, *v*, *w*), pressure (*p*), and turbulence variables (*k*, *ω*) are solved sequentially and independently. This means that when one equation is being solved, it is treated as decoupled by the others. The primary advantage of this method lies in its efficient memory usage, as only one discretized equation needs to be stored and processed at a time. However, this comes at the cost of slower convergence, as the decoupled treatment limits the interaction between variables during each iteration. Each iteration of the segregated algorithm consists of the following steps:

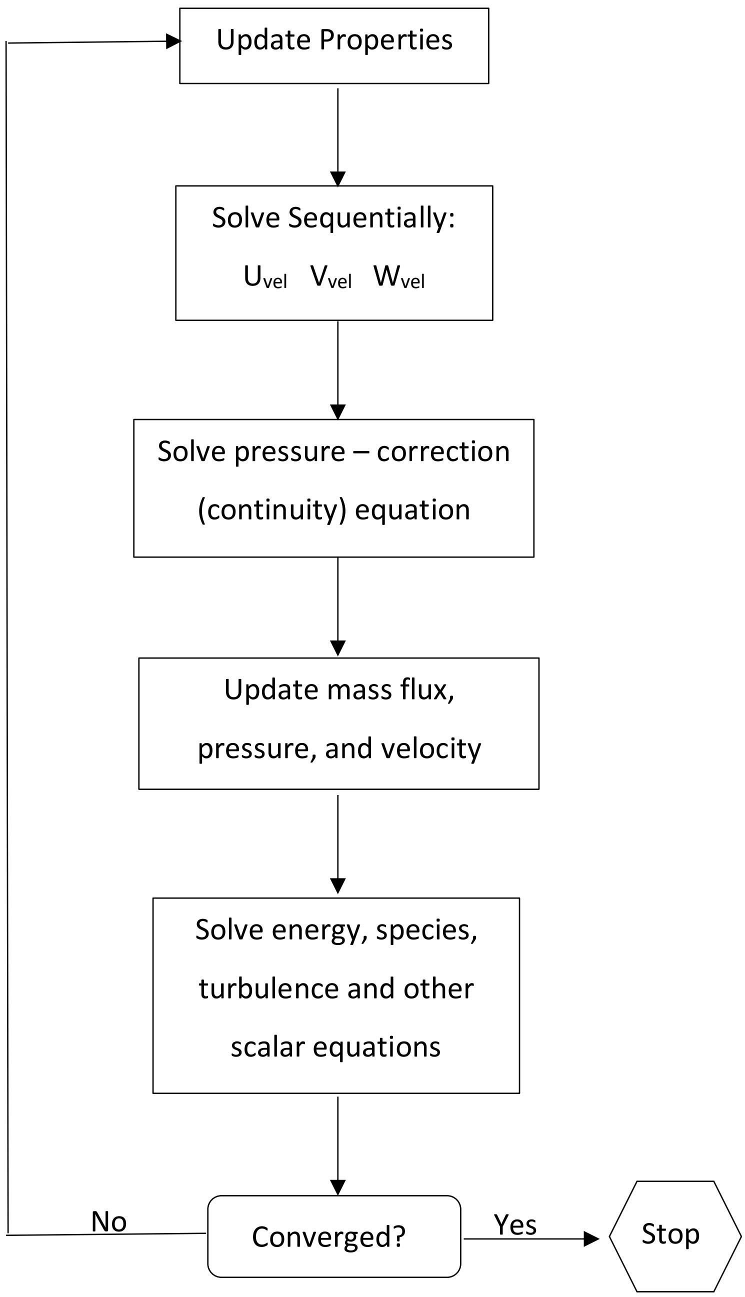


Figure 9: Pressure – Based Segregated Algorithm

2.2.5.4 Pressure – Based Coupled Algorithm

By contrast to the segregated, the coupled algorithm offers a more integrated approach, especially effective for steady-state, single-phase flows. In this scheme, the momentum equations and the pressure-based continuity equation are solved together as a single coupled system. This effectively replaces the separate momentum and pressure-correction steps seen in the segregated method with a unified solution step. The remaining transport equations—such as those for turbulence or energy—are still treated in a decoupled manner, similar to the segregated scheme.

Because the core flow equations are solved in a tightly coupled fashion, the convergence rate of the coupled algorithm is generally faster and more robust than that of the segregated approach. However, this improvement in performance comes with an increased memory demand, typically requiring 1.5 to 2 times more memory. This is due to the need to store and solve the entire coupled system of momentum and continuity equations simultaneously, as opposed to handling one equation at a time.

The pressure-based coupled algorithm iterations are illustrated in Figure 2.6.

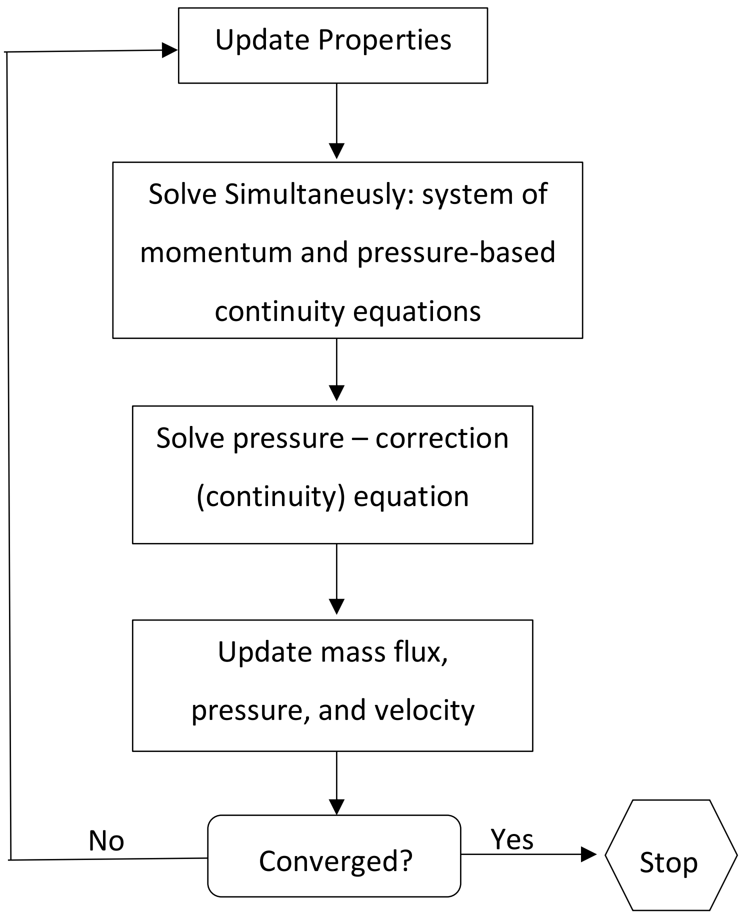


Figure 10: Pressure – Based Coupled Algorithm

2.2.5.5 Discretization

i) First Order Upwind Scheme

When first-order accuracy is desired, quantities at cell faces are determined by assuming that the cell-center values of any field variable represent a cell-average value and hold throughout the entire cell, meaning that the face quantities are identical to the cell quantities. Thus, when first-order upwind is selected, the face value, φ_f, is set equal to the cell-center value φ of in the upstream cell.

ii) Second Order Upwind Scheme

When second-order accuracy is desired, quantities at cell faces are computed using a multidimensional linear reconstruction approach. In this approach, higher-order accuracy is achieved at cell faces through a Taylor series expansion of the cell-centered solution about the cell centroid. Thus, when using second-order upwind discretization method, the face value φ_f is computed using the following expression:

$$\varphi_{f,SOU} = \varphi + \nabla\varphi \cdot \vec{r} \tag{2.19}$$

Where φ and ∇φ are the cell-centered value and its gradient in the upstream cell, and $\vec{r}$ is the displacement vector from the upstream cell centroid to the face centroid. This formulation requires the determination of the gradient ∇φ in each cell. Finally, the gradient is limited so that no new maxima or minima are introduced.

iii) High Order Term Relaxation

Higher order schemes can be written as a first-order scheme with additional terms for the higher order scheme. The higher-order relaxation may be applied for these additional terms. The under-relaxation of high order terms is according to the generic formulation for any generic property φ.

$$\varphi_{NEW} = \varphi_{OLD} + f(\varphi_{Intermediate} - \varphi_{OLD}) \tag{2.20}$$

Where f is the under-relaxation factor. Note that the default value of f for steady-state cases is 0.25 and for transient cases is 0.75. The same factor is applied to all equations solved.

iv) Pseudo Time Method Under-Relaxation

The pseudo time method is an implicit under-relaxation approach of advanced nature that adapts the relaxation parameter in a dynamic manner throughout simulation as a function of the response of the flow field. After the integration of the pseudo time method with the generic transport equation, the integral form of the governing equation for the steady-state calculations is of the nature:

$$\int_V \frac{\partial \rho\varphi}{\partial \tau} dV + \oint \rho\varphi\vec{v}\cdot\overrightarrow{dA} = \oint_S \Gamma_\varphi \nabla\varphi\cdot\overrightarrow{dA} + \int_V S_\varphi \, dV \tag{2.21}$$

Where τ denotes the pseudo time. After discretizing the governing equation using the finite volume method, the following algebraic form of the steady-state equation is obtained:

$$\rho_p \Delta V * \frac{\varphi_p - \varphi_p^{old}}{\Delta\tau} + \alpha_p\varphi_p = \sum_{nb} a_{nb}\varphi_{nb} + b \tag{2.22}$$

Where φ_{OLD} denotes the value of at the previous iteration, and $\Delta\tau$ is the pseudo time step size that can be estimated by characteristic length and velocity scales.

v) Hybrid Initialization

Initialization methods are used to create an initial pressure and velocity field over the entire computation domain, as required for the solution of partial differential equations, to establish a better starting point for the main calculation and reduce the number of iterations to achieve convergence. Hybrid Initialization is a combination of a number of methods. It computes the Laplace equation to generate a velocity field that is tailored to intricate domain geometries, and a pressure field that interpolates smoothly high and low pressure levels within the computational domain. It is preferable to other methods, including traditional initialization which used constant values across the entire field domain. Equations are at the Appendix.

vi) FMG Initialization

The FMG (Full Multigrid) initialization method is used to improve convergence efficiency. Unlike hybrid initialization, which generates an initial field from simple flow profiles, FMG computes a preliminary solution on a coarse grid and progressively interpolates it to finer grids. This multilevel approach captures both large-scale and local flow features from the outset, providing a more accurate initial guess. As a result, FMG typically requires fewer iterations on the fine mesh and offers greater robustness for flows with strong nonlinearities or shocks compared to hybrid initialization. In this thesis, this form of initialization was selected for usage, after providing faster convergence in comparison to the hybrid initialization method.

3. Numerical Setup

3.1 Meshing

This section describes the “Poly – Hexcore” meshes that were used for the simulations of all the models used for this study, as well as the mesh characteristics and mesh independence study results from both the HB-2 and the AGARD-B models.

3.1.1 Meshing Technology

Mosaic meshing technology, a patent-pending feature of Ansys Fluent, is designed for CFD simulations. It eases meshing by reducing the face count, improving cell quality, and achieving parallel scalability efficiently. It allows polyhedral connectivity among different mesh types.

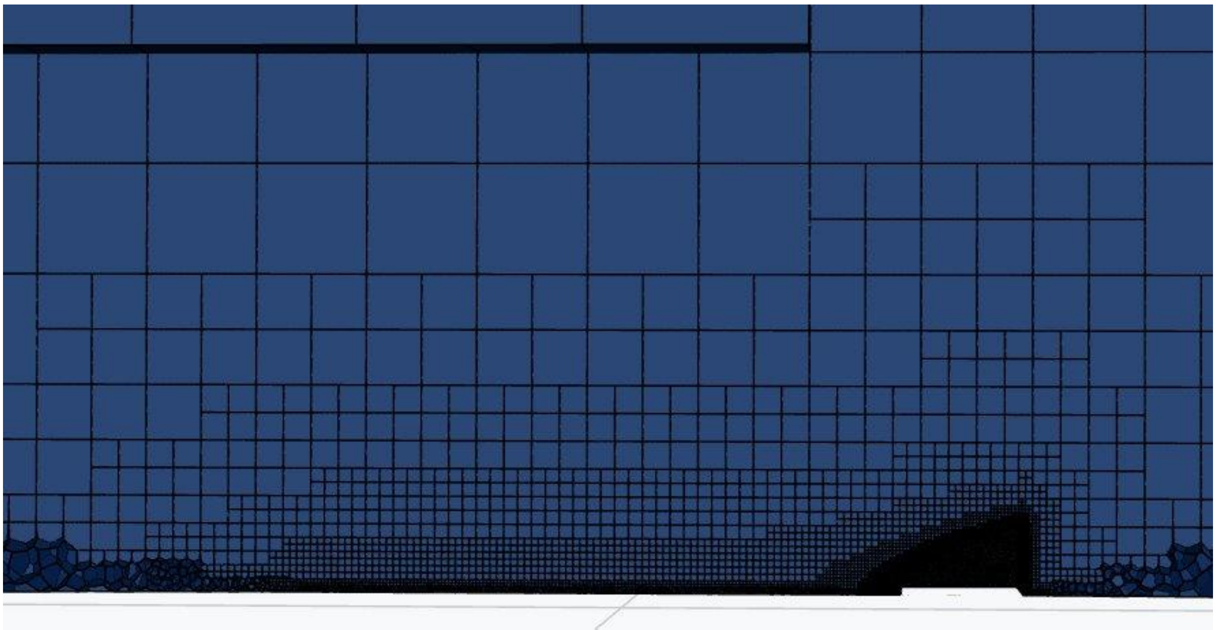


Figure 11: Mosaic Mesh Example

In this study, a "Poly-Hexcore" mesh was used. It combines high-quality hexahedral elements in the core flow region, polyhedral prismatic near-wall layers to capture boundary layer behavior properly, and efficient polyhedral transition between the prismatic and hexahedral regions. Compared to conventional hexahedral-tetrahedral meshes, this approach offers better element quality with fewer overall cells.

3.1.2 Metrics

3.1.2.1 Orthogonal Quality

In Fluent, **orthogonal quality** is a critical metric used to assess the alignment of mesh elements with respect to the flow field and solution gradients. It is defined based on the angle between the vector from the cell centroid to the face centroid and the face's normal vector. An orthogonal quality value of 1 indicates perfect alignment (i.e., orthogonality), where the face normal and the centroid vector are parallel. Values closer to zero represent increasingly skewed or poorly aligned cells, which can lead to numerical instability, poor convergence, and inaccuracies in the solution.

For ideal cell shapes, such as equilateral triangles or squares in 2D, and cubes or regular tetrahedra in 3D, the orthogonal quality approaches unity due to their symmetric geometry and uniform angles. In contrast, highly skewed or degenerate cells, such as narrow sliver tetrahedra or stretched hexahedra, tend to have much lower orthogonal quality values.

This metric is especially important in regions of high gradients, such as boundary layers or shock regions, where misaligned cells can distort the representation of physical phenomena. Orthogonal quality is applicable across all mesh types, including structured, unstructured, and polyhedral meshes, making it a universal and essential criterion for evaluating mesh suitability in computational fluid dynamics (CFD) simulations.

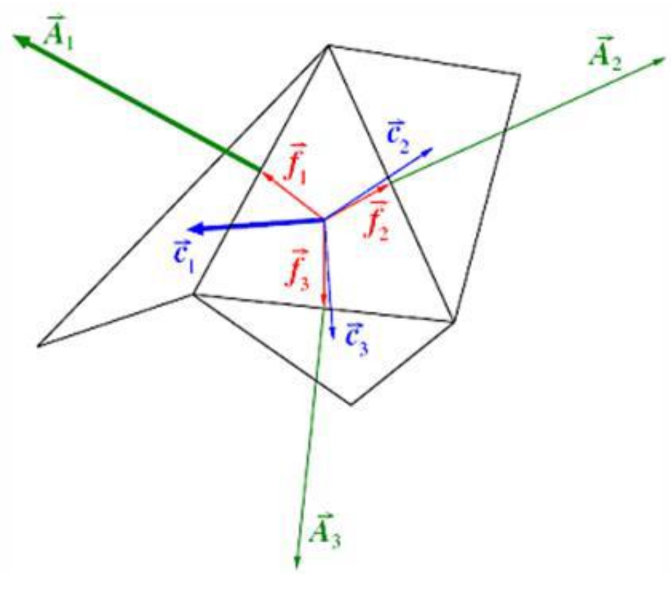


Figure 12: Orthogonality Vectors

3.1.2.2 *Aspect Ratio*

In Fluent, the aspect ratio is used to quantify the stretching of a cell. It is calculated as the ratio between the maximum and minimum values of the normal distances from the cell centroid to the centroids of its faces, as well as to its nodes, as illustrated in Figure 23. For a unit cell, the maximum distance is 0.866 and the minimum is 0.5, resulting in an aspect ratio of 1.732. This definition is applicable to all mesh types, including polyhedral meshes.

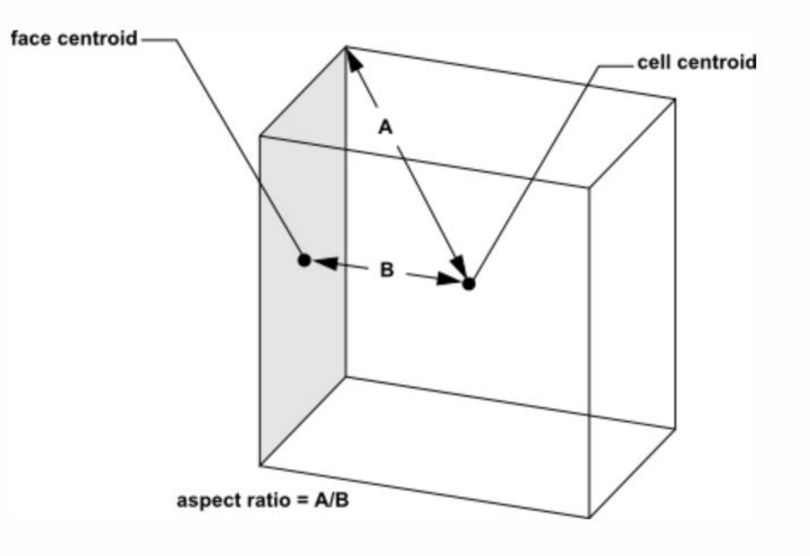


Figure 13: Aspect Ratio for a Unit Cube

3.1.2.3 *Surface Skewness*

Skewness is an indicator of how close to ideal a face or a cell is (equilateral or equiangular).

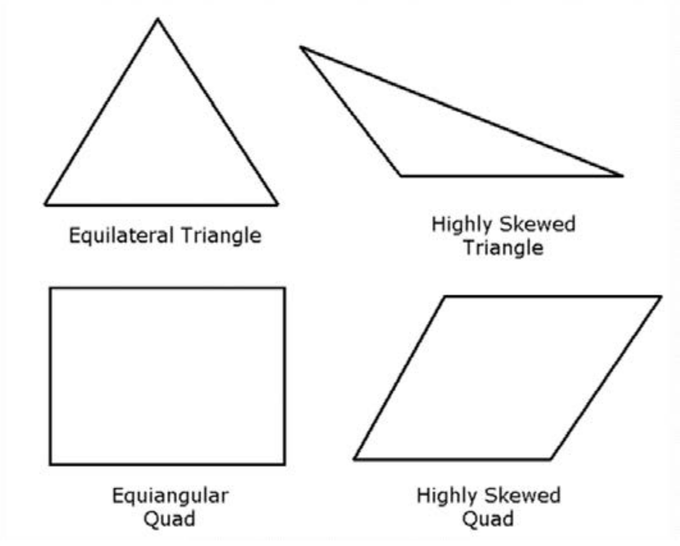


Figure 14: Skewness in different geometrical shapes

It is defined as:

$$Skewness = \frac{Optimal\ Cell\ Size\ \ -\ \ Cell\ Size}{Optimal\ Cell\ Size}$$

Where, the optimal cell size is the size of an equilateral cell with the same circumradius.

According to the definition, a value of 0 is an equilateral cell (best quality), and a value of 1 is a completely degenerate cell. Degenerate cells (slivers) are characterized as nodes being nearly coplanar. Cells with a skewness measure greater than 1 are not acceptable.

3.2 HB-2 Workflow

3.2.1 Input Geometry

The geometry selected for the meshing workflow was based on a half-model representation of the HB-2 generic missile. Utilizing a half-model takes advantage of the missile's geometric symmetry, significantly reducing computational cost without compromising the fidelity of the results. This approach is commonly employed in external aerodynamic simulations, especially when symmetry can be assumed in the flow behavior.

To encapsulate the missile and capture the relevant aerodynamic phenomena, a conical computational domain was constructed around the body. The conical shape, as well as the spherical one, is particularly well-suited for high-speed external flows, as it allows for natural expansion of the domain in the downstream direction, ensuring that shock waves, wake structures, and boundary layer interactions are fully resolved without artificial reflections or domain truncation effects. This configuration also supports smooth far-field boundary conditions, which is essential for accurate flow - field prediction around high-speed, axisymmetric bodies like missiles.

Overall, this setup ensures a balance between computational efficiency and physical accuracy, providing a robust foundation for generating high-quality meshes and performing reliable CFD simulations.

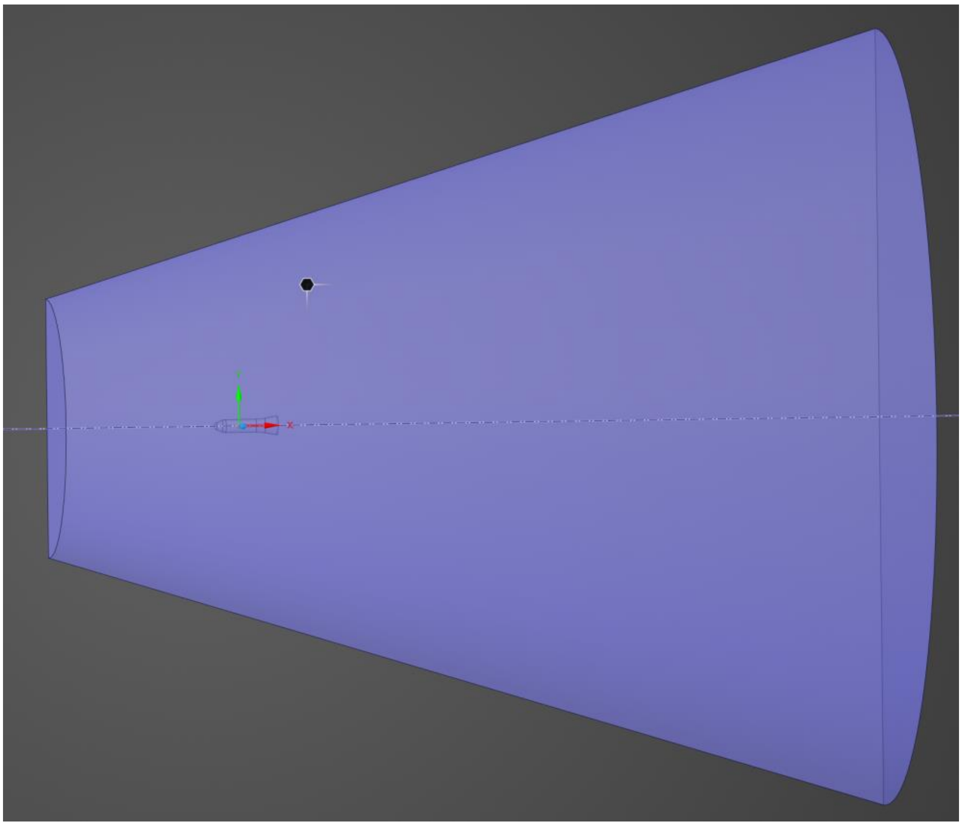


Figure 15: Computational Domain of HB-2

3.2.2 Mesh Characteristics

For the mesh generation, the “Watertight Geometry Workflow” in Fluent meshing was chosen. A triangulated surface mesh was generated after different cell sizes were specified for each surface part of the model, in order to capture sufficiently the geometry. A more refined mesh was needed for the definition of the HB-2 surfaces, while a larger mesh was chosen for the symmetry and the far – field domain.

For more curved surfaces, the "Curvature" size function was used, which computes edge and face sizes according to their size and normal angle parameters. It uses the normal angle parameter as the maximum allowed angle one element edge can cover.

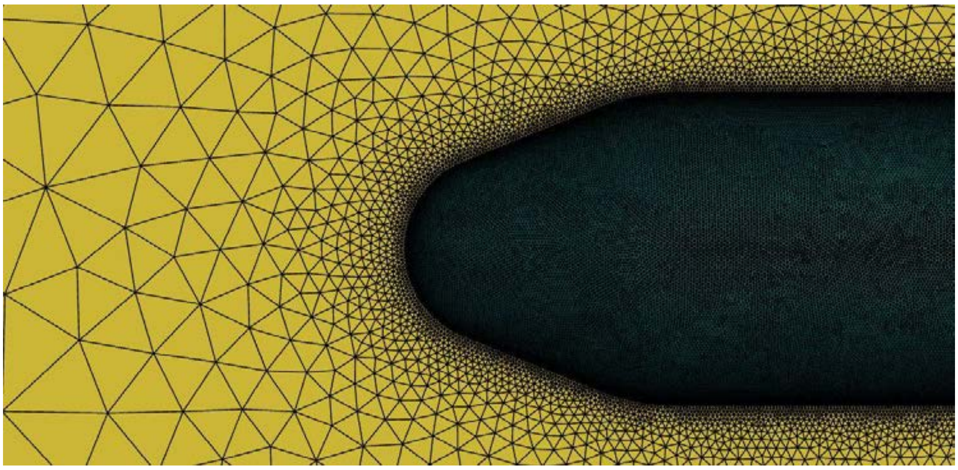


Figure 16: Triangular Surface Mesh

The BOI (Bodies of Influence) method was used to add extra refinement in the flow regions of the fluid domain, around the object and in the wake region. One box was added around the vehicle and was projected outwards and in the flow direction, aiming to capture all the desired areas.

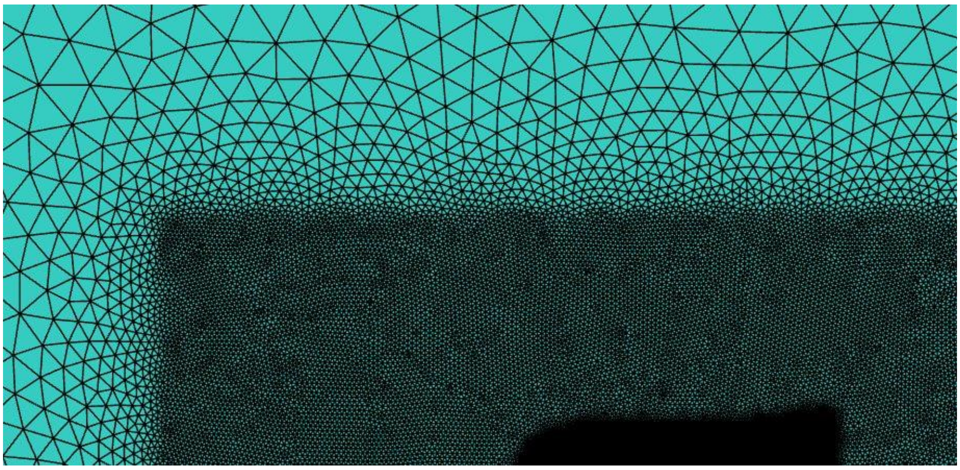


Figure 17: BOIs in Surface Mesh

Starting with a BOI cell size close to the surface cell size of the model and using the octree sizing method, a good transition was achieved between the box and the outer flow field cells, up to the far – field walls in the volume mesh.

The region of the flow domain close to the vehicle surface, within the boundary layer region, was captured using polyhedral prismatic layers.

In Fluent Meshing, "Last Ratio" method was utilized, which can control the aspect ratio of the cells of the prism extruded out from the base boundary zone. Local base mesh size is used after specifying the first prism layer height to calculate the offset height of the last layer. This is illustrated in Figure 3.8. A local rate of growth was used to calculate the other intermediate offset heights exponentially.

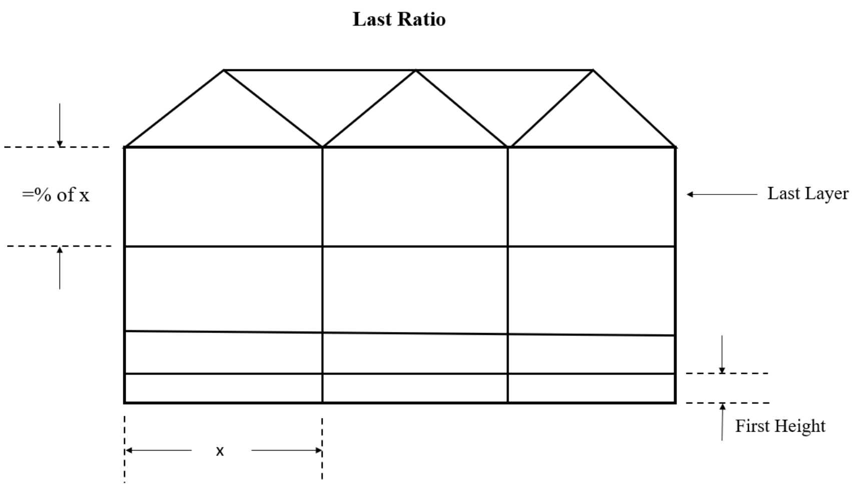


Figure 18: Last Ratio Prismatic Layers

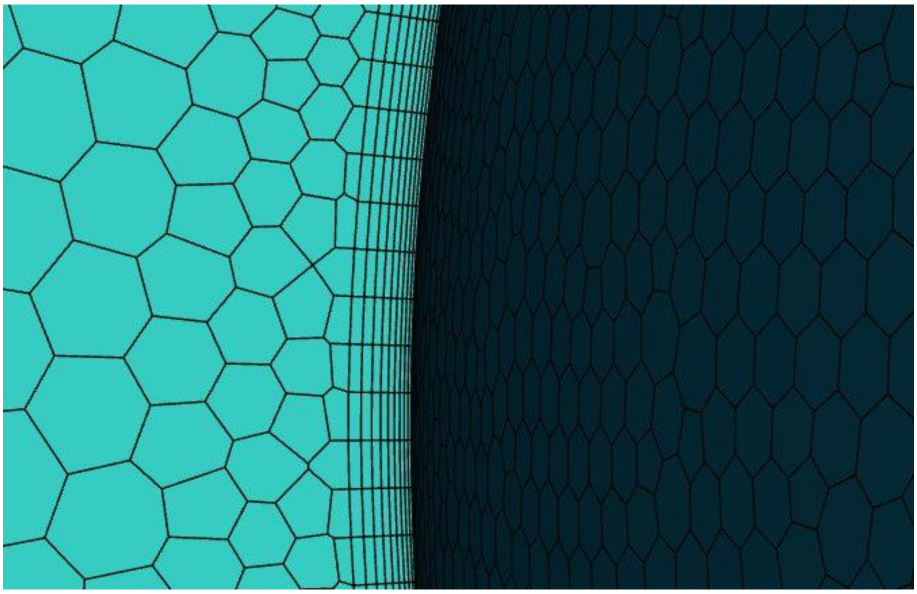


Figure 19: Layers on the HB-2

3.2.3 Solver Setup

The solver setup that was used during the HB-2 workflow, which was also applied to the final geometric variations, can be described in the following table.

Solver	Fluent
Version	2025R1
Type	Pressure - Based
Viscous Model	K – omega SST
Pressure – Velocity Coupling	Coupled
Spatial Discretization	
Pressure	Second Order
Momentum	Second Order Upwind
Turbulent Kinetic Energy	Second Order Upwind
Specific Dissipation Rate	Second Order Upwind

Table 1: HB-2 Solver Setup

3.2.4 Mesh Independence Study

A mesh independence study was conducted, to guarantee a solution where the end result would not be a function of the refinement level of the mesh. Three meshes were created with the refinement level ranging from coarse to fine. A factor of 2 was employed to refine between each mesh.

A wall resolved mesh was generated for each of the three meshes. This means that the first prismatic layer is located inside the viscous sublayer of the flow boundary layer.

For external aerodynamic flows, it is a common practice to have at least 10 cell layers inside the boundary layer, and for highly accurate simulations even up to 40 layers.

When the first layer is in log-layer, $y^+>30$, additional wall functions have to be used to capture the near wall behavior. When the first layer is in the viscous sublayer, no wall functions are needed. K-ω based models are y^+-insensitive, which enhances the method that was followed in this thesis.

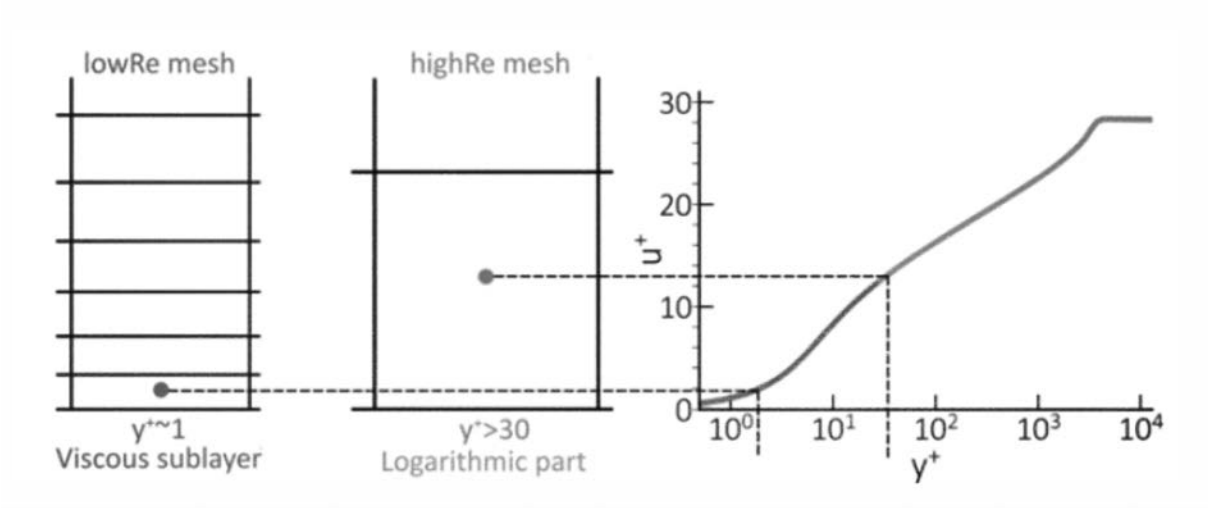


Figure 20: Near wall Profiles

The experimental model of the HB-2 generic missile, from which all the reference data were taken is depicted in Figure 3.11.

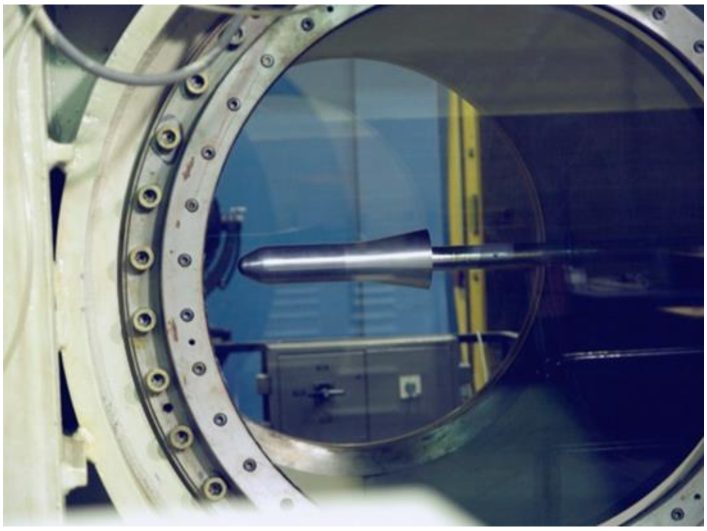


Figure 21: Experimental HB-2 Generic Missile

The available data were concerning supersonic analysis of the modes, so the operating conditions presented in the following table and used for the mesh independence study are for a Mach Number of 1.5.

Based on Experimental Data	
Model Diameter [mm]	100
AoA [deg]	0
Mach Number	1.5
Altitude [m]	10000
Reynolds	6182841
Operating Pressure [Mpa]	26500
Speed of Sound [m/s]	299.46
True Airspeed [m/s]	449.19
Reference Area [m^2]	0.02

Table 2: HB-2 Experimental Data

A pressure far-field boundary condition was applied to the domain walls, as it is standard practice in CFD for open flow domains where disturbances from the object decay with distance. It enforces freestream conditions at the boundaries, allowing waves to exit without reflection. In incompressible flow, it specifies pressure while allowing velocity to adjust, whereas in compressible flow it specifies pressure and Mach number, minimizing artificial effects on the solution near the object.

The ideal gas law was selected for the air density, since the flow involves a gas at conditions where ideal gas behavior is a reasonable approximation, simplifying calculations while accurately capturing pressure–density–temperature relationships.

The coupled pressure-velocity coupling method was used to accelerate convergence in a small number of iterations. A CFL (Courant–Friedrichs–Lewy) number of 20 was selected based on preliminary testing, optimizing the trade-off between stability and convergence rate.

The FMG (Full Multigrid) initialization method was applied to further enhance convergence. The solution was first computed on a coarse grid and then interpolated to progressively finer grids, providing a close initial guess that reduces iterations on the fine mesh and improves solver stability.

Some surface and volume mesh details for each refinement level are presented in the tables below:

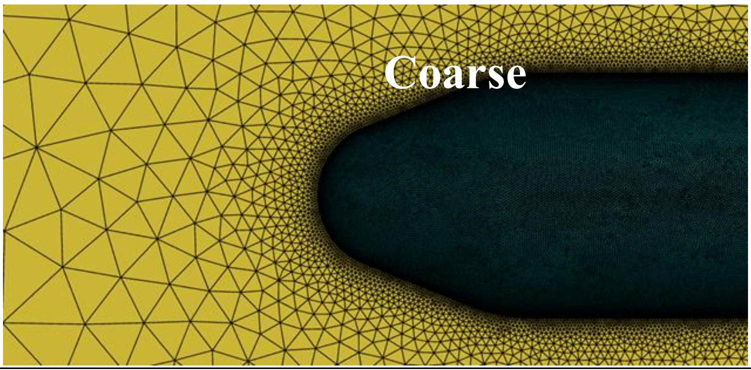
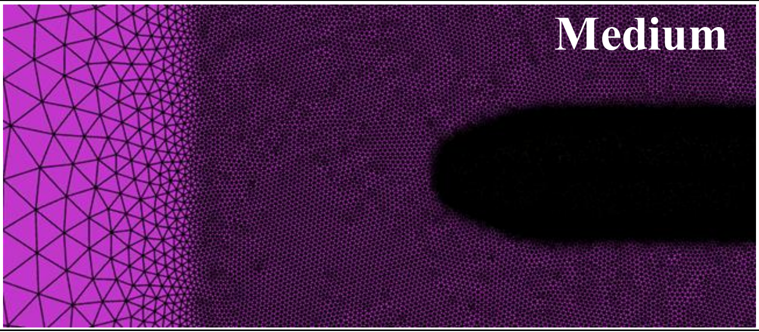
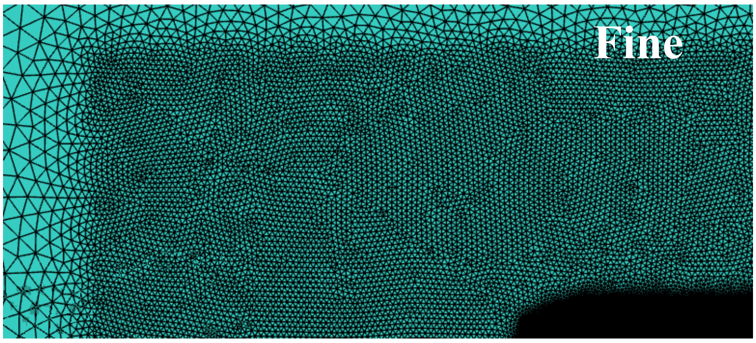
<table>
<tr><th>Surface Mesh</th></tr>
<tr><td>
</td></tr>
<tr><td>
</td></tr>
<tr><td>
</td></tr>
</table>

Table 3: HB-2 Refinements Surface Mesh

The coarse mesh is used as a baseline mesh, without bodies of influence used. The medium mesh then proceeds with a factor 2 reduction in the cell size and the addition of a refinement box with the same reduced cell size applied. Lastly, the fine mesh completes the refinement triplet, with an extra factor 2 reduction to the cell size, to both the surface mesh and the box that embodies the vehicle.

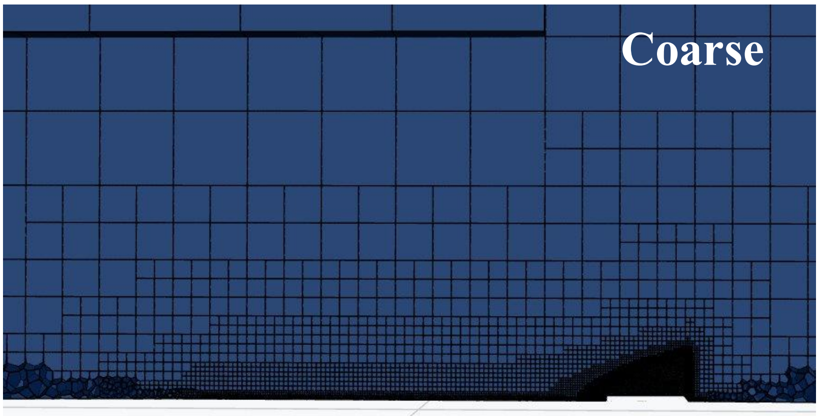
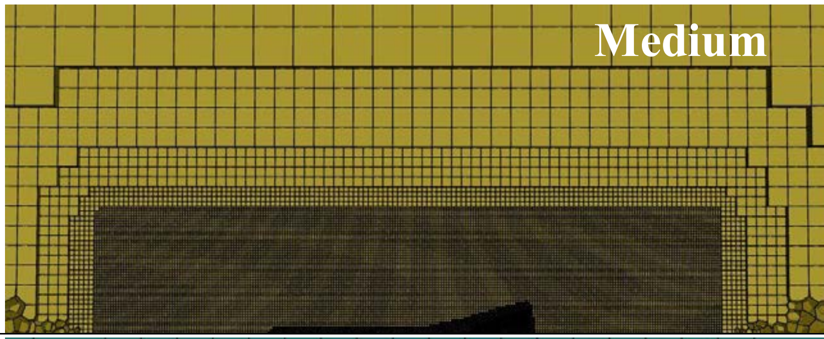
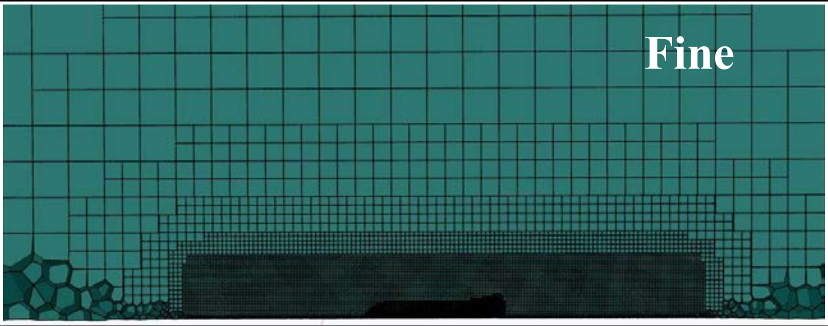
Volume Mesh




Table 4: HB-2 Refinements Volume Mesh

The cell count of the meshes as well as quality metrics and solution details are shown at the table that follows.

Mesh Independence Study			
	Coarse	**Medium**	**Fine**
Cell Count	2607925	5508800	10579209
BOIs	0	1	1
Min Cell Size [mm]	2	1	0.5
Aspect Ratio	529.7	441.9	471.3
Min Orthogonal Quality	0.07	0.05	0.04
Initialization Method	FMG	FMG	FMG
Convergence Time [min]	37	42	69
Iterations	268	331	350

Table 5: HB-2 Mesh Independence Metrics

After the solution was completed, some contours were extracted, to help us visualize the effect of the refinement levels in the shock waves and the general flow – field. In both the Mach number contours and the schlieren visualization contours, it is clear that as the refinement level increases from coarse to fine, the more defined the lines of the shocks are.

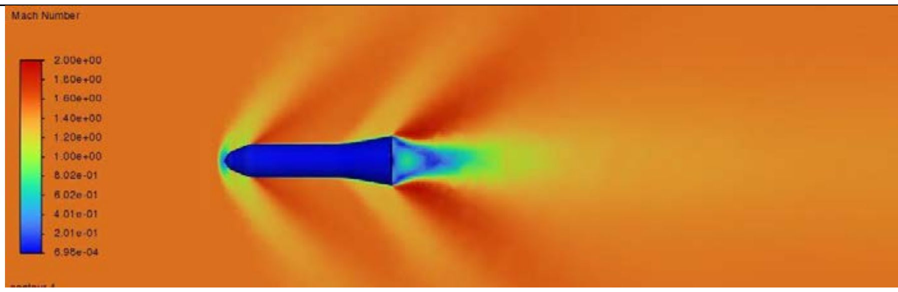
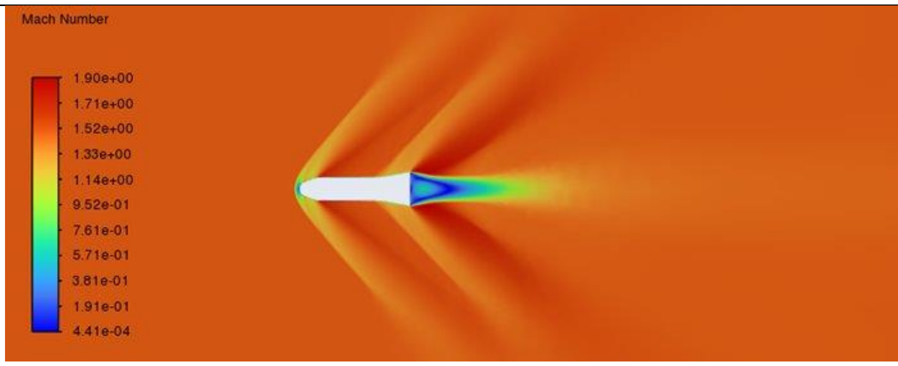
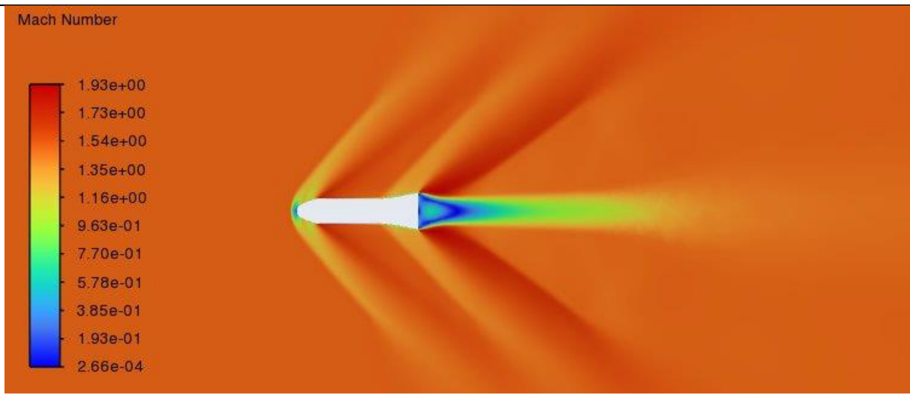
	Mach Number Contours
Coarse	
Medium	
Fine	

Table 6: Mach Number Contours

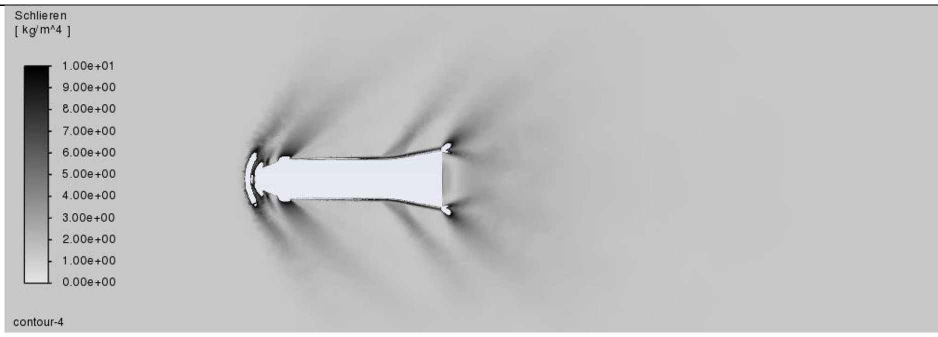
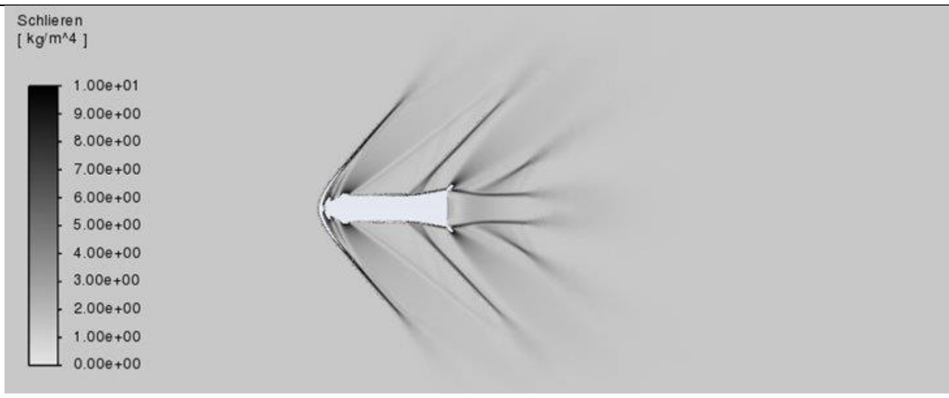
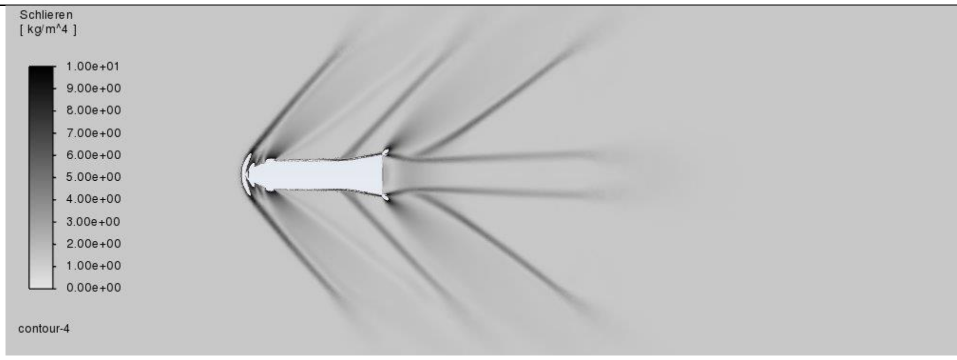
<table>
<tr><th></th><th>Schlieren Number Contours</th></tr>
<tr><td>Coarse</td><td>
</td></tr>
<tr><td>Medium</td><td>
</td></tr>
<tr><td>Fine</td><td>
</td></tr>
</table>

Table 7: Schlieren Number Contours

Mesh Independence Study 0 Deg AoA Results				
	Coarse	**Medium**	**Fine**	**Experimental**
Drag Coefficient	1.39607	1.46358	1.45589	1.45
Lift Coefficient	-0.00047	-0.00049	-0.00049	0
Pitching Moment Coefficient	-9.38E-07	1.17E-07	-5.67E-07	0
Workflow Selection → Medium				

Table 8: HB-2 Mesh Independence Results

The results from the mesh independence study, at zero-degree angle of attack, indicate that the level of refinement for the medium grid is the best compromise of the three grids. The coefficients of drag, lift, and pitching moment, relative to the fine level of refinement, have minimal differences, with the values very close to the experimental values from the classic HB-2 wind tunnel test database. This consistency justifies the application of the medium grid as the optimal choice for the present workflow.

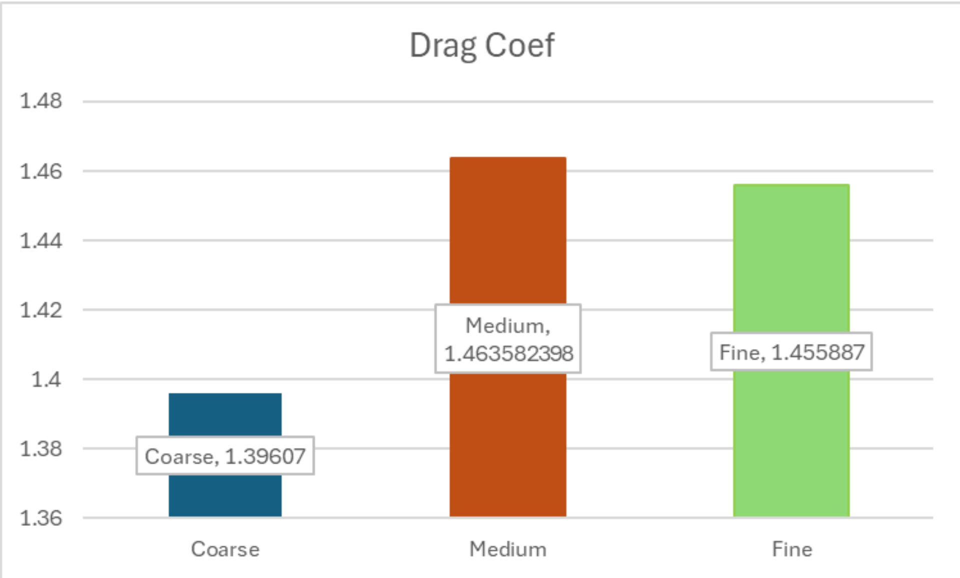


Figure 22: Mesh Independence Study – Drag Coefficient

After selecting the medium mesh from the mesh independence study, the flow conditions were updated to reflect the actual test case. The Mach number was reduced from 1.5 to 0.8 to match the conditions of interest, and the target y+ value was adjusted to ensure appropriate near-wall resolution for turbulence modeling at this regime. In addition, all solver and setup parameters related to compressibility effects were updated accordingly. The validated workflow was then applied to the geometrical variations of the custom maneuverable vehicle that will be used for the SimAI model creation.

A spherical domain was deployed for the application on the final model, since it was considered that it encloses the variating parameters in a better way than the conical for the nose – changing geometries.

The average mesh size of the ten initial variations was 5.6 million cells, and the average solution time was 94 minutes. The domain and details of the generated volume mesh are depicted below.

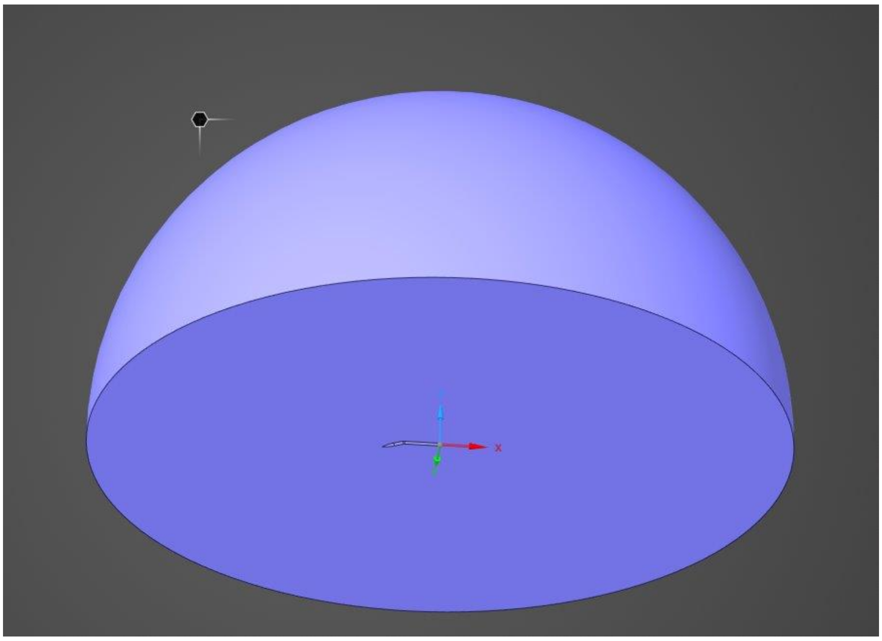


Figure 23: Maneuverable Vehicle Domain

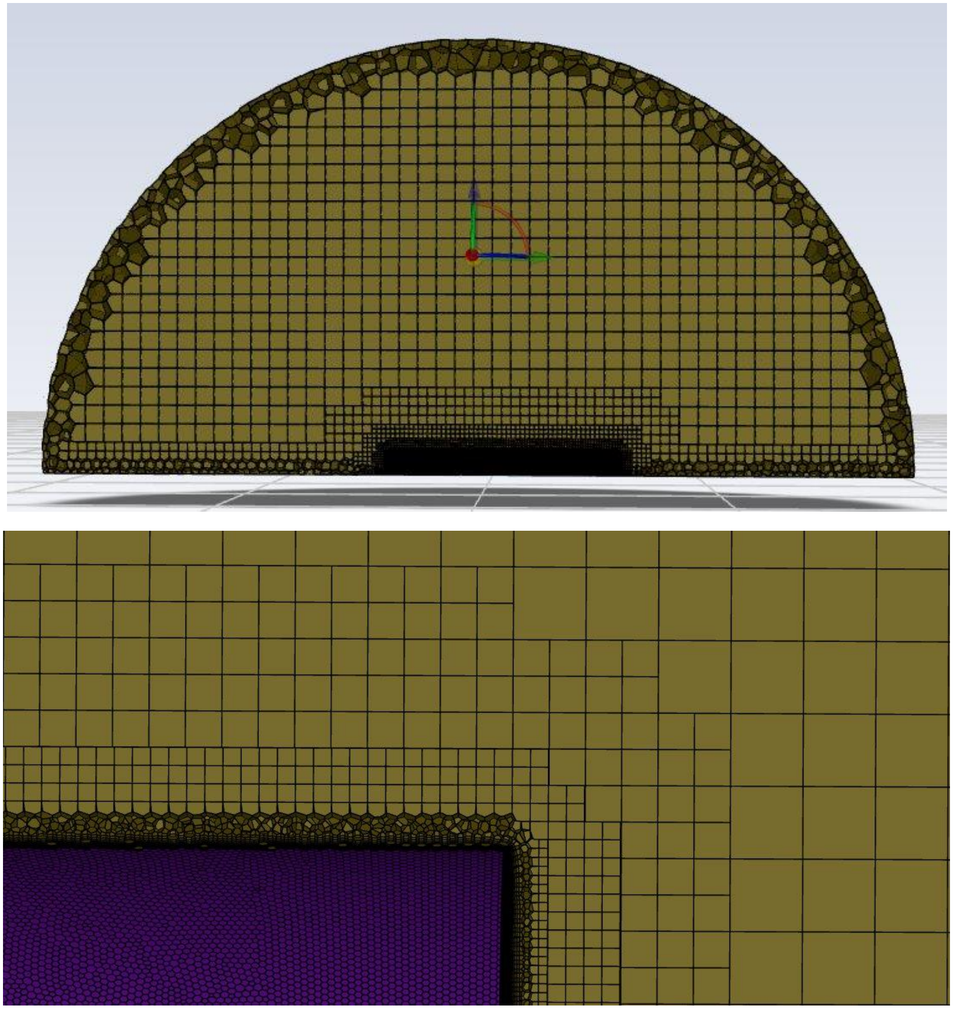


Figure 24: Maneuverable Vehicle Mesh Details

3.3 AGARD Models Workflow

3.3.1 Geometry Description

The selected geometry on which the whole AGARD Models Workflow was based on, is the AGARD – B generic missile in a half model approximation. A spherical domain was used to surround the geometry, with a Farfield boundary condition applied on the surrounding wall.

3.3.2 Mesh Characteristics

Since this is also a high speed, external aerodynamics study in the same Mach regime as the HB-2 study, the same workflow logic was applied here as well. A watertight mesh with bodies of influence used for extra refinement were selected. Local sizing was added at the high proximity regions of the wings, to help us capture the small facets that part the curvature of them. A poly – hexcore volume mesh was generated, with last – ratio prism layers capturing the thickness of the boundary layer.

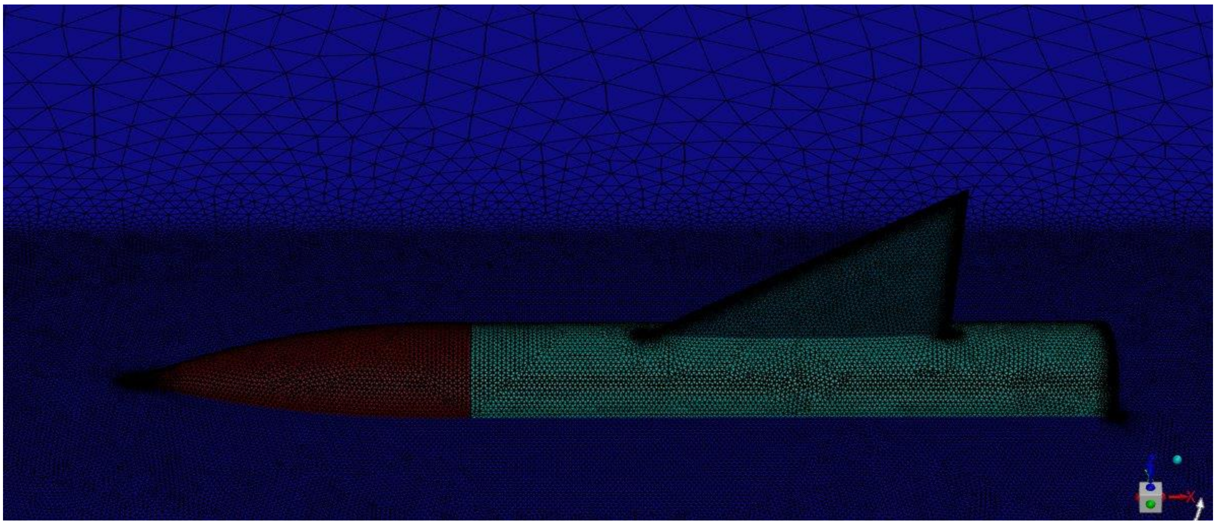


Figure 25: AGARD B Mesh Details

3.3.3 Mesh Independence Study Results

A separate mesh independence study was necessary for this case as well. There is extended bibliography and experimental data explaining the wind tunnel tests conducted on the AGARD – B and AGARD – C models, so it was easy to extrapolate the operating conditions and the reference data for the CFD tests of this study too.

The operating conditions are the following:

Operating Conditions	
Mach	0.8
Air Velocity [m/s]	267.99
Dynamic Viscosity [Pa*s]	0.00001745
Reynolds	7419929
Temperature [K]	279.225
Operating Pressure [Pa]	39361.297
Density [kg/m^3]	0.491

Table 9: AGARD B Operating Conditions

Three refinement levels were selected for the refinement study, as before. The first layer height and the total number of boundary layers vary for each case. It is obvious that as the refinement level increases from coarse to fine and the Y+ value is decreasing, the number of layers is increasing, aiming for all of them to have the same boundary layer thickness.

This increment in layers is visible in the pictures of Figure…

	Coarse	Medium	Fine
Min Cell Size [mm]	0.22	0.11	0.055
BOI Cell Size [mm]	7.04	3.52	1.76
First Layer Height [mm]	0.0036 **[Y+ 1]**	0.0025 **[Y+ 0.7]**	0.0015 **[Y+ 0.4]**
Number of Layers [mm]	20	24	28
Cell Count	1854806	4836610	17353204

Table 10: AGARD B Mesh Independence

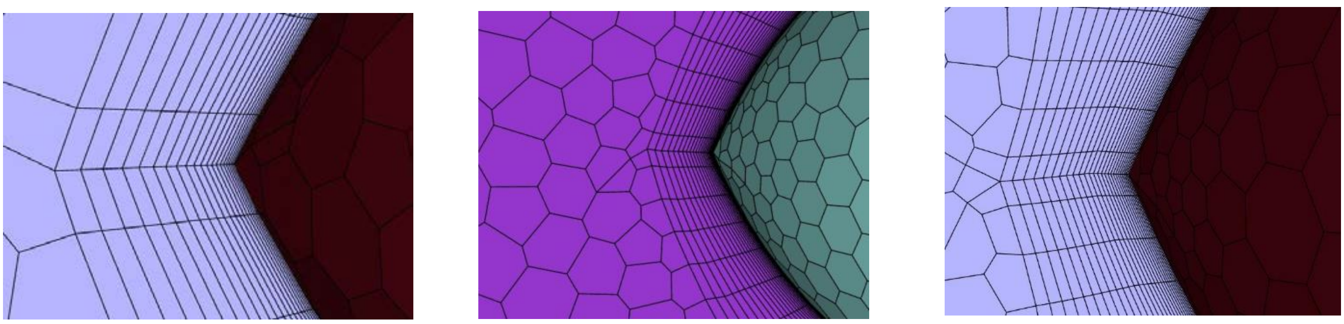


Figure 26: Refinement Level Layers from Coarse to Fine (Left to Right)

The same solver setup, as in the HB-2, was applied for this case, with the boundary and operating conditions changing to match the naming of the surfaces and the new operating velocity values. All the other settings, including the FMG initialization method, remained the same.

The solution data indicated that the medium refinement level represents the most suitable compromise for this case. The moderate increase in error compared to the fine mesh is outweighed by the significant reduction in computational cost, as the medium grid requires only about one-third of the cells of the fine refinement level.

0 AoA AGARD-B Model				
	Coarse	**Medium**	**Fine**	**Exp. Data**
Drag Coef.	0.005128 **[17.3%]**	0.005681 **[8.4%]**	0.005877 **[5%]**	0.0062
Lift Coef.	-3.48E-06	-2.23E-06	7.49E-07	0
Moment Coef.	-5.26E-07	-8.95E-08	1.08E-07	0
Solution Time [min]	75	169	512	
Selection → Medium				

Table 11: AGARD B Mesh Independence Results

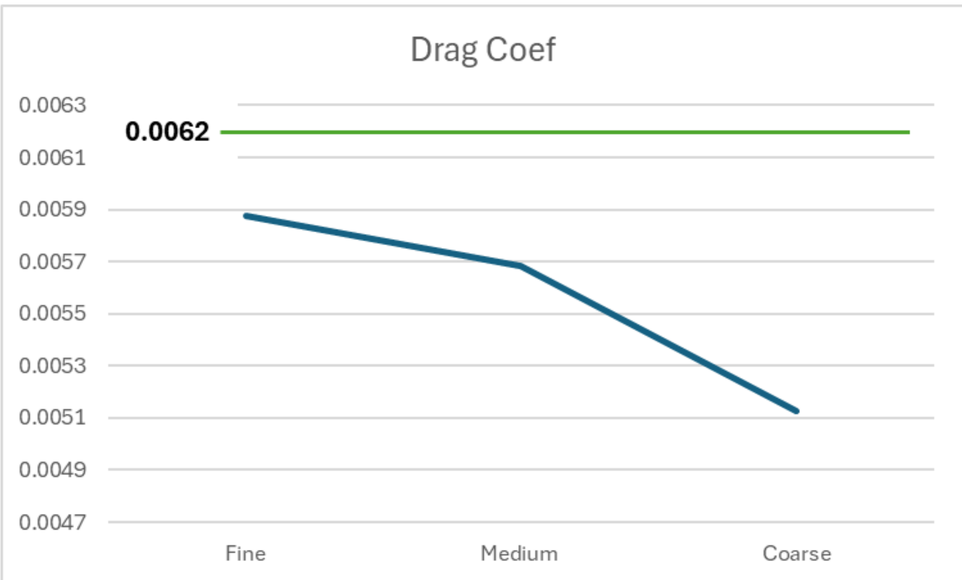


Figure 27: AGARD B Mesh Independence – Drag Coefficient

This workflow was then applied to the five intermediate geometries that I designed, which will be used in the next chapter for the generation of the second and most detailed SimAI case.

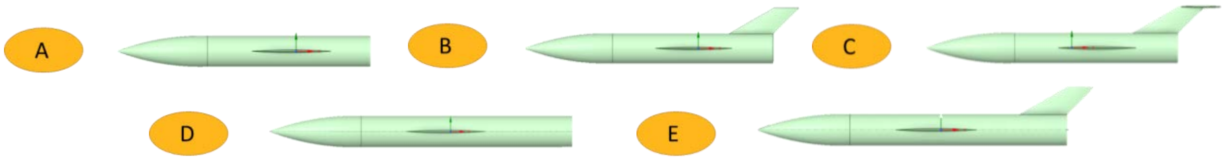


Figure 28: Custom Intermediate Geometries

4. SimAI, an AI Demonstrator

The main purpose of this research is to create high fidelity AI predicting models for aircraft aerodynamics. Such applications require accurate assessment of the vehicle performance to predict maneuverability, define mission controls and endure safety. It is known that experimental analysis is extremely costly and while CFD simulations can accelerate the design process and reduce costs, it can be computationally expensive. What is studied in this thesis and more specifically in this chapter, is how to create some AI models that can be trained using historical or new high fidelity CFD data, to perform 3D fluid flow predictions in seconds. In this way, the total procedure is speeding up to 480 times compared to CFD simulations, for the studied cases, while maintaining low errors in the predictions.

4.1 SimAI Platform

ANSYS SimAI is an advanced simulation-driven artificial intelligence tool that intends to revolutionize the process of engineering by seamlessly connecting the conventional physics-based simulation software with AI strengths. With the combination of ANSYS simulation software's robust, physics-accurate model strength with cutting-edge machine learning and deep learning algorithms, SimAI allows engineers to accelerate product development, support design optimization and enhance decision-making at speeds and precision unlike ever before. This system employs data sets generated from large-scale simulations to train AI models that are able to predict complex physical behavior, detect patterns, and provide novel solution suggestions without needing to perform extensive, repetitive simulations. SimAI is particularly beneficial in fields such as aerospace, automotive, energy, and electronics, where complex multiphysics phenomena need to be investigated and optimized with efficiency. Its easy-to-use interface makes it easy for engineers to incorporate AI into existing simulation workflows, enabling the development of surrogate models, automation of design explorations, and real-time analysis of performance under varying conditions. In addition, ANSYS SimAI facilitates cross-disciplinary, collaborative engineering by offering data-driven insights and reducing reliance on expensive physical prototypes. Overall, SimAI represents a paradigm shift in computational engineering, combining the strengths of simulation and AI to drive innovation more quickly, reduce costs, and deliver faster time-to-market for sophisticated engineered products.

4.2 Machine Learning Fundamentals

Machine learning (ML) is a branch of artificial intelligence that deals with developing algorithms that can discover patterns and make data-driven predictions or choices without relying on pre-programmed rules. The essence of ML lies in training models against historical or simulated data so they can generalize and make inferences on new, unseen examples. When applying ML techniques, the aim is to enable an algorithm to learn to carry out a task or general rule, presented with a set of examples. There are various approaches to doing this when integrating ML with numerical simulation.

Here are two categories of such methods:

1. **Physics-aware:** The simplest approach is to replace the bottleneck of conventional solvers with ML algorithms that are able to make quicker predictions, i.e., accelerate the inversion of a large linear system of equations. They do, however, exclude the solver that is involved in the calculation and preserve data about the governing equations, e.g., by incorporating penalization terms recommended by the equations into the loss function, i.e., the amount of error in your training data. That is, physics-aware methods aim at a particular physical task and succeed in achieving this, but they might do so at the expense of accuracy.
2. **Physics-agnostic:** The other option is for the ML algorithm to learn the latent physics representation from the numerical solver solution directly. The approaches are also solver-agnostic and equation-agnostic. For example, using instances of previous computations, ML algorithms can learn a data-driven definition of the Navier–Stokes function and utilize a new geometry and freestream condition as inputs like the speed of a car to deliver its flow field as output. Therefore, physics-agnostic methods introduce speed and prediction correctness.

SimAI is a physics-agnostic platform, accelerating your simulation pipeline without compromising on its predictive accuracy. Instead of providing geometric parameters to define a design, SimAI software takes the shape of a design as the input. This simplifies it to investigate more designs even when the structure of the shape is not uniform across the training data. Essentially, the SimAI platform is engaged in three simple steps: upload data, train AI model, and predict. As mentioned, customers can train the AI model using previously generated Ansys or non-Ansys data.

$$S(\underline{x}) = N(\underline{x}, BC) \tag{4.1}$$

4.3 Architecture

The generic architecture of the SimAI platform is based on a fusion of different techniques that combine multiple deep learning neural networks. This type of architecture makes it possible to capture all the important scales of physics. The architecture is composed of a significant number of nonlinear layers, including multiple parameters and complex interactions between variables.

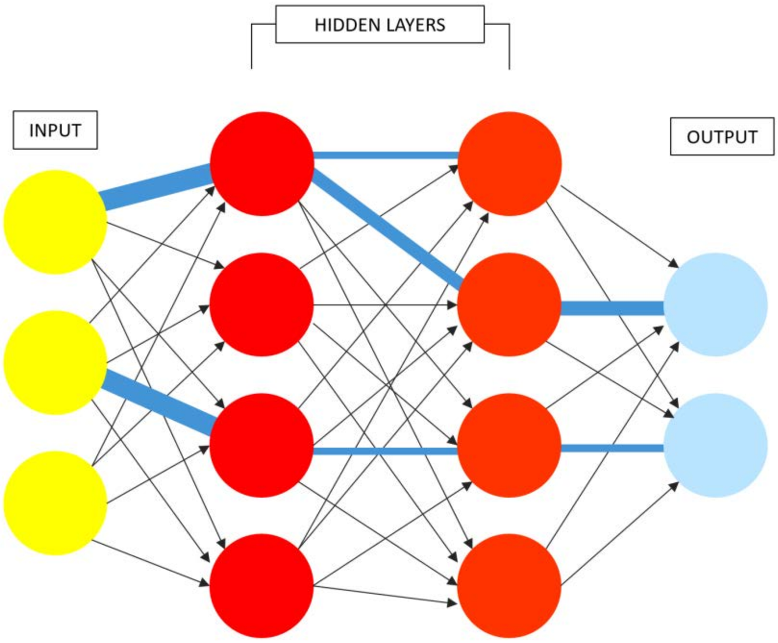


Figure 29: SimAI Generic Architecture

Instead of explicitly storing data points like pixel values in an image, the SimAI platfrom uses an implicit neural representation to learn a continuous function that can generate these data points. This means that SimAI software can take a set of points obtained from previous computations and generalize well to new geometries and freestream conditions.

This capability enables a continuous representation of the surface and volume fields, e.g., pressure and velocity. This makes it possible to request a prediction at a desired resolution. Additionally, global coefficients can be derived from the predicted fields as post-processing.

In fact, one of the SimAI platfrom's strongest assets is its use of regularization techniques to prevent overfitting and improve generalization for new geometries. What does this mean? Basically, overfitting is when an AI model's predictions are too narrowly focused on its training data and included geometry, lacking any generalization or scope to learn or generalize new geometries. Regularization techniques are designed to reduce overfitting. SimAI uses regularization techniques, including local methods that are embedded directly in the structure of the model and lead to a more stable and expressive model. It's the reason SimAI software can work with new geometries so quickly and easily.

Similarly, the SimAI platform uses an adequate representation of 3D shapes to describe arbitrary or irregular meshes with complex geometrical variations that do not follow a specific distribution, e.g., unparameterized geometries. To help quantify the uncertainty in the prediction, SimAI software uses a unique confidence score to calculate the distance from the nearest known geometry in a very high-dimensional space.

4.4 Generation of Training Data

In this section, the procedure of generating the training datasets required for the model buildup in the SimAI platform will be analyzed thoroughly, from the beginning of the concept behind the usage of this tool, up to the details of the model setup.

4.4.1 Benchmarking Quantities

After completing the CFD simulations for every one of the data points that will be used for the generation of the AI models in the platform, it is important to decide the quantities that we want to benchmark results on, between SimAI and CFD. This is a crucial step that needs to be taken first, because it will affect the following steps of the procedure, as it will be analyzed in the next sections. After deciding that we want to benchmark the predictions of e.g. lift force on SimAI in comparison with CFD results, we have to extract the quantities that part its formula from the solver in the appropriate format, convert the extracted files into another specific file version and then upload the new files to the platform, where we will use them for our purpose. So, it is obvious that we must be thoughtful about the research that we want to focus on from the beginning. For the first case of the parametric, maneuverable vehicle model, the forces of Drag and Lift were selected for the tests, while for the non – parametric, AGARD case, the drag, lift and moment coefficients were chosen for the same purpose.

4.4.2 Data Extraction and Conversion

As mentioned before, the forces that will be used for the evaluation of the SimAI models will be Drag, Lift and Moment in total. They can be converted to the respective coefficients in the data analysis stage for the second case. The formulas of these forces in integral form can be described as follows, defined on the axes used in both the cases of this study:

- Drag Force in the X - axis:

$$Drag[X] = \int_S (pressure \cdot Normals_X) dS + \int_S (x\_wall\_shear) dS \quad (4.2)$$

- Lift Force on the Y - axis:

$$Lift[Y] = \int_S (pressure \cdot Normals_Y) dS + \int_S (y\_wall\_shear) dS \quad (4.3)$$

- Moment on the Z – axis:

$$\begin{aligned} Moment[Z] = \int_S ((Centroids_X - 0) \cdot \left(pressure \cdot Normals_X + x_{wall_{shear}}\right) \\ - (Centroids_Y - 0) \\ \cdot (pressure \cdot Normals_Y + y\_wall\_shear)) dS \end{aligned} \quad (4.4)$$

After reviewing the equations, it is obvious that the quantities we need to extract for each of the data points are:

- X – Y – Z Velocity
- X – Y – Z Wall Shear Stress
- Static Pressure

The format in which the above quantities are exported from Fluent is Ensight Case Gold, which is the correct format to carry the information needed, but it is not compatible with the platform. SimAI accepts files in the .vtk file form, which is associated with the Visualization Toolkit (VTK), which is an open-source software system widely used for 3D computer graphics, image processing, and scientific visualization. To convert the exported Ensight Case Gold files to this format, a custom made PyVista script was used, and the desired output was one .vtp file, containing the surface data for the selected surfaces that were provided as inputs and a .vtu file for the volumetric data of the fluid domain.

4.4.3 Model Buildup

This subsection is focused on describing the development and configuration of the machine learning model within the SimAI platform, focusing on the input–output variable definition, boundary constraints, tuning of global coefficients, and evaluation of model confidence.

4.4.3.1 Input – Output Variables

The model inputs are based on a combination of geometry-derived features and aerodynamic boundary conditions. Geometry is extracted directly from surface data and the model does not take direct surface values as input in both studied cases.
The output variables are defined over both the volume and surface of the flow field. For the volume, the model is trained using static pressure and the three velocity components: x - velocity, y - velocity, and z – velocity, which are extracted from the simulation data. On the surface, the output features include pressure, wall shear in both the y and z directions, as well as x – wall shear. These quantities are obtained from the training dataset and used for inference during deployment. Collectively, these outputs are essential for assessing the aerodynamic characteristics of the configuration and for capturing detailed near-wall flow behavior, which is critical in aerodynamic performance evaluation.

4.4.3.2 Boundary Conditions

In SimAI, boundary conditions (BCs), even though not mandatory, can be a critical part of how the AI model interprets and predicts fluid behavior. Even though SimAI is a data-driven model, it still respects the structure and constraints of the CFD problems it is trained on. The AI model learns from CFD simulations where boundary conditions (like inlet velocity, pressure, wall conditions) are already defined in the solver setup. So, the BCs inside SimAI serve more like operating conditions, or constraining parameters that guide the predictions.
Finally, it is important to understand that SimAI doesn't enforce BCs analytically the way a traditional solver does. So, while it's fast and predictive, boundary-aware accuracy is only as good as the diversity and quality of training data.

4.4.3.3 Global Coefficients

Global aerodynamic coefficients are calculated and monitored as part of model configuration. They are defined prior to the model buildup and are derived from the predicted flow fields. They can be forces such as lift, drag, moment, or any other quantity that can be structured based on the extracted quantities from fluent during the data extraction stage. They are a very useful tool and can be extracted directly from the platform to guide optimization during post processing and data analysis.

4.4.3.4 Confidence Score

To ensure reliability at the early stages of design, a confidence score for each prediction is obtained. The score is derived from the model uncertainty estimates internally, either softmax entropy or Monte Carlo dropout-based variance, based on the architecture chosen. The confidence score provides a means to verify whether a specific prediction lies within the training region of the model. On SimAI, this allows the system to notify low-confidence predictions and either trigger fallback processes or request additional data, supporting secure and resilient design workflows.

4.5 Maneuverable Vehicle – Parametric Model

The first case that is studied in this thesis is regarding a custom high speed maneuverable vehicle. The key challenge is to speed up aerodynamic predictions of parametric geometrical changes through the power of AI. By creating this case and following up the forementioned research for the simulation case setup, it is time to move with the usability of SimAI for this case. More specifically, starting with the general model buildup, we are going to analyze the behavior of SimAI in predicting intermediate geometries, how the number of data points affect the accuracy of the model and finally the effect of BCs as a constraint in the platform.

4.5.1 Parametric Model Buildup

For the model buildup of this case 3 files were uploaded for each data point. One .vtp file containing the surface data, one .vtu file for the volumetric data and one boundary condition file, where the variable selected was the geometric nose angle of each data point variation.
The final goal is to create a map containing data of 46 predictions, from 0º to 45º, with 1º increments per variation.

4.5.2 10 Data – Points and Boundary Conditions

The first and most detailed model that was generated consists of 10 data points, starting from the 0° geometric nose angle variation, up to the 45° one, with 5° increments between them. SimAI follows a specific philosophy, where 90 percent of the data provided are used for the training of the model and the rest 10 percent is used to test the model after the training is complete. Following this logic, the 9 geometries indicated in orange in image 4.2 were used for the training and the 1 highlighted in red was used for the testing phase.

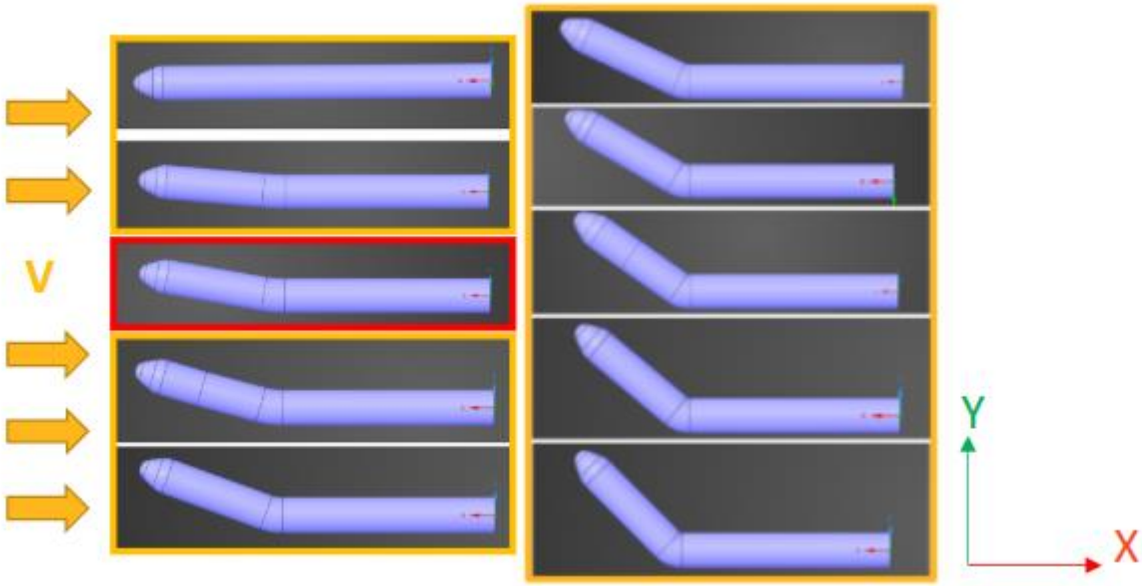


Figure 30: 10 Data Points (DP)

The model was trained two times, one with a training time of less than 24 hours and one with a training time of less than 7 days. These options are provided by the platform for the user to decide the depth of the accuracy that he wants in his training.

Two sets of 46 predictions were performed, one for each of the time - variant trained models. The following graphs represent the Drag [X] and Lift [Y] forces in the y – axis, versus the Geometric Nose Angle in the x - axis, allowing us to map the behavior of the model in all the intermediate geometrical variations. The discriminated categories are as follows:

- Blue rectangles for the available CFD Data used for the training of the model, 9 points in total, as the number of data points provided for this model.
- Red crosses for the SimAI predictions on the <7 days trained model, 46 in total.
- Green circles represent the SimAI predictions on the model with <24 hours of training, 46 in number.
- Yellow triangles are used for the depiction of the CFD data of all the Unseen geometrical variations. Those CFD data were not used for the training of the model, so they were not seen by the platform at all. Therefore, they are selected for the evaluation of the model's performance and the calculation of the mean error between SimAI predictions and CFD results, allowing us to quantify the deviation between the two methods.

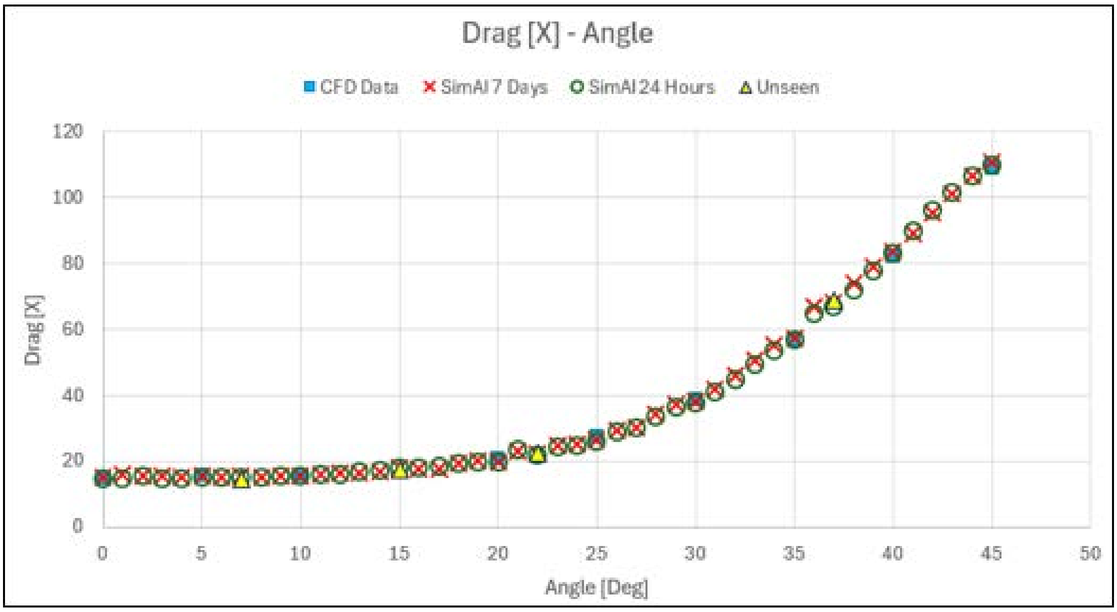


Figure 31: 10 DP – Boundary Conditions Drag vs AoA

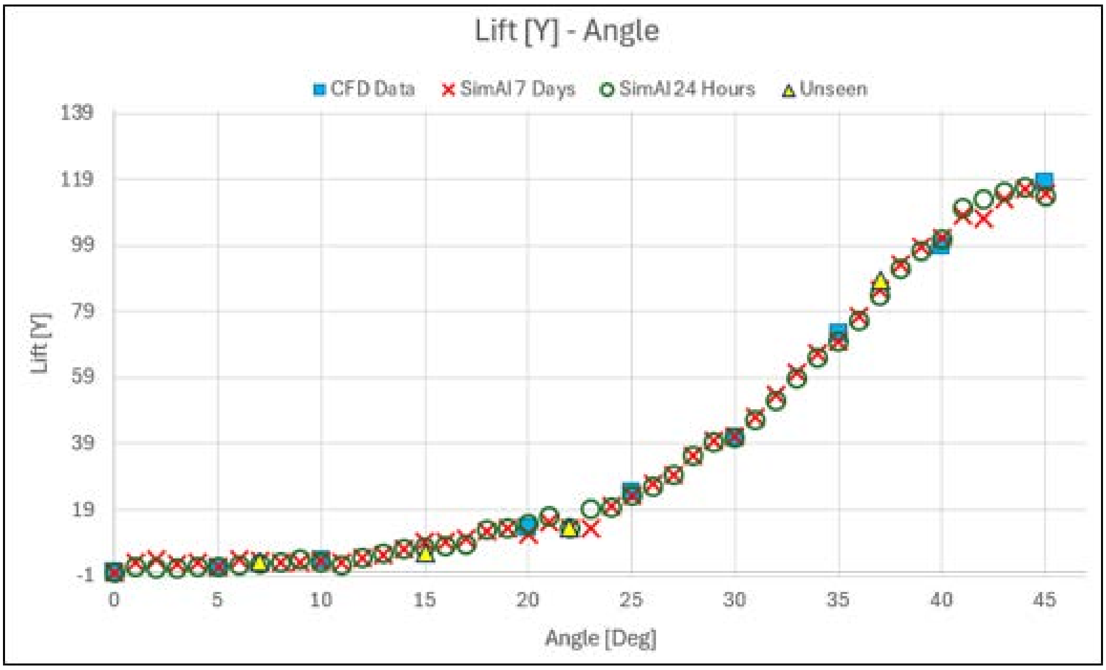


Figure 32: 10 DP – Boundary Conditions Lift vs AoA

The following table showcases the mean error percentage for drag and lift forces between SimAI predictions and CFD simulation results on the four selected unseen geometries. Three of those geometries, 7°, 22° and 37° were randomly selected and the fourth geometry, 15°, corresponds to the data point that was used for the testing of the model during the buildup.

Mean Error [%] SimAI Pred Vs CFD Data Unseen Geometries [7°, 15°, 22°, 37°]		
Training Time	< 24 Hours	< 7 Days
Drag [X]	1.98	2.09
Lift [Y]	16.3	14.6

Table 12: 10 DP - Boundary Conditions Results

The formula based on which the mean error percentage is calculated is presented below.
Mean Error Calculation SimAI to CFD:

$$ME = \frac{1}{4} * \sum_{i=1}^{4} 100 * \frac{\mathrm{CFD(i)} - \mathrm{SimAI(i)}}{\mathrm{CFD(i)}} \tag{4.5}$$

The big mean errors in lift come as a result of a big error in one of the predictions, as can be seen for example in the 37° prediction. When the total mean error is calculated, this error is distributed between the four variations that contribute to it and it leads to the errors indicated to the table.

4.5.3 10 Data – Points without Boundary Conditions

Another significant aspect of this test is the investigation of SimAI's boundary conditions effect on the accuracy of the model. This subsection analyzes the buildup and evaluation of the second model regarding this case. In this model, the initial 10 data points were also used for the training and testing of the model in the exact same way as in the previous model, in both the two prementioned training periods, but the geometric nose angle was removed from the boundary conditions during the model setup.
The results of the predictions in comparison to the CFD data are plotted in the following graphs, following the same logic as in the previous model.

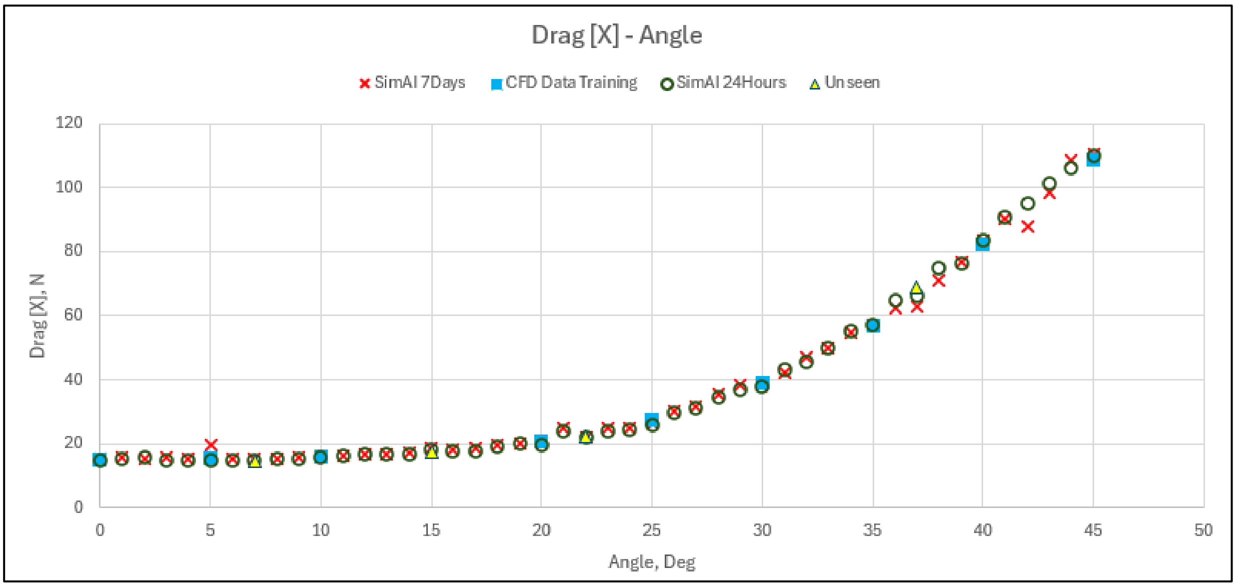


Figure 33: 10 DP – No Boundary Conditions Drag vs AoA

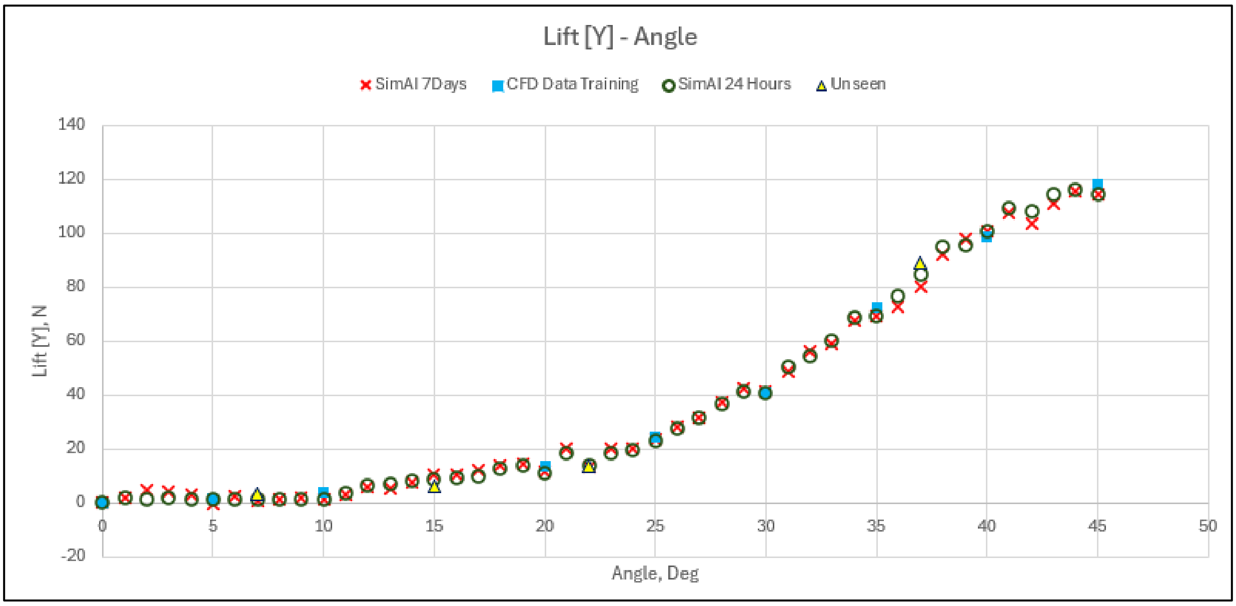


Figure 34: 10 DP – No Boundary Conditions Lift vs AoA

Once again, the numerical data are calculated for the four geometries that were unseen by the platform during the training of the model.

Mean Error [%] SimAI Pred Vs CFD Data Unseen Geometries [7°, 15°, 22°, 37°]		
Training Time	< 24 Hours	< 7 Days
Drag [X]	2.61	2.75
Lift [Y]	26.1	23.0

Table 13: 10 DP - No Boundary Conditions Results

4.5.4 Reduced Data – Points and Boundary Conditions

In fast paced industries, fast and accurate results are desired and every minute matters, especially during the preliminary design phase, where an initial bigger picture of the performance is enough to decide whether it is worth digging into more details or not. In this case, where less is more, a logical next step in the AI model research was investigating the effect of having less data points during the model buildup, in the final accuracy of the model.

This model was set up with 6 data points, for two different training times. The variations used for training were selected to have 10° increments between them, starting with 0° and followed by 10°, 20°, 30°, 40° and the 45° one, which is the highest. The geometry used for testing was the 22° variation. The plots that are follow demonstrate the data for the <7 days training time.

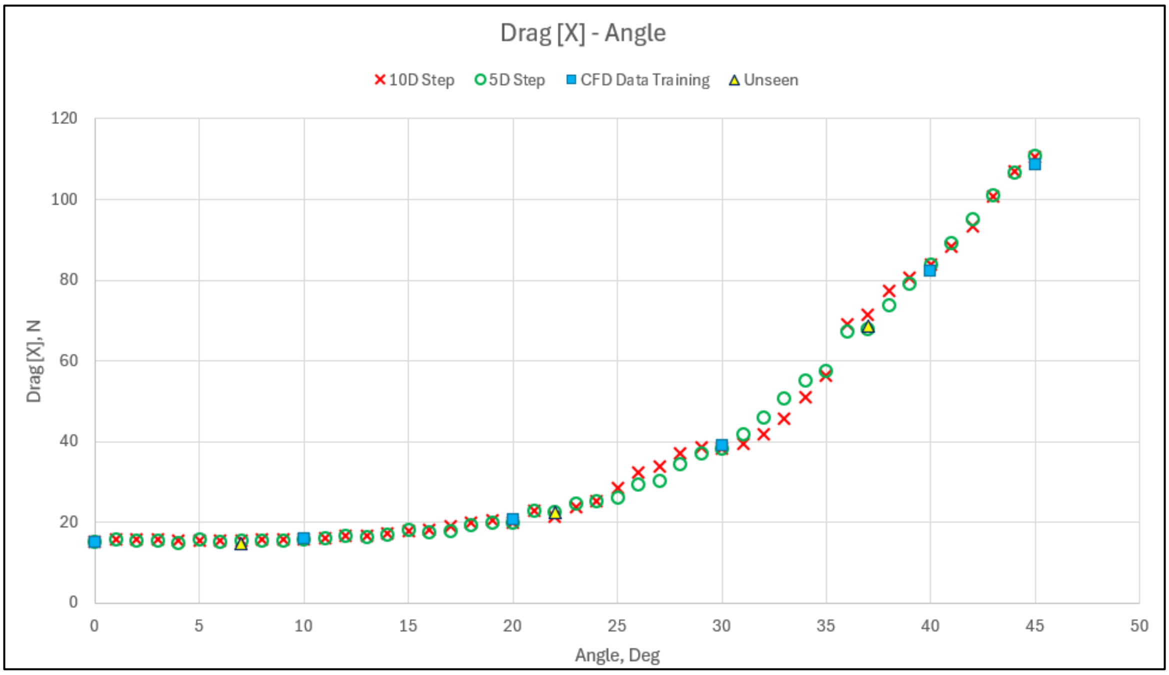


Figure 35: Reduced DP – Boundary Conditions Drag vs AoA

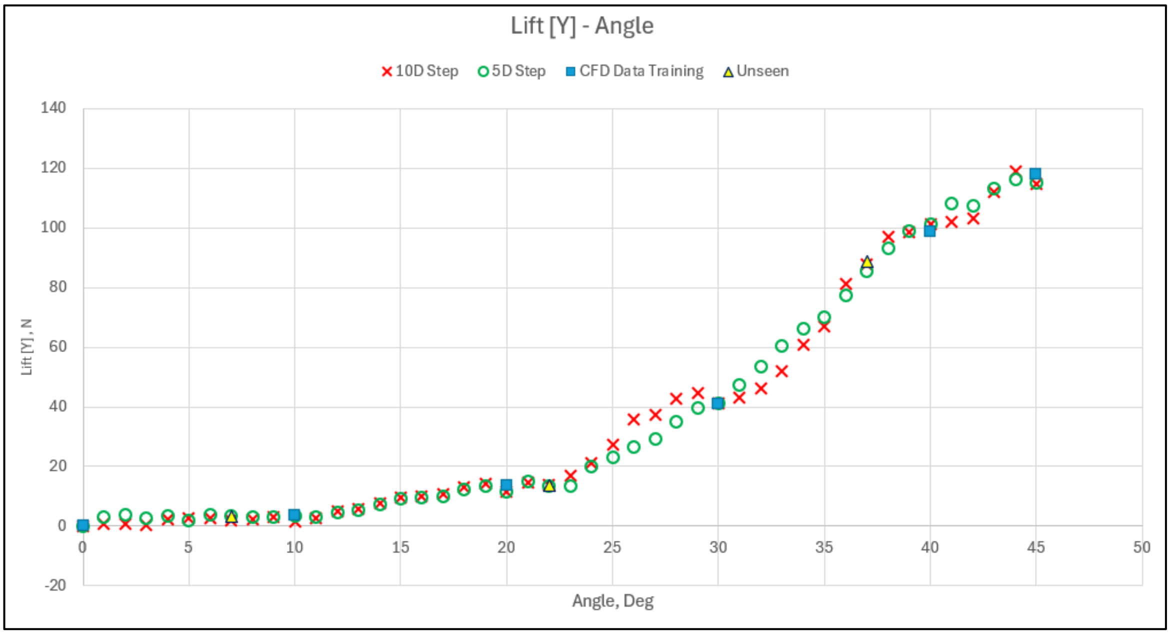


Figure 36: Reduced DP – Boundary Conditions Lift vs AoA

The numeric results that come from the graphs are gathered on the table below:

Mean Error [%] SimAI Pred Vs CFD Data Unseen Geometries [7°, 15°, 22°, 37°]		
Training Time	< 24 Hrs	< 7 Days
Training Data Points	6	6
Drag [X]	3.72	3.53
Lift [Y]	24.6	24.1

Table 14: Reduced DP - Boundary Conditions Results

4.5.5 Results Comparison

Mean Error % SimAI Predictions vs CFD Data → 4 Unseen Geometries						
	Less Sets of Data – BC		**10 Sets of Data – No BC**		**10 Sets of Data - BC**	
	24 Hours	**7 Days**	**24 Hours**	**7 Days**	**24 Hours**	**7 Days**
Drag [X]	3.72	3.53	2.61	2.75	1.98	2.09
Lift [Y]	24.6	24.1	26.1	23.0	16.3	14.6

The comparison of performance among the different SimAI model configurations reveals in a very systematic way how training data set size, boundary conditions (BCs), and training time affect accuracy of aerodynamic predictions relative to high-fidelity CFD data. In all experiments, the general trend is that drag predictions are always accurate, while lift predictions are data availability and constraint sensitive and have higher variance in error rates.

In the case of 10 data points with boundary conditions, SimAI worked best across the board. The mean drag error was extremely low, below ~2.1% in both training times. With lift, although the error was higher (14.6–16.3%), it is much lower than with any of the other test configurations run. This implies that the inclusion of boundary conditions, specifically the geometric angle of nose, provides the model with the critical physical restrictions required to be able to catch up better on the complex aerodynamic tendencies. In essence, the BCs are a stabilizing factor, promoting generalization towards novel geometries, particularly at middle angles where nonlinear aerodynamic impact is more pronounced.

Removing boundary conditions using the same 10 points created a clear decrease in accuracy, especially in lift prediction. Drag error increased moderately (to ~2.6–2.8%), which is still within tolerable preliminary design limits, but the lift error jumped to 23–26%. This means that without the additional constraint of geometric input, the model is worse at preserving fine aerodynamic distinctions, leading the model further away from CFD results. In physical terms, lift is far more sensitive to changes in geometry and flow conditions than drag, so boundary condition direction does not add potency to prediction error enhancement by the AI model.

The third case, with 6 data points and boundary conditions alone, is the low-data case. Performance degraded as expected: drag error rose to 3.5–3.7% and lift error remained high at 24–25%. This illustrates the model's sensitivity to sufficient training data to interpolate aerodynamic performance across unseen geometries. What is particularly interesting is that even though the decrease from 10 to 6 data points roughly doubled the drag error, the lift error remained at a similar magnitude to the "10 points without BC" case. It means that data sufficiency and the presence of BC can partly be compensating factors, and at least one of them must be present to keep errors in reasonable ranges.

Finally, regarding training time, comparing models trained from times less than 24 hours to models trained from times less than 7 days did not show significant gains with longer training time. In most cases, error reductions were less than 1 percentage point, suggesting that the limiting factor in model accuracy was not training time, but rather the availability of structured input data (dataset size and boundary conditions). This is evidence for the argument that past a certain point, more training time with longer training means gives diminishing returns unless paired with bigger and more body-aware input sets.

In short, results verify maximal accuracy with bigger sets of data points and with BCs well-specified and lack or reduction of BCs or dataset size in proportion affects prediction of lifts. Drag, being the more stationary aerodynamic parameter, is robust to configurations. From a design standpoint, this means a practical trade-off: lower datasets or idealized models can be acceptable in initial conceptual design, when qualitative trends suffice, but quantitative accuracy and guaranteed lift prediction in high-speed vehicle design call for boundary-conditioned data-dense models.

In the following pictures static pressure contours of each test case are displayed, all being scaled in the same climax.

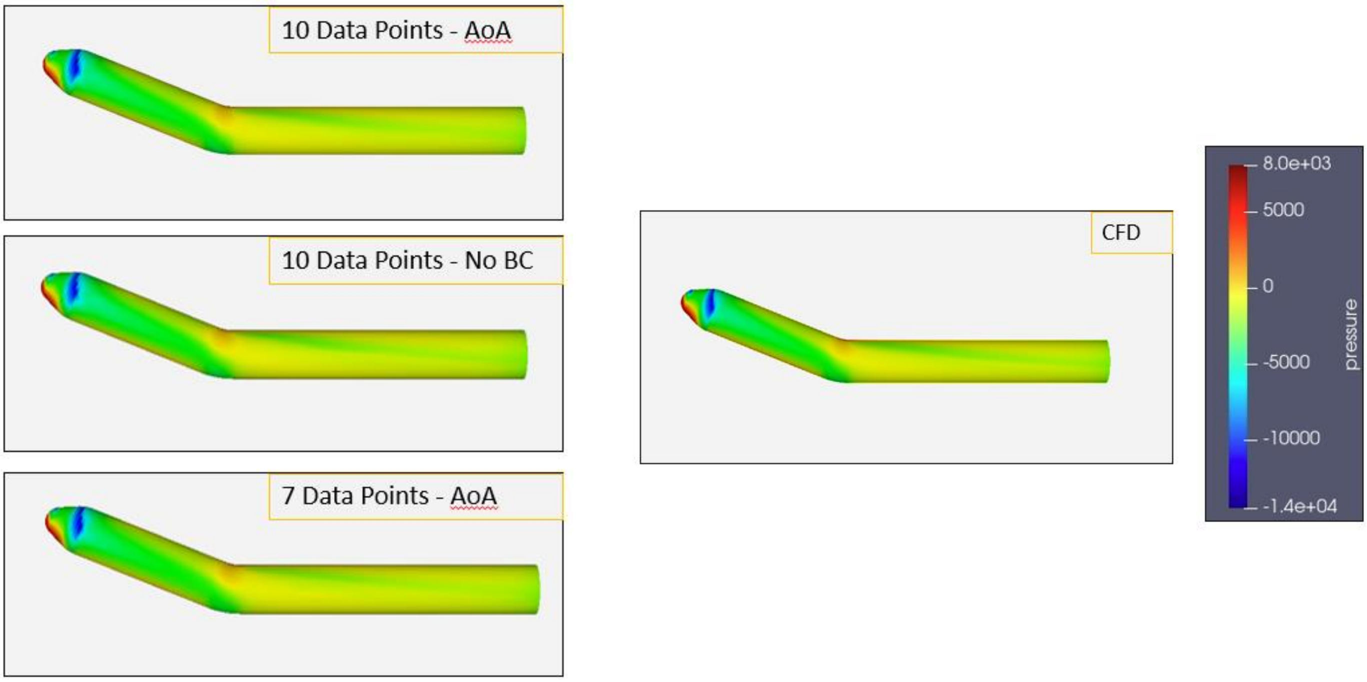


Figure 37: Maneuverable - Static Pressure Contours

AGARD Model

4.5.6 Non-Parametric Model Buildup

In this second case, the AGARD – C model is studied as a non-parametric geometry, meaning that unlike the maneuverable vehicle case, the shape does not change according to a parametric boundary condition such as the geometric nose angle. Instead, the aerodynamic predictions are derived solely from the set of CFD data points supplied, without an explicit parametric variable guiding the interpolation. The data points are all custom designs, inspired by the AGARD – B and the AGARD – C model. This allows us to evaluate how SimAI performs in reconstructing aerodynamic trends when only raw geometric–flow pairings are given as input.

The original 5 variations are all simulated in a Mach number of 0.8 and an angle of attack of 0°. For the assessment of the adaptability and robustness of the AI framework, three different SimAI models were created for this case. The same five baseline geometries were also simulated under sustained operating conditions: at a higher Mach number of 0.9 and at additional four different angles of attack. This dataset expansion provided a stronger basis for training and validation and led to the development of a full-range aerodynamic performance map (Figure 4.10). This map was instrumental in organizing the simulation results and in rigorously examining how SimAI interpolates and predicts an aerodynamic force under varying flow conditions.

Angles of Attack

Mach Numbers

A	0	4	8	10	20
0.8	V10	V16	V21	V7	V11
0.9	V26	V41	V46	V31	V36
B	**0**	**4**	**8**	**10**	**20**
0.8	V2	V17	V22	V9	V12
0.9	V27	V42	V47	V32	V37
C	**0**	**4**	**8**	**10**	**20**
0.8	V8	V18	V23	V3	V13
0.9	V28	V43	V48	V33	V38
D	**0**	**4**	**8**	**10**	**20**
0.8	V1	V19	V24	V4	V14
0.9	V29	V44	V49	V34	V39
E	**0**	**4**	**8**	**10**	**20**
0.8	V5	V20	V25	V6	V15
0.9	V30	V45	V50	V35	V40

Figure 38: Variations Map

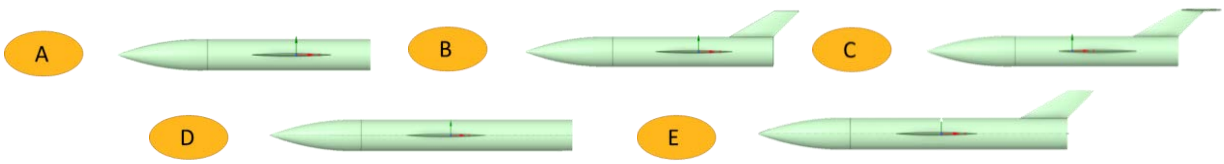


Figure 39: Variations

It is important to note that the AGARD – C, an extended version of AGARD – B with a tail attached, was the unseen geometry on which all the aerodynamic predictions were performed.

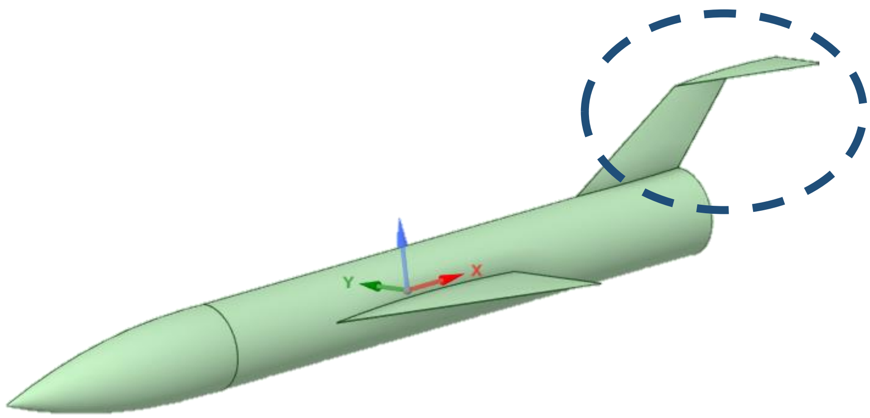


Figure 40: AGARD C

4.5.7 5 Data – Points

The initial set uses only 5 CFD points for training. All data sets are simulated in Mach 0.8 and angle of attack 0 degrees.

- 5 Data Points
- Mach 0.8 - AoA [0°]
- Domain of Analysis: Absolute Bounding Box
- Training Time < 7 days

The variation used during the testing of the model is highlighted in red. It is important to note that with such limited data points, SimAI has very little information upon which to base reconstruction of the nonlinear aerodynamic response.

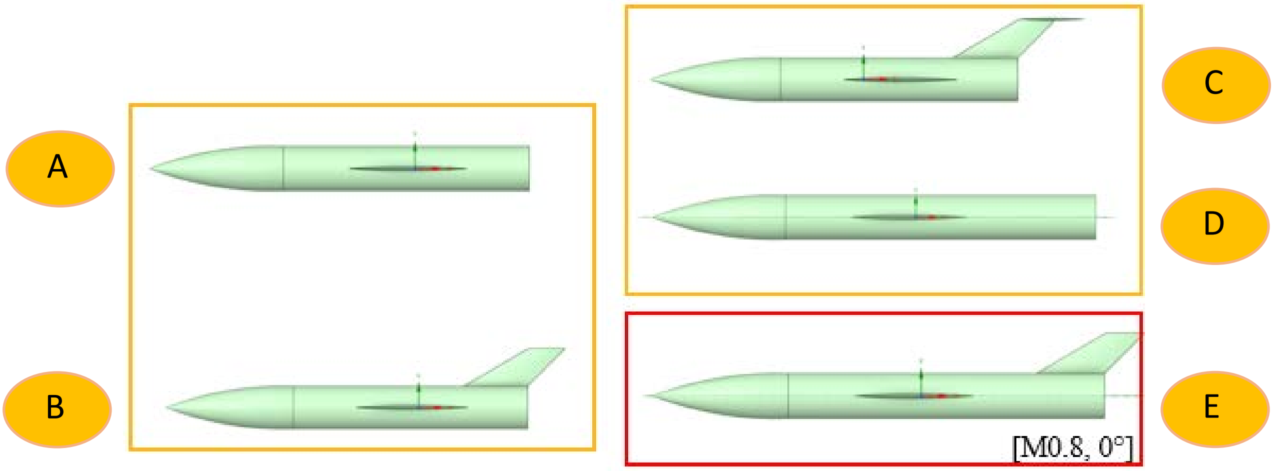


Figure 41: 5 Data Points

	CFD	SimAI	Error (%) SimAI vs CFD
Drag [X]	31.34	30.22	3.58
Lift [Y]	-9.50	-8.10	14.76

Table 15: 5 Data Points Results

4.5.8 25 Data – Points

The second case significantly increases the dataset to 25 points. The data are all in a Mach of 0.8 and each of the five is simulated in 5 different angles of attack. More specifically:

- 25 Data Points
- Mach 0.8 - AoA [0°, 4°, 8°, 10°, 20°]
- Domain of Analysis: Absolute Bounding Box
- Training Time < 7 days

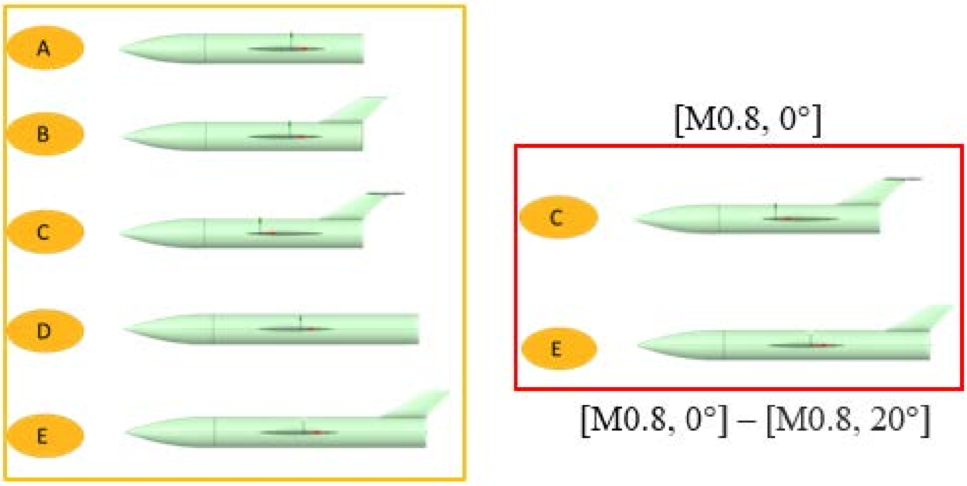


Figure 42: 25 Data Points

In the corresponding table, the SimAI predictions now align more closely with the CFD reference data across the aerodynamic range. The interpolation is smoother, and both drag and lift predictions exhibit smaller deviations from the unseen CFD points.

Upon closer examination, the improvement is most noticeable in lift predictions. Errors still exist in regions of steep aerodynamic variation, but the overall accuracy has improved substantially.

	CFD	**SimAI**	**Error (%) SimAI vs CFD**
Drag [X]	31.34	30.84	1.59
Lift [Y]	-9.50	-8.39	11.72

Table 16: 25 Data Points Results

This result highlights the strong dependence of AI model reliability on dataset density: by increasing coverage across the design space, SimAI reduces the extrapolation gap, leading to more trustworthy predictions.

4.5.9 50 Data – Points

The third case brings the dataset density to 50 points. Here, the plots show the best agreement between SimAI predictions and CFD data. The interpolated aerodynamic curves essentially overlap with the reference CFD across the full range of angles of attack.

- 50 Data Points
- Mach 0.8, 0.9 - AoA [0°, 4°, 8°, 10°, 20°]
- Domain of Analysis: Absolute Bounding Box
- Training Time < 7 days

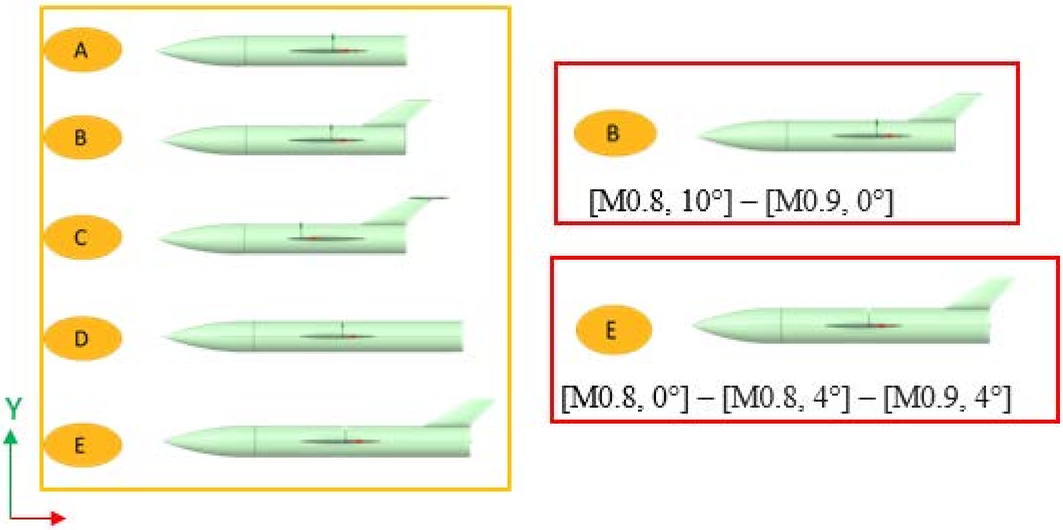


Figure 43: 50 Data Points

Divergence is still apparent at the extreme angles of attack, where very nonlinear flow characteristics may possibly not be precisely replicated even with 50 points. General predictive accuracy is nevertheless very good, and it is evident that for non-parametric geometries, SimAI directly and positively benefits from dataset richness.

In this case, the system accurately reproduces the aerodynamic performance of the AGARD model with near-CFD accuracy, demonstrating the scalability of AI-based prediction with more data, while also having a good agreement with the experimental results where applicable.

The plots represent drag, lift and moment coefficients respectively, plotted against angle of attack. The angles of attack used for the plots are from 0º up to 8º and the experimental data were taken from the widely used AGARD tables.

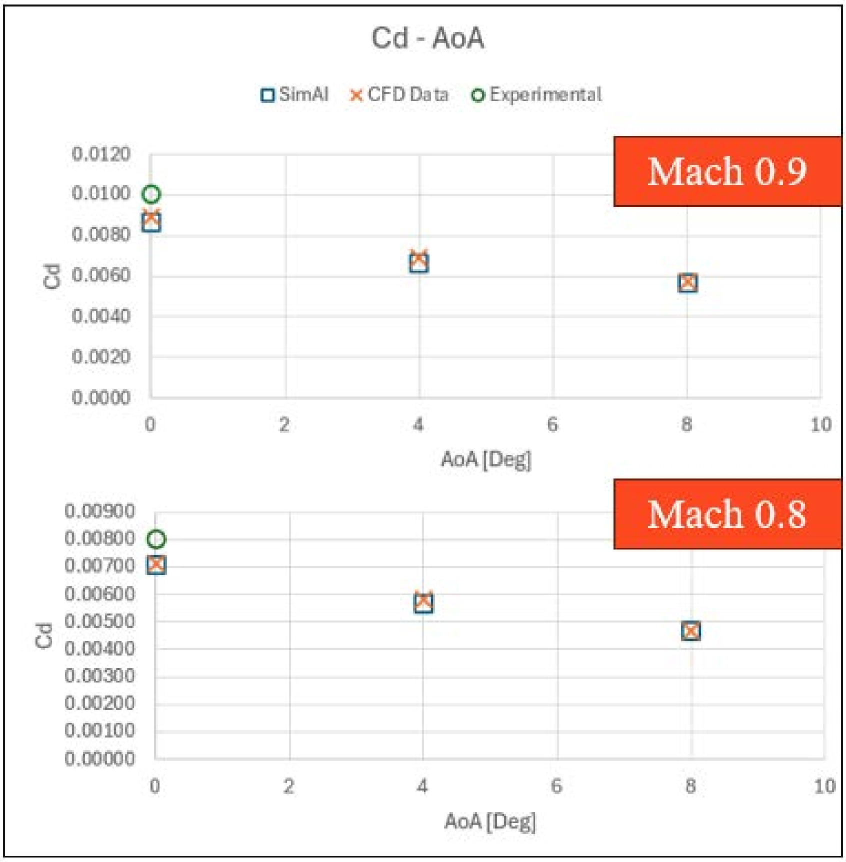


Figure 44: Cd vs AoA for 50 Data Points

The plots for both the Mach numbers of 0.8 and 0.9 indicate good agreement between the SimAI predictions (blue rectangles) and the CFD results (orange exes), while they also validate the experimental results where applicable, for Cd, Cl and Cm.

Figure 45: Cl vs AoA for 50 Data Points

Figure 46: Cm vs AoA for 50 Data Points

What is important to mention in the Cm versus AoA plots, is the behavior of the coefficient. It follows a trend where it starts at an initial value at 0º, then it is reduced for the angle of 4º and then it rises again for the 8º. SimAI captures this trend very well in its predictions, something that is very useful, since it enhances reliability.
Some wing contours of the additional feature of AGARD C, which was unseen and used for the predictions, are displayed below:

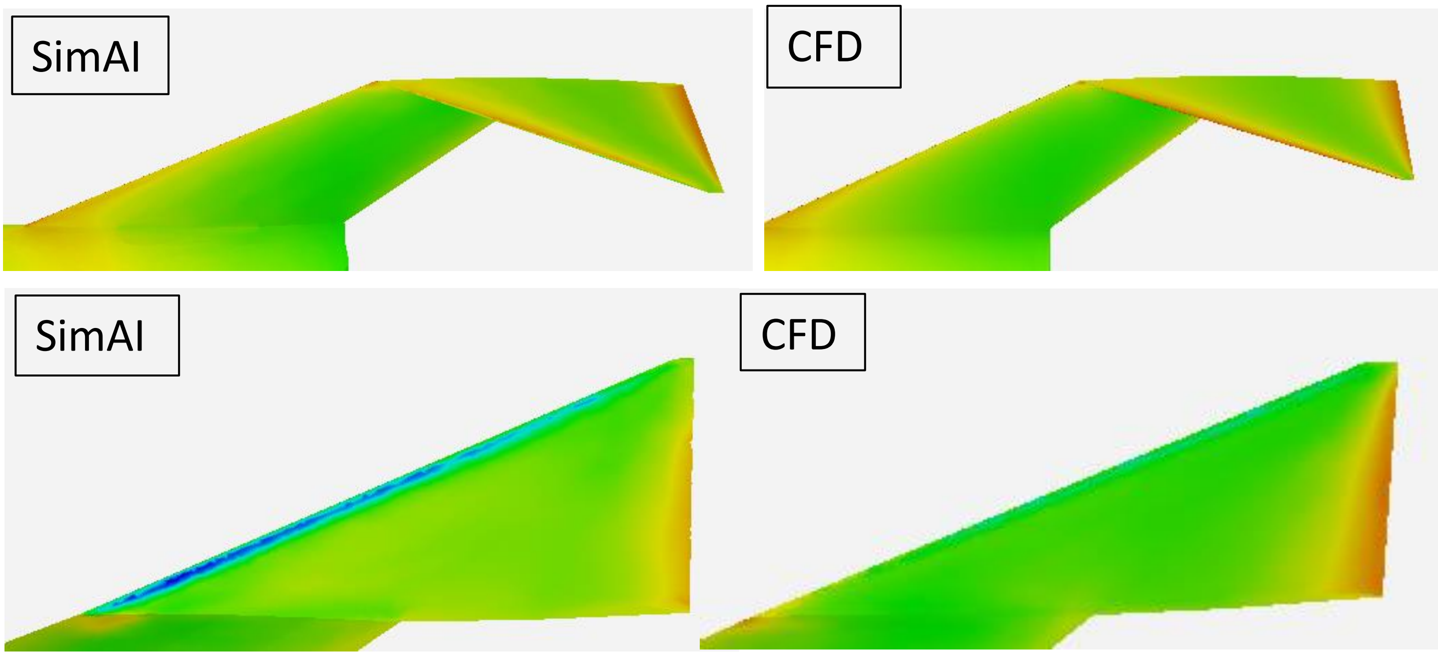


Figure 47: Static Pressure Contours

The top and bottom contours are on the same wing. The predictions on the top surface are very good. The bottom surface predictions indicate some excessive low prediction at the low side of the leading edge, which suggests some sensitivity.

Mean Error % SimAI to CFD of AGARD C [Unseen Geometry]				
	5 Sets of Data No BC	**25 Sets of Data AoA**	**50 Sets of Data AoA – Mach**	
	0.8	**0.8**	**0.8**	**0.9**
Drag Coef. Error	1.60	3.02	1.28	2.41
Lift Coef. Error	11.7	4.66	1.80	4.30
Moment Coef. Error	-	3.95	4.52	5.51

Table 17: Total AGARD C Results

A final set of contours is presented below, comparing the predictions on the top surface of the additional non – parametric feature in the three sub – models that were trained for this case.

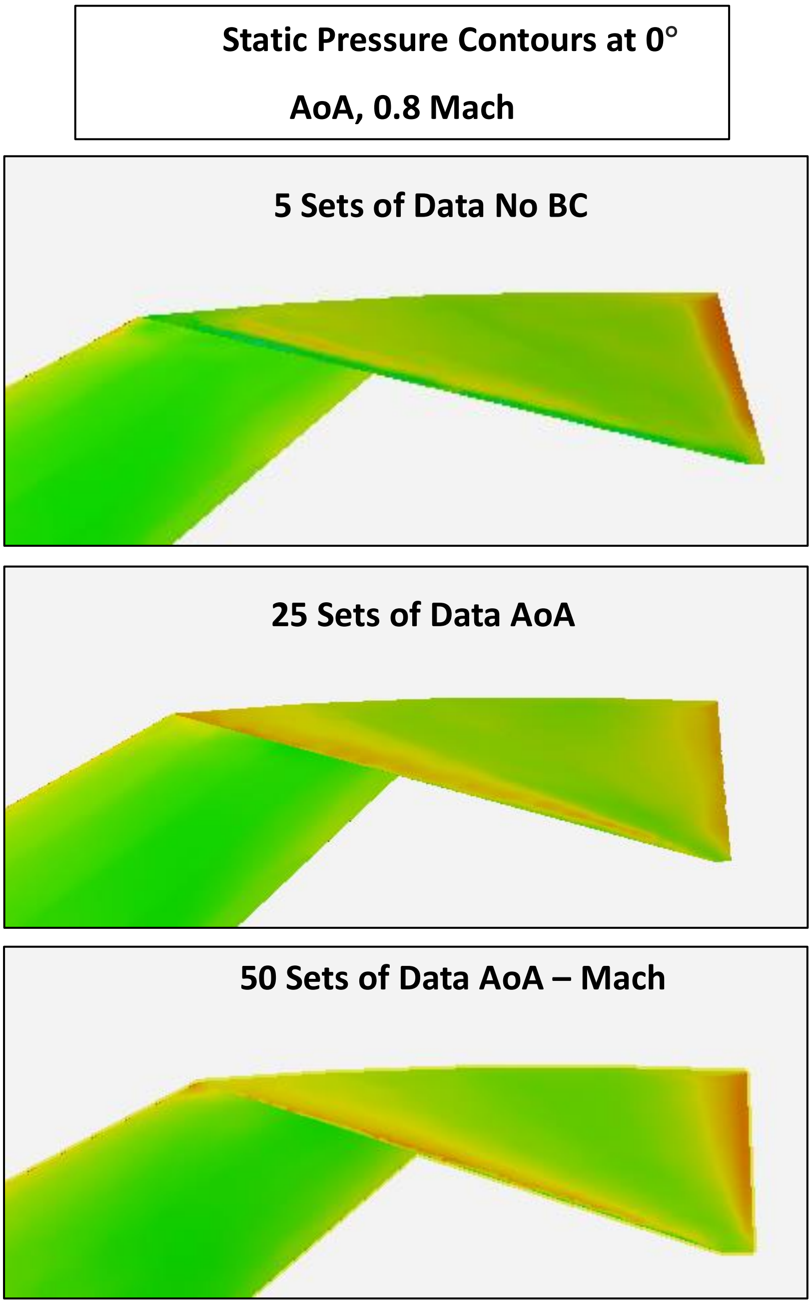


Figure 48: Static Pressure Contours Comparison

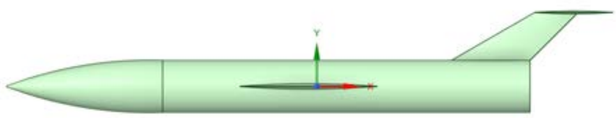


Figure 49: AGARD C Symmetry View

4.5.10 Discussion of Results

Comparing the three dataset configurations, a clear trend emerges:

- With 5 points, SimAI predictions provide a rough approximation of aerodynamic behavior but are insufficient for accurate design analysis, especially for lift.
- With 25 points, the predictive quality improves significantly, reaching a level that may be acceptable for preliminary aerodynamic assessments.
- With 50 points, SimAI predictions closely replicate CFD data, reducing errors to a minimum and providing confidence even for more detailed aerodynamic evaluations.

The AGARD non-parametric study demonstrates the importance of dataset richness in AI-based aerodynamic prediction. Sparse datasets (5 data points) provide only a coarse approximation of trends, limiting their usefulness in engineering decision-making. With moderate dataset sizes (25 points), SimAI achieves good predictive capability, especially for drag, making it suitable for conceptual phases of design. However, for high-fidelity applications, larger datasets (50 points or more) are essential, as they enable the model to reproduce CFD-level results with minimal error observed especially in lift coefficient.

5. Conclusions and Further Research

This thesis presented the development of 2 AI models in SimAI, an aerodynamics demonstrator which integrates high-fidelity CFD data sets with machine learning models for predicting aerodynamics. Workflows adopted for the HB-2 and AGARD standard cases ensured the generation of reliable training and validation data through rigorous meshing schemes, solver settings, and mesh-independence studies.

The results indicated that the behavior of aerodynamics can be emulated by AI models with encouraging accuracy at a very small computational cost compared to the conventional CFD. Sensitivity analysis highlighted that data density during training, boundary condition application, and model architecture selection are the predictors of the quality of predictions. The use of parametric and non-parametric examples ensured that data-driven models can learn to both controlled geometry variations and to general test cases.

The key results are as follows:

- AI-supported aerodynamic approaches provide remarkable computational cost reductions with preservation of prediction quality.
- Predictive accuracy largely depends on the representativeness and quality of training data.
- Hybrid approaches, by combining CFD knowledge and AI flexibility, are very promising for future applications.
- SimAI proved to be a viable proof-of-concept platform for improving aerodynamic analysis acceleration.

Future research is suggested in the following directions:

- Hybrid CFD–AI model formulation for better generalization and robustness.
- Formulation of advanced turbulence modeling techniques into AI-based frameworks.
- Scaling towards large-scale, diverse training data sets for facilitating scalable surrogate models.

- Flight simulation, control, and design optimization exploration in real-time.
- Incorporating uncertainty quantification and interpretability into AI-enabled aerodynamic prediction.

APPENDIX

A1. Continuity Equation

The continuity equation expresses the conservation of mass in a fluid system:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0 \qquad (A.1)$$

where ρ is the fluid density and u is the velocity vector. For incompressible flows (ρ=*constant*), this reduces to:

$$\nabla \cdot u = 0 \qquad (A.2)$$

A2. Derivation of the Navier – Stokes Equation

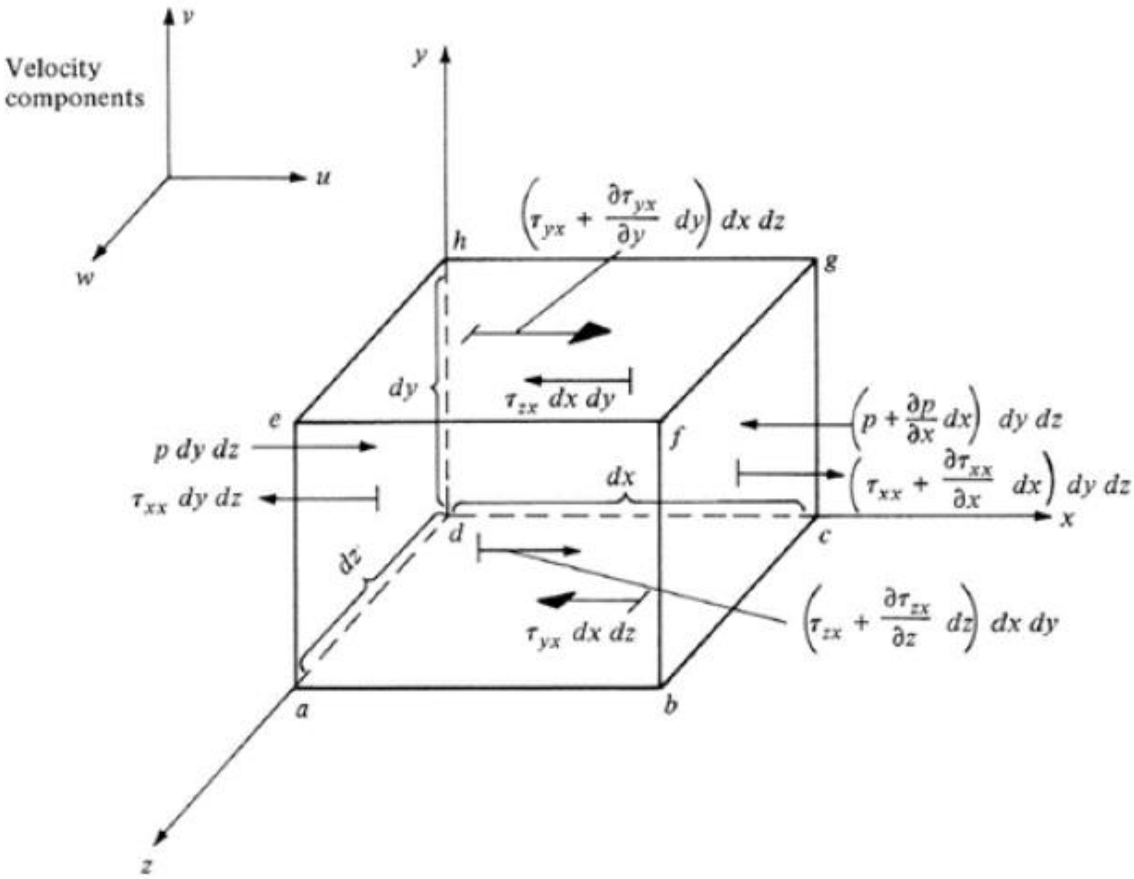


Figure 50: Forces Acting on a Fluid Element [19]

The Navier–Stokes equations result from the conservation of momentum applied to a control volume. Starting from Newton's second law:

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u \times u) = -\nabla p + \nabla \cdot \tau + \rho f \qquad (A.3)$$

where p is pressure, f is the body force per unit mass, and τ is the viscous stress tensor:

$$\tau = \mu \left[\nabla u + (\nabla u)T - \frac{2}{3}(\nabla \cdot u)I \right] \qquad (A.4)$$

These, combined with the continuity equation, form the complete set of governing equations for fluid motion. More details can be found on reference [19].

A3. Reynolds Decomposition

Reynolds decomposition is a mathematical technique used to separate the expectation value of a quantity from its fluctuations in turbulent flows. For a quantity u the decomposition would be:

$$u_i = \overline{u_i} + \acute{u_i} \qquad (A.5)$$

Where $\bar{u}$ is the mean (time-averaged) component of the velocity field and u' is the fluctuating component.

Applying this to the Navier–Stokes equations and averaging leads to the Reynolds-Averaged Navier–Stokes (RANS) equations:

$$\frac{\partial(\rho \overline{u_i})}{\partial t} + \frac{\partial\left(\rho \overline{u_i}\, \overline{u_j}\right)}{\partial xj} = -\frac{\partial \bar{p}}{\partial xi} + \frac{\partial}{\partial xj}\left(\mu \frac{\partial \overline{u_i}}{\partial xj} - \rho \overline{ui'uj'}\right) \qquad (A.6)$$

The additional term $-\rho \overline{ui'uj'}$ represents the Reynolds stresses, which require turbulence modeling for closure.

A4. Turbulence Models

A4.1 K – epsilon Realizable

The realizable k–ε model is an improved version of the standard k–ε turbulence model, designed to satisfy certain mathematical constraints on the Reynolds stresses and to provide better predictions for flows involving rotation, separation, and recirculation.

It solves transport equations for the turbulent kinetic energy k and the turbulent dissipation rate ε:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho k u_j)}{\partial xj} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + Gk - \rho\varepsilon \qquad (A.7)$$

$$\frac{\partial(\rho \varepsilon)}{\partial t} + \frac{\partial(\rho \varepsilon u_j)}{\partial xj} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + \rho C1 S\varepsilon - \rho C2\frac{\varepsilon^2}{k + \sqrt{\nu\varepsilon}} \qquad (A.8)$$

where:

- Gk is the production of turbulent kinetic energy,
- $\mu t = \rho C\mu(k^2/\varepsilon)$ is the turbulent viscosity, but in the realizable model, Cμ is not constant and is instead computed from a realizability condition,
- S is the mean strain rate.

Compared to the standard k–ε model, the realizable version:

- ensures that predicted Reynolds stresses remain physically consistent (realizable),
- improves predictions of rotating and separated flows,
- modifies the equation for ε and makes Cμ a function of the local flow field instead of a fixed constant.

REFERENCES

1. Ansys, Inc. “Ansys Fluent Theory Guide”, Release 2025 R1.
2. Ansys, Inc. “Fluent User’s Guide”, Release 2025 R1.

3.https://www.researchgate.net/publication/301673979_Shape_Optimization_of_a_Car_Body_for_Drag_Reduction_and_to_Increase_Downforce

4. https://www.grc.nasa.gov/www/k-12/BGP/boundlay.html

5. https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/what-is-drag/

6. https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/aerodynamic-force/

7. https://www.researchgate.net/publication/354771490_Wind_Tunnel_Testing_of_ONERA-M_AGARD-B_and_HB-2_Standard_Models_at_Off-Design_Conditions

8.https://www.researchgate.net/publication/356415323_Extension_and_Validation_of_NES_vs_IAI_Blow-Down_Cold_Hypersonic_Wind-Tunnel_including_High_Angle_of_Attack_Hypersonic_Generic_Missile_CFD_Prediction_Extension_and_Validation_of_NES_vs_IAI_Blow-Down_C

9. F.R. Menter, R. Lechner and A. Matyushenko “Best Practice: RANS Turbulence Modeling in Ansys CFD”, Version 1.0.

10. D. Damljanović, A. Vitić, and Đ. Vuković, "Testing of AGARD-B calibration model in the T-38 trisonic wind tunnel," *Scientific-Technical Review*, vol. 56, no. 2, pp. 52–62, 2006.

11. D. Damljanović, J. Isaković, and M. Miloš, "Inter-facility correlations of transonic test results of the AGARD-C standard wind-tunnel model," *Materials Today: Proceedings*, vol. 5, no. 13, pp. 26476–26481, 2018. doi: 10.1016/j.matpr.2018.08.102

12. Workflow and Usability for Fluids Application Best Practices using Ansys SimAI Platform (ABP) - Ansys Learning Hub

13. A. Reverberi and J. Procario, “Explaining SimAI: How AI is applied to numerical simulation in practice,” *Ansys Blog*, May 13, 2024. [Online]. Available: https://www.ansys.com/blog/explaining-simai

14. D. Ferguson and J. Procario, “Maximize Simulations With Secure, Cloud-native AI,” *Ansys Blog*, June 25, 2024. [Online]. Available: https://www.ansys.com/blog/maximize-simulations-with-secure-cloud-native-ai

15. Hills, R. A Review of Measurements on AGARD Calibration Models; AGARDograph, 64; Hill, A., Ed.; Aircraft Research Association: Bedford, UK, 1961.

16. https://www.grc.nasa.gov/www/k-12/airplane/normal.html

17. Ansys Learning Hub “Real External Flows – Lesson 4: Flow Separation and Reattachment”.

18. Ansys Learning Hub “Ansys CFD (Fluent & CFX) Turbulence Modelin, Lecture 03: Near Wall Modeling”.

19. Bruce R. Munson, Donald F. Young, Theodore H. Okiishi, “Foundamentals of Fluid Mechanics 8th edition”.