Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
aerospace engineering

Development of a Graphical Interface for Aerodynamic Simulations of NACA Profiles with OpenFOAM

Author: Thiago Vinicius Costa SIlva
supervisor: Prof. Dr.: Rafael Castilho Faria Mendes

Brasília, DF
2024

Thiago Vinicius Costa SIlva

Development of a Graphical Interface for Aerodynamic Simulations of NACA Profiles with OpenFOAM

Monograph submitted to the undergraduate course in aerospace engineering from the University of Brasília, as a partial requirement to obtain the Title Bachelor's degree in aerospace engineering.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr.: Rafael Castilho Faria Mendes

Brasília, DF

2024

Thiago Vinicius Costa SIlva
Development of a Graphical Interface for Aerodynamic Simulations of NACA Profiles with OpenFOAM/ Thiago Vinicius Costa SIlva. – Brasília, DF, 2024-
4 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr.: Rafael Castilho Faria Mendes

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2024.

1. Airfoil. 2. Openfoam. I. Prof. Dr.: Rafael Castilho Faria Mendes. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Development of a Graphical Interface for Aerodynamic Simulations of NACA Profiles with OpenFOAM

CDU 02:141:005.6

Thiago Vinicius Costa SIlva

Development of a Graphical Interface for Aerodynamic Simulations of NACA Profiles with OpenFOAM

Monograph submitted to the undergraduate course in aerospace engineering from the University of Brasília, as a partial requirement to obtain the Title Bachelor's degree in aerospace engineering.

Approved work. Brasília, DF, :

Prof. Dr.: Rafael Castilho Faria Mendes
supervisor

Guest 1

Guest 2

Brasília, DF
2024

Acknowledgements

I would like to first thank God for all the opportunities and for giving me strength. To my mother, Liecia Costa Bezerra, who always fought for me to pursue my dreams and gave me all the support needed throughout my life. I love you, Mom. To my sisters, Thaís Marianne Costa Silva dos Santos and Thaina Marianne Costa Silva, who have always been by my side. I love you both.

I also want to thank the friendships I've built along the way, especially Carolina Rego de Oliveira, who has supported and helped me since the beginning, including during the process of the exchange to Spain. To my friend Patricia, who has been with me since high school. To the "Panelinha,"comprised of Angélica, Ana Carolina, Júlia Messias, and Maria Eduarda, who have inspired me to always give my best. If I am the average of the five people I spend the most time with, then these people have certainly helped me to be better, more dedicated, and dream bigger.

I would like to express my gratitude to my girlfriend, Júlia Vargas, who supported me throughout the process of the exchange and helped me achieve my dreams.

I extend my thanks to Prof. Dr. Rafael Castilho Faria Mendes, who accepted me as his student at University of Brasilia (UNB) and provided me with the opportunity to study at the University of Cadiz. Also, to Prof. Dr. Marianela Machuca Macias, who was my advisor and assisted me throughout this journey.

Essentially, all models are wrong, but some are useful.

"Essencialmente, todos os modelos estão errados, mas alguns são úteis."
(Box, George EP; Norman R. Draper (1987). Superfícies empíricas de construção de modelos e resposta, p. 424.)

Abstract

This work presents the development and validation of an automation program for aerodynamic simulations, using the OpenFOAM software in conjunction with a Python application. The program was designed to automate the process of mesh creation, boundary condition definition, and simulation execution, enabling efficient analysis of aerodynamic profiles. Simulations were conducted using different NACA profiles, under various speed and angle of attack conditions. The simulation results were compared with literature data, demonstrating good agreement. Additionally, the program allowed for a comparative analysis between NACA profiles, revealing differences in terms of lift and drag generation. Overall, this work offers a significant contribution to the field of aerodynamics, providing an effective and automated tool for studies and analyses related to profile aerodynamics.

Key-words: Aerodynamics. CFD. NACA airfoil profile. Simulation automation. Python.

Resumen

Este trabajo presenta el desarrollo y la validación de un programa de automatización para simulaciones aerodinámicas, utilizando el software OpenFOAM en conjunto con una aplicación Python. El programa fue diseñado para automatizar el proceso de creación de mallas, definición de condiciones de contorno y ejecución de simulaciones, lo que permite un análisis eficiente de perfiles aerodinámicos. Las simulaciones se realizaron utilizando diferentes perfiles NACA, bajo diversas condiciones de velocidad y ángulo de ataque. Los resultados de la simulación se compararon con datos de la literatura, demostrando un buen acuerdo. Además, el programa permitió un análisis comparativo entre perfiles NACA, revelando diferencias en términos de generación de sustentación y arrastre. En general, este trabajo ofrece una contribución significativa al campo de la aerodinámica, proporcionando una herramienta efectiva y automatizada para estudios y análisis relacionados con la aerodinámica de perfiles.

Palabras clave: Aerodinámica. CFD. Perfil aerodinámico NACA. Automatización de simulación. Python.

Resumo

Este trabalho apresenta o desenvolvimento e validação de um programa de automação para simulações aerodinâmicas, utilizando o software OpenFOAM em conjunto com uma aplicação Python. O programa foi projetado para automatizar o processo de criação de malhas, definição de condições de contorno e execução de simulações, permitindo uma análise eficiente de perfis aerodinâmicos. Foram realizadas simulações utilizando diferentes perfis NACA, em diversas condições de velocidade e ângulo de ataque. Os resultados das simulações foram comparados com dados da literatura, demonstrando uma boa concordância. Além disso, o programa permitiu uma análise comparativa entre perfis NACA, revelando diferenças em termos de geração de sustentação e arrasto.

Palavras-chaves:Aerodynamics. CFD. NACA airfoil profile. Simulation automation. Python.

List of illustrations

List of tables

List of Abbreviations and Acronyms

CFD	Computational Fluid Dynamics
FVM	Finite Volume Method
LES	Large Eddy Simulation
DNS	Direct Numerical Simulation
RANS	Reynolds-Averaged Navier-Stokes
DES	Detached Eddy Simulation
PISO	Pressure-Implicit Split-Operator
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
NACA	National Advisory Committee for Aeronautics
NASA	National Aeronautics and Space Administration
SAE	Society of Automotive Engineers
CAD	Computer-Aided Design
OpenFOAM	Open source Field Operation And Manipulation
SST	Shear Stress Transport
SA	Spalart-Allmaras
FVM	Finite Volume Method
MAPE	Mean Absolute Percentage Error
SA	Spalart-Allmaras

Symbols

c	Chord
α	Angle of attack
$Alpha$	Angle of attack
ρ	Pressure
S	Area
τ	Shear stress
R	Resultant force
V_∞	airflow velocity
D	Drag
N	Normal force
A	Axial force
L	Lift
ρ_∞	specific mass
q_∞	dynamic pressure
C_L	Coefficient of lift
C_D	Coefficient of drag
C_M	Coefficient of moment
Y_c	Camber line ordinate
a_0, a_1, a_2, a_3, a_4	Polynomial coefficients
x	Coordinate along the chord length
y_t	Airfoil thickness
t	Maximum thickness of the airfoil (as a fraction of the chord)
β	angle of inclination

M	Mach number
c_e	speed of sound in the study environment
Re	Reynolds number
τ_R	Reynolds stress tensor
u	Velocity component in the x-direction
v	Velocity component in the y-direction
p_k	Pressure component in the k-direction
U	Flow velocity in the x-direction
V	Flow velocity in the y-direction
P_k	Pressure in the k-direction
u'	Velocity fluctuation component in the x-direction
v'	Velocity fluctuation component in the y-direction
p'_k	Pressure fluctuation component in the k-direction

Summary

1 Introduction

Engineering seeks to solve complex problems and innovate in different sectors. From structural design to the implementation of advanced technologies, engineers employ a variety of tools and methods to achieve these objectives. Within this broad context, aerospace engineering stands out as an area that frequently faces significant challenges, requiring precise analyses and creative solutions to optimize the performance of aircraft and spacecraft (COURTNEY, 2014).

Practical experiments play a vital role in validating theories and testing prototypes. However, conducting such experiments can be limited by factors such as high costs, complexity, and the need for specialized infrastructure. This is where computational simulation, especially Computational Fluid Dynamics (CFD), becomes an essential tool (MIRANDA, 1984).

While various computational packages are available for tasks such as computer-aided design, scientific calculations, fluid dynamics, heat transfer, and electromagnetism, many of these are proprietary and require expensive annual licenses (LAKHAN; JHUNJHUNWALA, 2008; PIRES; ROGERS, 2002). Therefore, to conduct studies affordably and economically, the use of open-source tools is primarily proposed.

In aerospace engineering, the simulation of aerodynamic profiles is crucial for understanding and optimizing the behavior of objects interacting with fluid flow (COURTNEY, 2014). Therefore, omputational simulation plays a pivotal role in the field, offering an efficient approach to analyzing complex phenomena and enhancing the development and performance of aerospace technologies.

Aerodynamic profiles play a fundamental role in various applications, from aircraft and spacecraft design to wind turbine conception. Essentially, these shapes are designed to optimize the interaction between an object and the fluid around it, usually air, to generate lift, reduce drag, and control the direction of movement. The study of these profiles is motivated by the need to understand and predict how different geometries influence aerodynamic behavior, consequently enabling the development of more efficient and safer technologies. By comprehending the characteristics of aerodynamic profiles and how they affect the airflow around them, engineers can design more efficient aircraft, more productive wind turbine blades, and even more aerodynamic land vehicles, driving significant advances in various engineering fields (ANDERSON, 2001).

However, effective use of aerodynamic simulation software also presents challenges and difficulties. Accurate modeling of airflow behavior around complex objects requires deep knowledge of computational fluid dynamics and experience in selecting and applying different mathematical models. Additionally, validating simulated results is a critical step, as computational models are not always able to accurately represent all aspects of real flow (AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS, 1998).

Other challenges include the initial stage of geometry creation, which involves precise definition and extraction of coordinates describing the aerodynamic profile, demanding a high level of rigor and precision as any errors can directly impact the validity and reliability of simulation results. Additionally, the computational mesh used in the simulation is crucial for result accuracy, requiring a delicate balance and a deep understanding of numerical discretization principles to generate a mesh that efficiently captures fluid dynamics phenomena while maintaining computational efficiency (AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS, 1998).

The learning curve and familiarization with simulation software pose significant challenges, requiring a substantial investment of time and effort to master the complex tools and techniques involved. To address these difficulties, software developers are constantly working to improve modeling and mesh generation techniques, as well as the accuracy of mathematical models used in simulations (ROACHE, 2009; BAKER, 2005).

In the Aerospace Engineering course, various studies are conducted on aerodynamic profiles, where the process of searching for or creating profile coordinates, mesh generation, domain definition, and simulation configuration are employed. These steps are repeated when making modifications to study different angles, ultimately resulting in the manual compilation of data.

Furthermore, the Mamutes do Cerrado Aerodesign team, a competition team from the Aerospace Engineering course at the University of Brasília, was founded in 2015. Through its activities, the team aims to promote the emergence of culture and interest in the aeronautical sector within Brazilian society. Aligned with the guidelines of the SAE Brasil AeroDesign Competition, the team participates in the competition by providing a unique learning opportunity in the aeronautical field through the conception of a multidisciplinary project involving the construction of competition aircraft.

The team is divided into various areas: Administration, Finance, Marketing, Aerodynamics, Loads and Aeroelasticity, Performance, Stability and Control, Electronics, Structure, and CAD. Many students join at the beginning of the course without prior knowledge of simulation software. We seek to foster the aeronautical field by integrating university students into the environment, promoting the development of team spirit, forming future professionals with full leadership, planning, and articulation capabilities for new ideas and projects, as well as building ethical and professional behavior for its members.

In this context, our team specifically focuses on meeting the requirements of the SAE Brasil Aerodesign competition related to aerodynamic simulations, which include:

- Selection and/or design of airfoils;
- Determination of aerodynamic coefficients;

- General aerodynamic evaluations.

These criteria are outlined in Appendix 3 of the regulations for the 26th SAE Brasil Aerodesign Competition 2024 for the Regular, Advanced, and Micro classes (SAE Brasil, 2024).

However, since many students join at the beginning of the course without prior knowledge of simulation software, the lack of a graphical interface in OpenFOAM can hinder familiarity with the program. Additionally, the repetitiveness of the aerodynamic profile simulation process and the need for validations pose challenges. Given this context, developing a program that automates repetitive processes and facilitates initial contact with OpenFOAM through a user-friendly interface can allow a better understanding of the software's operation and the standarts for creating aerodynamic simulations.

1.1 Objectives

The work consists of developing intuitive and accessible software to automate computational fluid dynamics simulations, especially geared toward NACA four-digit airfoil profiles. The goal is to provide a tool that simplifies and democratizes the access to aerodynamic analyses, reducing the technical barrier for users with no prior CFD simulation experience. This software will be particularly valuable to members of aeromodelling teams in competitions, who are often students in the early years of technical or engineering courses. Additionally, the program will function as an introductory platform to OpenFOAM, one of the most used open-source CFD software, facilitating the initiation and familiarization of users with advanced simulation tools.

To achieve these objectives, the software will incorporate the following key features:

- Automated generation of the aerodynamic profile based on specific NACA parameters.
- Conversion of geometries to the format compatible with OpenFOAM.
- Automated creation of meshes and execution of simulations with defined boundary conditions.
- Compilation of simulation data into detailed reports, including analyses of aerodynamic components such as lift and drag.

2 Literature Review

The study of aerodynamic profiles is a crucial field in aerodynamics, with applications ranging from aeronautics to wind turbine engineering. The earliest systematic studies of aerodynamic profiles date back to the early 20th century, with pioneering work by researchers such as Ludwig Prandtl and his boundary layer theories. The formulation of specific profiles, such as the NACA (National Advisory Committee for Aeronautics) profiles, was developed in the 1920s and 1930s, providing a standard basis for aerodynamic research. The NACA profiles, developed between 1930 and 1950, are characterized by their series encoding (e.g., 4-digit, 5-digit NACA) describing their geometry and aerodynamic characteristics. Among these, the NACA 0012 is one of the most studied profiles due to its simplicity and relevance in various aerodynamic contexts (ANDERSON, 2011).

> "The NACA profiles were developed from systematic studies of aerodynamics, providing a standardized basis for wing design and analysis. These profiles, particularly the NACA 0012, have been widely used and studied due to their well-defined aerodynamic characteristics."(ABBOTT; DOENHOFF, 1959)

The NACA 0012 profile is a symmetrical profile with a maximum thickness of 12 percent of the chord. Its simple geometry makes it ideal for experimental and numerical studies. The aerodynamics of the NACA 0012 profile is a widely studied topic in both physical experiments and numerical simulations, due to its simple and well-documented geometry. Generally, aerodynamic profiles are evaluated mainly in terms of lift, drag, and moment coefficients. These coefficients depend on variables such as angle of attack, Reynolds number, and Mach number. Additionally, it is important to study the flow around the profiles, as it involves observing phenomena such as boundary layer separation, the transition from laminar to turbulent flow, and vortex formation (MCALISTER; TAKAHASHI, 1991).

In regard to experimental studies, Ladson (1988) conducted extensive tests in the NASA Langley wind tunnel, measuring the aerodynamic coefficients of the NACA 0012 profile at different angles of attack and Reynolds numbers, providing a fundamental database for the validation of numerical models. On the other hand, flow visualization around the NACA 0012 profile has been the subject of numerous studies, providing a detailed understanding of associated aerodynamic phenomena. McAlister e Takahashi (1991) performed a comprehensive analysis using flow visualization techniques to study flow behavior under oscillatory conditions and at different Mach numbers, observing flow separation and boundary layer transition. Broeren e Bragg (2001) investigated the variation of unsteady separation flow along the profile using advanced visualization techniques, revealing three-dimensional characteristics during separation.

Lee e Su (2010) employed Particle Image Velocimetry (PIV) and smoke visualization to examine flow separation under low Reynolds number conditions, highlighting flow structures and transition phenomena. Rinoie e Takemura (2004) used fluorescence and photography to analyze flow behavior under dynamic inflow conditions, providing valuable insights into vortex formation and flow separation. Finally, Brandon e Seidel (1988) applied visualization techniques such as high-speed photography and smoke visualization to study leading-edge vortices on an oscillating NACA 0012 profile, focusing on unsteady flow dynamics. These studies have been crucial for validating numerical models and optimizing aerodynamic designs, underscoring the importance of flow visualization in aerodynamic research.

The numerical simulation of the NACA 0012 profile has undergone significant evolution, starting with panel methods and ideal flow approaches and progressing to more advanced techniques incorporating turbulence models. The early approaches employed panel methods, which are effective in predicting potential flow around aerodynamic profiles under ideal flow conditions, providing a quick and efficient solution for pressure distribution on the profile surface. However, these methods do not capture complex phenomena such as flow separation and boundary layer transition. To address these limitations, ideal flow simulations based on the Euler equations were introduced, which improve flow representation by considering its nonlinearity, although they still omit viscosity effects (KHALID et al., 2015).

The numerical simulation of aerodynamic profiles, such as the NACA 0012 profile, has significantly advanced due to the implementation of various turbulence models. These models have been essential for accurately predicting turbulent flow behavior around profiles. Among the most prominent models are the Spalart-Allmaras model, the k-ε and k-ω models, and the k-ω SST (Shear Stress Transport) model.

The Spalart-Allmaras model, developed by Spalart and Allmaras, is a one-equation turbulence model designed specifically for external flow applications with moderate separation. This model is known for its low computational cost its ability to adequately predict the boundary layer and moderate separation and recirculation phenomena. It is especially useful in aerodynamic studies of profiles such as the NACA 0012 due to its simplicity and effectiveness in flows with these characteristics (SPALART; ALLMARAS, 1992).

On the other hand, the k-ε and k-ω models are two-equation turbulence models that solve the transport of turbulent kinetic energy (k) along with its dissipation rate (ε) or the specific dissipation frequency (ω), respectively. The k-ε model, extensively described by Launder e Spalding (1972), is robust and stable in highly turbulent flows, although it may encounter difficulties in flows with separation and large pressure gradients. In contrast, the k-ω model, as proposed by Wilcox (1988), provides better predictions in flows with separation and the vicinity of solid surfaces due to its formulation that better adapts to the near-wall boundary layer.

The k-ω SST model, developed by Menter (1994a), combines the advantages of the k-ε and k-ω models, utilizing k-ω in the near-wall region and k-ε in the free-flow region. This hybrid

approach significantly improves predictions in flows with separation and high-pressure gradient conditions, making it particularly effective for aerodynamic profiles like the NACA 0012. The k-ω SST model is known for its accuracy and robustness, providing more stable and reliable results.

Numerous studies have applied these turbulence models to analyze the NACA 0012 profile, validating their results with experimental data. For example, Genc, Karasu e Kaynak (2010). used the Spalart-Allmaras model to simulate the flow around the NACA 0012 at different angles of attack, finding that this model adequately predicts flow separation and reattachment under laminar and transition flow conditions. Additionally, Rumsey, Gatski e Sellers (2001) assessed the flow over the NACA 0012 profile using the k-ε and k-ω models, finding that the k-ω model provided better prediction of boundary layer and separation phenomena compared to the k-ε model.

The k-ω SST model has been widely validated in numerous studies, such as the one conducted by Menter, Kuntz e Langtry (2003), which demonstrated its effectiveness in predicting transient flows and separation in the NACA 0012 profile. These studies have shown that the k-ω SST model handles flow separation and turbulent transition better, providing more accurate and stable results in complex flows.

On the other hand, Large Eddy Simulations (LES) and Direct Numerical Simulations (DNS) offer a more detailed resolution of turbulent structures and transient phenomena in the flow. Although more computationally demanding, studies like those of Spalart e Allmaras (1992) and Khalid et al. (2015) have demonstrated the capability of these techniques to capture complex flow dynamics around the NACA 0012 profile.

The application of the NACA 0012 aerodynamic profile in Computational Fluid Dynamics has been extensively investigated, with a particular focus on its simulation using OpenFOAM, an open-source platform. Studies such as that of Javaherian e Sahin (2012) utilized OpenFOAM to conduct detailed simulations of transient flow over the profile, evaluating boundary layer transition and flow separation effects. These studies employed various turbulence models, including the Spalart-Allmaras model and the k-ω and k-ω SST models, and demonstrated good agreement with experimental data.

Furthermore, Nilsson et al. (2015) conducted a comprehensive comparison of turbulence models using OpenFOAM to simulate flow around the NACA 0012. Their study assessed the k-ε, k-ω, and k-ω SST models under different flow conditions, concluding that the k-ω SST model provided the most accurate predictions in terms of aerodynamic coefficients. Other studies, such as that of Zhang, Li e Zhu (2014), focused on the simulation of unsteady flows around the NACA 0012 profile using OpenFOAM and Large Eddy Simulation (LES) techniques, highlighting OpenFOAM's ability to capture transient phenomena and large-scale turbulent structures.

The validation of OpenFOAM in simulations of aerodynamic profiles was addressed by

Tabor, Smith e Brown (2013), who conducted a detailed study comparing OpenFOAM results with experimental data and other numerical models. This study established the accuracy and reliability of OpenFOAM in predicting aerodynamic coefficients and flow separation phenomena. Lopez, Rodriguez e Garcia (2016) investigated the effects of surface roughness on the aerodynamic behavior of the NACA 0012 profile using OpenFOAM, providing a deeper understanding of the effects of roughness in practical applications.

In summary, the reviewed studies demonstrate OpenFOAM's capability to provide accurate and detailed results, continuously validated against experimental data and other numerical models, allowing researchers to explore and understand a wide range of aerodynamic phenomena, from boundary layer transition to surface roughness effects (TABOR; SMITH; BROWN, 2013; LOPEZ; RODRIGUEZ; GARCIA, 2016).

3 Theory

3.1 Aerodynamic Profiles

In the context of an airfoil project, the fundamental requirement is to optimize the design to provide high lift in relation to drag. Increasing the lift coefficient not only boosts thrust but also contributes to reducing the drag coefficient, thus promoting efficient airflow passage. Airfoils from aircraft are subject to a sudden loss of lift, known as a stall, resulting in power loss. Additionally, these airfoils face rapid degradation with increased thickness (TANGLER; SOMERS, 1995).

The camber line is the set of points equidistant from the airfoil's upper and lower surfaces. The most forward and rearward points on the camber line are located at the leading edge (windward region) and trailing edge, respectively. The straight line connecting these two edges is called the chord line of the airfoil, and the distance between the leading and trailing edges, measured along the chord line, is the chord, represented by c. The angle of attack, α, is defined as the angle between the relative wind speed, denoted in the figure as U_{rel}, is also commonly represented as V_{∞}, and the chord line.

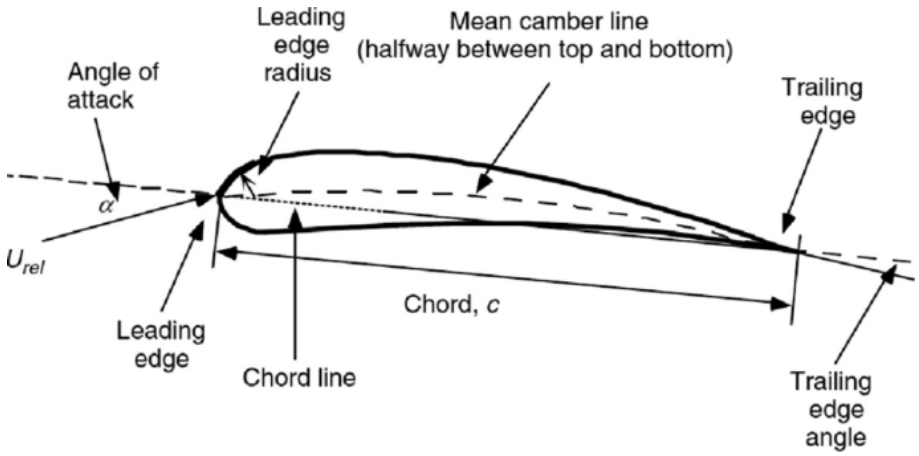

Fig. 1 – Parts of an airfoil. Source: Manwell, McGowan e Rogers (2009)

3.1.1 Aerodynamic Coefficients of a Profile

The aerodynamic forces on a body subjected to flowing air arise solely from two effects (ANDERSON, 2001):

- Pressure distribution on the body (normal to the surface);
- Shear stress along the body (tangential to the surface).

As shown in Fig 2, the pressure ρ acts normal to the surface, and the shear stress τ acts tangential to the surface. Shear stress is due to the "tugging action"on the surface, which is caused by friction between the body and the air.

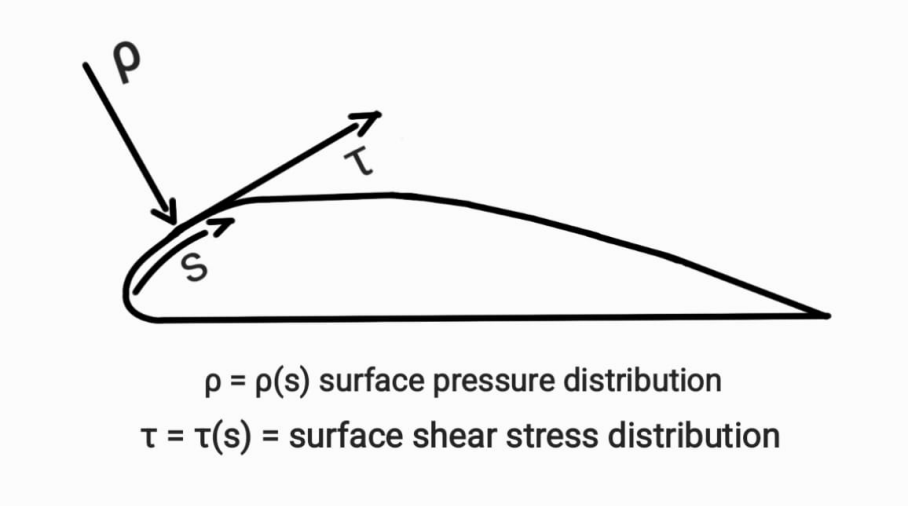

Fig. 2 – Pressure and shear stress on an aerodynamic surface. Source: Author based on Anderson (2001)

The effect of these pressure and shear stress distributions integrated over the entire body surface results in the resultant force R and moment M. The force R can be decomposed into two components relative to the airflow velocity V_∞ as shown in Fig 3. The first is L, called lift and perpendicular to V_∞. The second is D, called drag and parallel to V_∞. If R is decomposed in relation to the chord c (distance between the leading and trailing edges of the body), we have: N, called the normal force and perpendicular to c; and A, called the axial force and parallel to c.

The angle of attack α is defined as the angle at which the aerodynamic profile attacks the flow. This is measured through the angle between the chord c and the airflow velocity V_∞. It is also given by the angle between L and N and between D and A. The relationships between the components and α are given below:

$$L = N\cos\alpha - A\sin\alpha \tag{3.1}$$

$$D = N\sin\alpha + A\cos\alpha \tag{3.2}$$

To simplify calculations, aerodynamic forces and moments are made dimensionless, what we call aerodynamic coefficients. Considering the specific mass ρ_∞ and the velocity V_∞ of the flow away from the body, we define the quantity called dynamic pressure, represented in Equation 3.3 and having units of pressure (N/m^2) (ANDERSON, 2001).

$$q_\infty \equiv \frac{1}{2}\rho_\infty V_\infty^2 \tag{3.3}$$

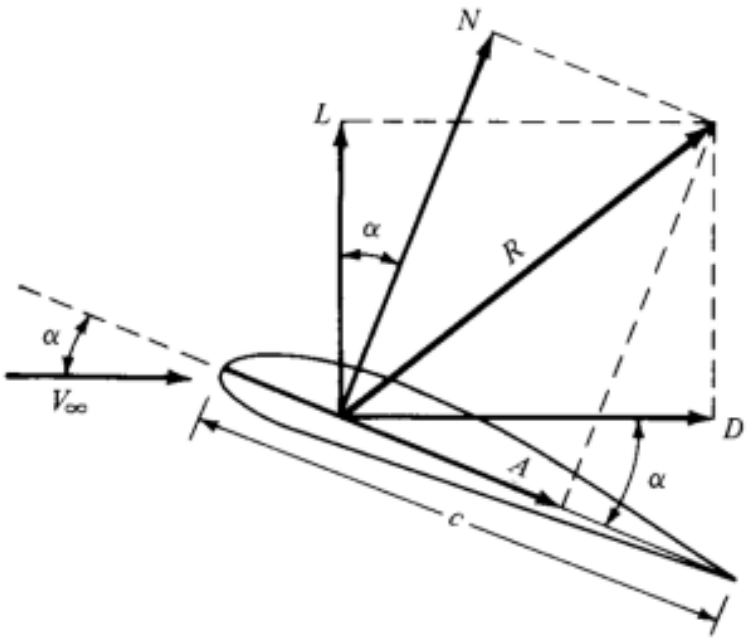

Fig. 3 – Resultant aerodynamic force and its components. Source: Anderson (2001)

Considering S as the reference area and l as the reference length, we find that the aerodynamic coefficients are the lift coefficient (Eq. 3.4), C_L, the drag coefficient (Eq. 3.5), C_D, and the moment coefficient (Eq. 3.6), C_M.

$$C_L \equiv \frac{L}{q_\infty S} \tag{3.4}$$

$$C_D \equiv \frac{D}{q_\infty S} \tag{3.5}$$

$$C_M \equiv \frac{M}{q_\infty S l} \tag{3.6}$$

3.1.2 Analysis of Coefficients

Polar curves illustrate the relationship between lift, which measures an aircraft's ability to rise, and drag, which measures resistance to motion through the fluid, at different angles of attack (α).

The phenomenon of boundary layer separation results in the primary consequence on aerodynamic bodies, stall, which is a phenomenon caused by boundary layer separation, leading to a decrease in lift force on the aerodynamic surface due to incident flow. Boundary layer separation usually occurs due to the high angle of attack formed by the surface relative to the airflow, or due to the low relative velocity between the flow and the body. The reduction in lift occurs due to flow separation, which causes the flow direction to reverse in certain areas of the aerodynamic surface, thereby reducing the suction generated by the air. This results in a significant reduction

in wing lift capacity, while considerably increasing aerodynamic drag. This can be observed by comparing the figures before stall, Fig: (4), and after stall, Fig: (5), (RUIZ; SMITH; JOHNSON, 2011).

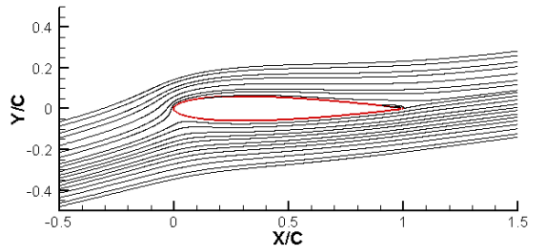

Fig. 4 – NACA 0012 before stall. Source: (THILEEPANRAGU et al., 2010)

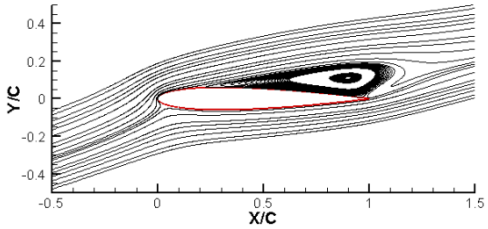

Fig. 5 – NACA 0012 with separation. Source: (THILEEPANRAGU et al., 2010)

The C_L increases as the angle of attack grows, reaching a maximum value ($C_{L_{\max}}$). However, after reaching this maximum point, C_L decreases dramatically due to stall. Stall occurs when the airflow over the wing separates, resulting in a rapid loss of lift. The relationship between C_L and α is typically linear at low to moderate angles of attack but becomes nonlinear near $C_{L_{\max}}$.

The C_D consists of two main parts: parasite drag and induced drag. Parasite drag is unrelated to lift and increases with the square of velocity. On the other hand, induced drag is directly related to lift production and increases with the square of C_L. As the angle of attack increases, C_D also increases due to the increase in induced drag and the formation of turbulent airflow.

In the following Figure 6, data from the profile NACA 0012, where the mentioned region C_L decreases can be observed at an angle of attack between 18° and 19°.

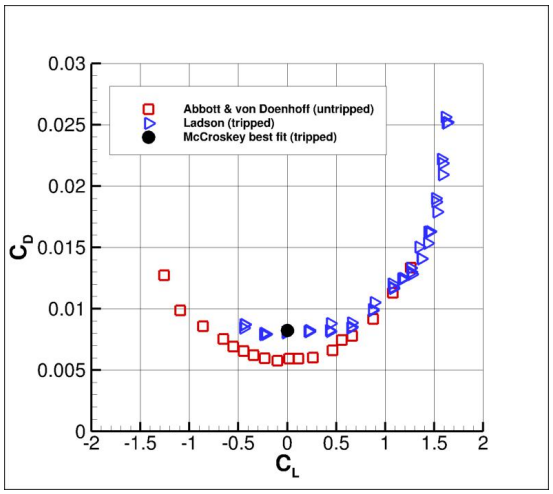

Fig. 6 – Polar of $C_l - C_d$. (NASA, 2021b)

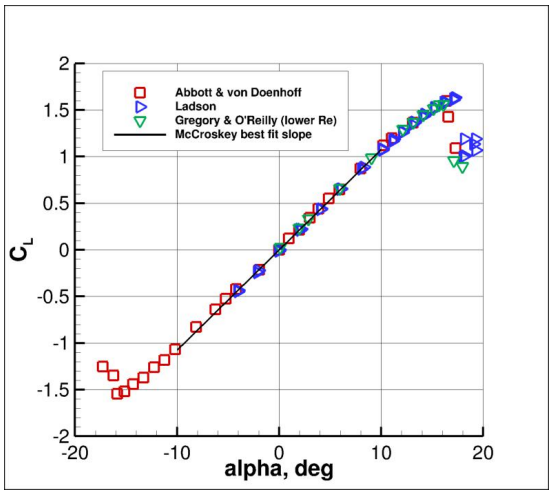

Fig. 7 – Polar of $C_l - \alpha$. (NASA, 2021b)

3.1.3 The NACA Family of Aerodynamic Profiles

The NACA family of airfoils, originating in the 1930s, stands out as a significant contribution to the field of aeronautical engineering. Developed by renowned researchers, this family of aerodynamic profiles was designed to optimize the aerodynamic performance of aircraft (LADSON; JR, 1975).

3.1.3.1 Specification of 4-Digit NACA Airfoil

This series of NACA airfoils is controlled by 4 digits, for example, NACA 2412, which designates the curvature, position of maximum curvature, and thickness. If an airfoil number is NACA *mp*XX, for example, NACA 2412, then (JACOBS; WARD; PINKERTON, 1933):

- m is the maximum curvature divided by 100. In the example, m= 2, so the curvature is 0.02 or 2% of the chord.
- p is the position of the maximum curvature divided by 10. In the example, $p = 4$, the maximum curvature is 0.4 or 40% of the chord.
- XX is the thickness divided by 100. In the example, XX = 12, so the thickness is 0.12 or 12% of the chord.

Calculation of 4-Digit NACA Airfoil

The section of the NACA airfoil is created from a curvature line and a thickness distribution drawn perpendicular to the curvature line.

The equation for the curvature line is divided into sections on each side of the maximum curvature position p. To subsequently calculate the position of the final airfoil envelope, the gradient of the curvature line is also needed. The equations are (AEROSPACEWEB.ORG,):

Calculation of Curvature Line of 4-Digit NACA Airfoil :

$$Y_c = a_0 x^{0.5} + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 \tag{3.7}$$

Calculation of Thickness of Symmetric 4-Digit NACA Airfoil:

$$y_t = 5t\left(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4\right) \tag{3.8}$$

Calculation of Thickness of Asymmetric 4-Digit NACA Airfoil :

$$y_t = 5t\left(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4 + \frac{0.1036}{0.20}(0.5 - x)\right) \tag{3.9}$$

For this curved airfoil, as the thickness needs to be applied perpendicular to the curvature line, the coordinates (x_U, y_U) and (x_L, y_L) respectively of the upper and lower surface of the airfoil become (MARZOCCA,):

$$x_U = x - y_t \sin\theta, \quad y_U = y_c + y_t \cos\theta, \tag{3.10}$$

$$x_L = x + y_t \sin\theta, \quad y_L = y_c - y_t \cos\theta, \tag{3.11}$$

where:

$$\theta = \arctan\left(\frac{dy_c}{dx}\right), \tag{3.12}$$

$$\frac{dy_c}{dx} = \begin{cases} \dfrac{2m}{p^2}(p-x), & 0 \le x \le p, \\ \dfrac{2m}{(1-p)^2}(p-x), & p \le x \le 1. \end{cases} \tag{3.13}$$

The constants m and p are extracted from the first two digits of the NACA designation.

Using the equations above, for a given value of x, it is possible to calculate the position of the curvature line Y_c, the gradient of the curvature line, and the thickness. The position of the upper and lower surfaces can then be calculated perpendicular to the curvature line.

Calculation of Coordinates

The most obvious way to plot the airfoil is to iterate through equally spaced x values, calculating the coordinates of the upper and lower surfaces. Although this works, the points are more spaced out around the main edge, where the curvature is greater. To group the points at the ends of the airfoil sections, a cosine spacing with uniform increments of β is used (TOOLS,).

Cosine spacing can be calculated by the following formula:

$$\Delta x = \frac{1 - \cos(\beta)}{2} \tag{3.14}$$

Where Δx is the interval between the points and β is the angle of inclination of the curve relative to the x-axis. This method provides a more uniform distribution of points along the airfoil, resulting in more accurate representations, especially in areas of higher curvature.

3.2 Important Dimensionless Numbers

The Mach number, M, expresses the relationship between inertial forces, represented by the flow velocity, V, and elastic forces, represented by the local speed of sound in the study

environment, also known as celerity, c_e:

$$M = \frac{V}{c_e} \tag{3.15}$$

As mentioned earlier, the flows can be considered incompressible for a Mach number less than 0.3, where most cases of Mechanical Engineering occur. The flow can be classified according to the Mach number (ANDERSON, 2011; FOX et al., 2018):

1. If $M < 1$, the flow is subsonic;
2. If $M = 1$, the flow is sonic;
3. If $M > 1$, the flow is supersonic.

Two other flow regimes are often defined: transonic flow ($0.8 < M < 1.2$) and hypersonic flow, where $M > 5$.

One more important number for the study of aerodynamic profiles is the Reynolds number, Re, which expresses the relationship between Inertial Forces and Viscous Forces of a fluid:

$$Re = \frac{\rho V L}{\mu} = \frac{V L}{\nu} \tag{3.16}$$

where ρ is the specific mass of the fluid, V is the velocity, L is the length traveled by the flow in the study area, μ is the absolute viscosity, and ν is the kinematic viscosity (μ/ρ).

If the Reynolds number is exceedingly large, viscous effects will be negligible; if it is small, viscous effects will be dominant. The Reynolds number is important for estimating whether friction, i.e., Inertial Forces, will be dominant or not.

For flow in a pipe, the flow regime can be determined according to the Reynolds number (FOX et al., 2018):

- $Re < 2{,}000$, laminar flow;
- $2{,}000 < Re < 4{,}000$, transitional flow;
- $Re > 4{,}000$, turbulent flow.

For flow over a surface, such as a plate, the flow regime can be determined by Re (FOX et al., 2018):

- $Re < 500{,}000$, laminar flow;
- $Re > 500{,}000$, turbulent flow.

Another important parameter in CFD is Y^+, especially in turbulent flows near walls. It measures the proximity of mesh cells to the wall and is calculated as (VERSTEEG; MALALASEKERA, 2007):

$$Y^+ = \frac{\rho \cdot u_\tau \cdot y}{\mu} \tag{3.17}$$

Where:

- ρ is the fluid density;
- u_τ is the wall shear velocity;
- y is the distance from the mesh cell to the wall;
- μ is the dynamic viscosity of the fluid.

Maintaining the value of Y^+ between 30 and 300 is common to ensure adequate resolution of the boundary layer in CFD simulations of flows around solid objects, such as airfoils or ship hulls. Controlling the value of Y+ is essential to ensure reliable results in CFD simulations (VERSTEEG; MALALASEKERA, 2007).

3.3 Computational Fluid Dynamics

CFD is a discipline dedicated to analyzing the behavior of fluids, heat transfer, and associated phenomena through numerical simulations. This approach finds applications in various industrial and non-industrial sectors, such as aircraft aerodynamics, turbines, vehicles, ship hydrodynamics, chemical process engineering, and meteorology, among others (VERSTEEG; MALALASEKERA, 2007). CFD utilizes numerical methods to analyze and solve problems related to fluid flow, which are described by the Navier-Stokes equations. These partial differential equations express the conservation of mass, momentum, and energy. To obtain approximate numerical solutions, discretization methods are employed, approximating the differential equations into systems of algebraic equations that can be solved by computers. These approximations are applied to small domains in space and/or time, allowing detailed and efficient analysis of the studied phenomena.

The classification of flows in CFD includes categories such as laminar, turbulent, viscous, non-viscous, compressible, incompressible, internal, and external (FOX et al., 2018). Laminar flow is characterized by smoothness and stability, where adjacent layers of fluid move in an orderly manner, maintaining their structure. In contrast, turbulent flow exhibits rapid mixing and irregular behavior due to random fluctuations in the three-dimensional velocity field, resulting in chaotic and random behavior even under constant boundary conditions. Viscosity, a fluid's resistance to shear, corresponds to internal friction in fluids due to intermolecular interactions and is generally a function of temperature. Viscous flow takes into account the friction of the

fluid with any solid through which it flows, exhibiting drag, while non-viscous flow is frictionless. Incompressible flow assumes negligible variation in the fluid's specific mass, with gasses considered incompressible if the specific mass variation is less than 5%. The Mach number is an important parameter in identifying compressible or incompressible flow, with flows below a Mach number of 0.3 treated as incompressible. External flow, like the airflow through a wing, occurs in the open air, classifying it as external (AWRUCH; BRAUN; GRECO, 2015).

This detailed classification and simulation of fluid flows are vital for the efficient design and analysis of structures influenced by fluid dynamics, such as aerodynamic profiles impacted by winds (VERSTEEG; MALALASEKERA, 2007).

3.3.1 Navier-Stokes Equations

The Navier-Stokes equations describe the motion of fluid substances and are fundamental in fluid dynamics. They represent the conservation of momentum and are derived from Newton's second law applied to fluid motion, considering the effects of viscosity (ANDERSON, 2011).

The general form of the Navier-Stokes equations is:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i \tag{3.18}$$

where:

- u_i is the component of the fluid velocity vector,
- t is time,
- ρ is the fluid density,
- p is the pressure field,
- ν is the kinematic viscosity, $\nu = \frac{\mu}{\rho}$,
- f_i is the body force per unit mass (e.g., gravity).

The terms in the Navier-Stokes equations can be explained as follows:

- $\frac{\partial u_i}{\partial t}$ represents the local acceleration of the fluid.
- $u_j \frac{\partial u_i}{\partial x_j}$ represents the convective acceleration, which considers changes in velocity due to the movement of fluid particles.
- $-\frac{1}{\rho}\frac{\partial p}{\partial x_i}$ is the pressure gradient force, driving the fluid flow from high to low pressure.

- $\nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}$ is the diffusion term, representing viscous effects and momentum diffusion.
- f_i represents external forces acting on the fluid, such as gravitational force.

For an incompressible fluid, the incompressibility condition is given by:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{3.19}$$

This condition ensures mass conservation for incompressible flows, implying that the divergence of the velocity field is zero.

For a compressible fluid, mass conservation is expressed by the continuity equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0 \tag{3.20}$$

3.3.2 Turbulence Modeling

The primary objective in the study of turbulent flows is to develop a theory or quantitative model capable of calculating values of interest variables with practical relevance (NETO; SOARES, 2014). Various approaches for analyzing turbulent flows can be grouped into three major categories.

The first method is **RANS** (Reynolds-Averaged Navier-Stokes). In this model, turbulence scales are separated between those relative to the mean behavior and those relative to fluctuations around the mean. The model represents transport equations only for quantities relative to the mean behavior, with all turbulence scales being modeled. This approach, by allowing a solution for mean flow variables, significantly reduces computational time. If the mean flow is steady, the governing equations will not contain time derivatives, enabling easier attainment of a steady-state solution. This is the most widely used method for industrial calculations (VERSTEEG; MALALASEKERA, 2007).

The second method is **LES** (Large Eddy Simulation). This is an intermediate model for turbulent calculation that tracks the behavior of larger vortices. The method involves spatial filtering of the unstable Navier-Stokes equations before calculations, removing smaller vortices than the filter size. The effects on the resolved flow, composed of the mean flow plus large vortices, due to the smaller unresolved vortices, are included through a sub-grid scale model. The resolution of unstable flow equations demands significant computational resources in terms of storage and calculation volume. The advantage of LES is that by modeling fewer vortices, the error induced by the turbulence model is reduced (VERSTEEG; MALALASEKERA, 2007).

The third method is **DNS** (Direct Numerical Simulation). These simulations calculate both the mean flow and all turbulent velocity fluctuations. The unstable Navier-Stokes equations are solved on spatial grids fine enough to resolve Kolmogorov length scales, where energy

dissipation occurs, and with time steps small enough to resolve the period of fastest fluctuations. These calculations are highly costly and are not used on an industrial scale (VERSTEEG; MALALASEKERA, 2007).

After this overview of the different turbulence modeling methods, we now delve into the Reynolds-Averaged Navier-Stokes (RANS) equations, which are widely used in simulations due to their computational efficiency.

3.3.3 Reynolds-Averaged Navier-Stokes Equations

RANS models are used in simulations that aim to obtain average values of forces on a solid, considerably reducing computational time, making it the most recommended method when there are limitations due to local resources. These models are mainly based on averaging the different equations using a mean value along with a fluctuation component, $u = U + u'$, $v = V + v'$ and $p_k = P_k + p'_k$, thereby averaging the Navier-Stokes equations (FERZIGER, 2002; POPE, 2000). In the previous expression, p_k is defined as the kinematic pressure, such that $p_k = p/\rho$.

The result of this averaging leads to the following resolution of the equations, written in index form:

$$\frac{\partial U_i}{\partial x_i} = 0 \tag{2.13}$$

$$\frac{\partial U_j U_i}{\partial x_j} = \nu \frac{\partial^2 U_i}{\partial x_j^2} - \frac{\partial u'_i u'_j}{\partial x_j} - \frac{\partial p_k}{\partial x_i} \tag{2.14}$$

where $u'_i u'_j$ is the Reynolds stress tensor, τ_R, and can be defined as:

$$\tau_R = \nu_T \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij}, \tag{2.15}$$

where ν_T, k and δ_{ij} are the turbulent viscosity, turbulent kinetic energy, and the Kronecker delta, respectively. The values of ν_T and k will be calculated differently depending on the model used to determine the turbulence (FERZIGER, 2002; POPE, 2000).

3.3.4 Spalart-Allmaras Turbulence Model (SA)

The turbulence model Spalart-Allmaras, often used for aeronautical applications, therefore, the boundary layer will be resolved directly with $y^+ \approx 1$.

This model performs well in calculating external flows and provides good results for flows with adverse pressure gradients in the boundary layer. It solves a single transport equation for the turbulent kinematic viscosity (equation 2.21) (GONZáLEZ, 2011; Fluent Inc., 2006; openfoam, ; NASA, 2021b).

$$\frac{D}{Dt}(\rho\tilde{\nu}) = \nabla\cdot(\rho D\tilde{\nu}\tilde{\nu}) + C_{b2}\sigma\frac{\tilde{\nu}^2}{|\nabla\tilde{\nu}|^2} + C_{b1}\rho\tilde{S}\left(1-\frac{f_t^2}{2}\right) - \left(\frac{C_{w1}f_w - C_{b1}}{\kappa^2 f_t^2}\right)\frac{\rho\tilde{\nu}^2}{\tilde{\omega}^2\delta^2} + \tilde{S}\tilde{\nu}(2.21) \tag{3.21}$$

From this equation, the modified turbulent viscosity ($\tilde{\nu}$) can be calculated using the following expressions:

$$\tilde{\nu}_t = \nu f_{v1}(2.22) \tag{3.22}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}(2.23) \tag{3.23}$$

$$\chi = \frac{\tilde{\nu}}{\nu}(2.24) \tag{3.24}$$

In equations 2.21, 2.22, 2.23, and 2.24, the coefficients are defined by default as the model does and are defined below (NASA, 2021b):

$$\sigma_\tau = \frac{2}{3}, \quad C_{b1} = 0.1355, \quad C_{b2} = 0.622, \quad \kappa = 0.41,$$

$$C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1+C_{b2}}{\sigma_\tau}, \quad C_{w2} = 0.3, \quad C_{w3} = 2, \quad C_{v1} = 7.1, \quad C_\omega = 0.3.$$

It is worth mentioning that there are other types of RANS turbulence models also very valid for use in this problem, such as k-ε, k-ω, and also the k-ω SST. In the latter, attempts are made to solve the problems that appeared with the other two named models. All of these models make very good predictions of flow separation, but ultimately SA was chosen because it is a turbulence model originally designed for aerodynamic applications.

3.3.5 SST Turbulence Model

The turbulence model k-ω SST (Shear Stress Transport) model, is widely used in computational fluid dynamics (CFD) simulations due to its ability to handle complex flows, including those with boundary layer separation and adverse pressure gradients.

This model combines the advantages of the k-ω and k-ε models, using the k-ω in the near-wall regions and the k-ε in the far-field regions, thus providing greater accuracy in predicting flows with separation. The model solves two transport equations: one for the turbulent kinetic energy (k) and one for the specific dissipation rate (ω) (MENTER, 1994b; NASA, 2021c) The transport equations are defined as:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_i k)}{\partial x_i} = P_k - \beta^* \rho k \omega + \frac{\partial}{\partial x_i}\left[\left(\mu + \sigma_k \frac{\mu_t}{\rho}\right)\frac{\partial k}{\partial x_i}\right] \quad (2.1) \qquad (3.25)$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_i \omega)}{\partial x_i} = \alpha \frac{\omega}{k} P_k - \beta \rho \omega^2 + \frac{\partial}{\partial x_i}\left[\left(\mu + \sigma_\omega \frac{\mu_t}{\rho}\right)\frac{\partial \omega}{\partial x_i}\right] + 2(1 - F_1)\rho \sigma_{\omega 2} \frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i} \quad (2.2) \qquad (3.26)$$

Where μ_t is the turbulent viscosity, P_k is the production of turbulent kinetic energy, and the terms α, β, β^*, σ_k, and σ_ω are model coefficients defined as :

$$\alpha = 0.31, \quad \beta = 0.075, \quad \beta^* = 0.09, \quad \sigma_k = 0.85, \quad \sigma_\omega = 0.5.$$

The SST model also introduces a blending function F_1 that combines the k-ω and k-ε models in the respective regions where each is more effective.

It is important to mention that OpenFOAM implements the k-ω SST model, allowing its application in various case studies, from aerodynamics to industrial flows. This model is particularly effective in predicting flow separation and boundary layer behaviors, offering a balanced combination of accuracy and numerical stability (NASA, 2021c; OPENFOAM,).

4 Methodology

To ensure the effectiveness and ease of use of the developed program, the development process emphasized modularity and clarity. The Python script is structured to guide users through each step of the simulation process, from pre-processing to post-processing, with detailed instructions and prompts. The aim is to automate aerodynamic simulations using the OpenFOAM software, facilitated by a graphical interface built using the Tkinter module. This interface allows users to interact with the program's functionalities, enabling the automatic generation and refinement of mesh grids based on specified parameters.

Furthermore, the tool incorporates robust error-checking and validation features to minimize user input errors and ensure accurate simulations. Leveraging Python's extensive libraries and straightforward syntax, the program is accessible as it works with open-source code, becoming a resource for researchers and engineers in aerodynamics. This approach simplifies workflows and democratizes access to advanced simulation techniques, expanding the potential user base and enhancing opportunities for collaborative research.

4.1 OPENFOAM

OpenFOAM (Open - Field Operation and Manipulation) is an open-source CFD platform, developed in C++, consisting of modifiable libraries covering various numerical models and CFD tools. Utilizing the Finite Volume Method (FVM), OpenFOAM creates separate matrix equations for each equation and solves them through iterative solvers. This method adopts a co-located strategy, where the solution variables for each matrix equation are defined at the cell centers. FVM in OpenFOAM is applied to cells with arbitrary shapes in unstructured grids.

The coupling between the pressure and velocity equations is performed through the Pressure-Implicit Split-Operator (PISO) algorithm for transient simulations and the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm for steady-state simulations. OpenFOAM offers a wide variety of turbulence approaches.

This highly flexible structure of OpenFOAM, based on C++, facilitates the customization of simulations to meet the specific needs of various CFD problems. Figure 8 illustrates the organization of this software-driven analysis (OPENFOAM, 2013).

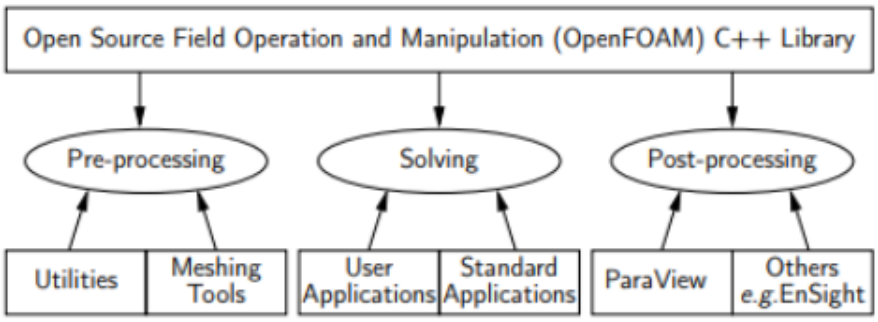

Fig. 8 – OpenFOAM Structure. Source: (OPENFOAM, 2013)

The SIMPLE method is a numerical technique used to solve equations describing fluid flow, especially in CFD problems. Developed to solve the Navier-Stokes equations, which describe fluid flow behavior, the SIMPLE method combines explicit and implicit steps in a semi-implicit manner to ensure an efficient and stable solution.

simpleFoam is a solver for laminar or turbulent incompressible flow for Newtonian fluids. It is based on the finite volume method. rhoSimpleFoam is an extension of simpleFoam that considers fluid compressibility.

4.1.1 Folder `0`

At the genesis of every OpenFOAM simulation lies the `0` folder, housing crucial initial condition files indispensable for solver initialization (OPENFOAM, 2013). Within this folder, each file plays a distinct role in defining the starting state of the simulation:

- **p**: Prescribing the pressure distribution, this file sets the initial pressure behavior within the computational domain.
- **U**: Defining the velocity field and flow direction, this file characterizes the initial flow conditions and interactions.
- **alphat** and **nut**: Essential for turbulence modeling, these files store parameters essential for resolving turbulent flow phenomena.
- **k** and **omega**: Containing turbulence fields vital for turbulence model computations, these files are fundamental for solving kinetic energy transfer equations.

4.1.2 Folder `system`

Central to governing the simulation's behavior are the files residing within the `system` folder (OPENFOAM, 2013). This directory hosts an array of control files orchestrating various aspects of the simulation process:

- **controlDict**: Dictating simulation parameters and convergence criteria, this file governs the temporal evolution and termination of the simulation.
- **fvSchemes** and **fvSolution**: Prescribing numerical schemes and solution methodologies, these files influence the discretization and resolution of governing equations.
- **blockMeshDict**: Crucial for mesh generation, this file delineates the computational domain's discretization, facilitating accurate representation of physical phenomena.
- **decomposeParDict**: Enabling parallel computation, this file defines mesh decomposition strategies, optimizing computational efficiency.
- **Forces**: Offering insights into surface pressures and friction, this file facilitates the analysis of aerodynamic forces and moments on specified components.

4.1.3 Folder `constant`

Persisting throughout the simulation duration are the files encapsulated within the `constant` folder (OPENFOAM, 2013). This repository houses data integral to the simulation's fidelity and consistency:

- `transportProperties` and `turbulenceProperties`: Containing fluid and turbulence model properties, respectively, these files uphold the simulation's physical fidelity by specifying fluid characteristics and turbulence modeling approaches.

In summary, meticulous organization and comprehension of the files and folders within the OpenFOAM framework are imperative for orchestrating successful CFD simulations, ensuring accuracy, efficiency, and fidelity throughout the computational endeavor.

4.2 Python

Python is one of the most popular and widely used programming languages in the world. The language is known for its simple and readable approach, making it ideal for both beginners and experienced programmers. Python's clear syntax allows programmers to write code efficiently and quickly, while its wide range of libraries supports a variety of applications. Additionally, Python is an interpreted language, which means the code can be executed immediately after being written, without the need for compilation. This makes it a versatile and flexible language for a wide range of projects and applications. With its active community and continuous support, Python is an excellent choice for those seeking a powerful and intuitive language for their programming needs (LUTZ, 2013; ROSSUM; JR., 2009; SUNDNES, 2020).

The software development was carried out using various libraries and tools, to provide a user-friendly graphical interface, efficient file handling, and execution of complex calculations.

The project utilized the `tkinter`, `tqdm`, `os`, `subprocess`, `functions`, `shutil`, `numpy`, `math`, `pandas`, `matplotlib.pyplot`, and `openpyxl` libraries.

The graphical user interface was developed using `tkinter`, which offers a simple and effective way to create windows, buttons, text boxes, and other interactive components. The dialog functions (`messagebox` and `filedialog`) are used for essential interactions such as opening and saving files and presenting messages to the user. The `tqdm` library was used to display progress bars in the console, providing visual feedback on the execution of loops and long processes (LUNDH, 1999; SUNDNES, 2020).

For file handling, the `os` library was employed for file system navigation, directory creation and removal, and other basic operations. The `shutil` library facilitated more complex operations, such as copying files and entire directories, providing an additional layer of abstraction and ease of use. The execution of external commands and scripts was facilitated by the `subprocess` library, which allows shell commands to be executed directly from Python code, facilitating automation and integration with other tools and scripts (MARTELLI, 2005; DOWNEY, 2015).

The core of the software's numerical processing was supported by `numpy` and `math`, which together provided a robust set of tools for mathematical calculations. Data manipulation and analysis were carried out with `pandas`, while the results were visualized using `matplotlib.pyplot`. The `openpyxl` library was used for reading and writing Excel files, ensuring that data could be easily imported and exported in a widely used format (MCKINNEY, 2017; HUNTER, 2007).

The `functions` library contains specific functions developed for the project, promoting code modularization and facilitating the maintenance and extension of the software's functionalities.

The software workflow follows these steps: initialization, where the software loads the graphical interface, allowing the user to interact with the system; user interaction, where the user can select files, perform calculations, and view results through the GUI; processing, where data is processed using `numpy` and `pandas`, with visualizations generated by `matplotlib`; automation, where necessary external commands are executed via `subprocess`; and results export, where results can be exported to Excel files using `openpyxl`.

4.3 Program

With the study of the OpenFOAM structure and utilizing Python, a methodology was developed to create a code aimed at interacting with OpenFOAM simulation documents. The presented flowchart, Fig: 9, shows the step-by-step process adopted to simplify and automate the entire simulation process, from mesh generation to result analysis.

The process begins with creating the airfoil points and the desired angles of attack. Afterwards, the program provides a validated standard mesh, which is especially useful for users who are not very familiar with OpenFOAM and the mesh creation process, offering a solid foundation for initial simulations. However, since the standard mesh creation process may not meet more complex and specific cases, the user can alter parameters that will be specified later, allowing for greater flexibility and adaptation to the specific needs of the project.

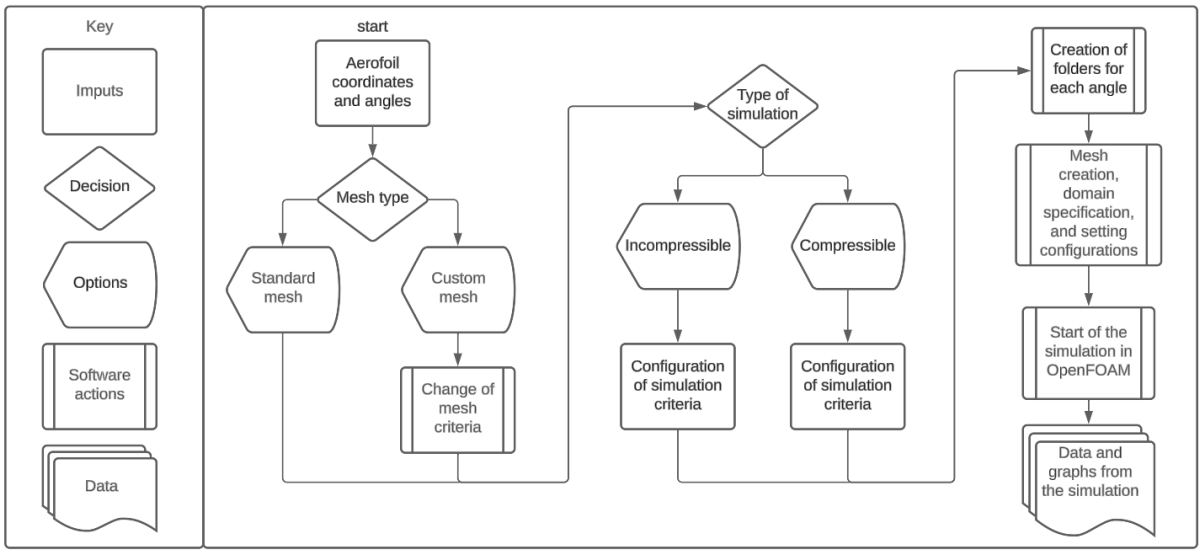

Fig. 9 – Flowchart of the program.

After defining the mesh type, the next step is to choose the type of simulation. Depending on the flow characteristics, the simulation type can be either incompressible or compressible. For flows where the fluid density does not vary significantly, an incompressible simulation is configured. For flows with significant variations in fluid density, a compressible simulation is configured.

The following step involves creating specific folders to store the simulation data for each angle of attack and setting up the cases in OpenFOAM's standard format with the mesh, boundary conditions, and all necessary configurations defined. With all configurations ready, the simulation is automatically started in OpenFOAM.

After the simulation execution, the data will be decentralized. The program will then aggregate the data for the drag coefficient and lift coefficient, creating images and a database for analysis. Some more specific data, such as the Y+ value, will remain in the folder for each angle, along with information to generate velocity and pressure fields that can be visualized using the ParaView program. Equipped with a graphical interface, the program efficiently manages each step, allowing easy configuration of boundary conditions, simulation initiation, and analysis of the resulting data.

4.3.1 Domain

The numerical mesh used in this study consists of hexahedral elements and was generated as a "body-fitted"structured mesh. This approach allows for a precise representation of the problem's geometry, with the mesh elements conforming to the body's surface. The mesh is divided into various regions that represent the region of interest in detail, providing adequate resolution for flow analysis.

Among various types of meshes, the C-mesh stands out for its flexibility and adaptability to complex geometries, characteristics that make it particularly useful in flow simulations around aerodynamic profiles. This typology is based on orthogonal lines that intersect at right angles, allowing the creation of prism cells that conform to the profile surface. The American space agency, NASA, applies the C-mesh, as demonstrated in the study of the NACA0012 airfoil. The C-mesh offers significant advantages in this context, providing an accurate representation of the profile geometry, including details such as curvature and boundary layer (COSTA, 2019; NACA,).

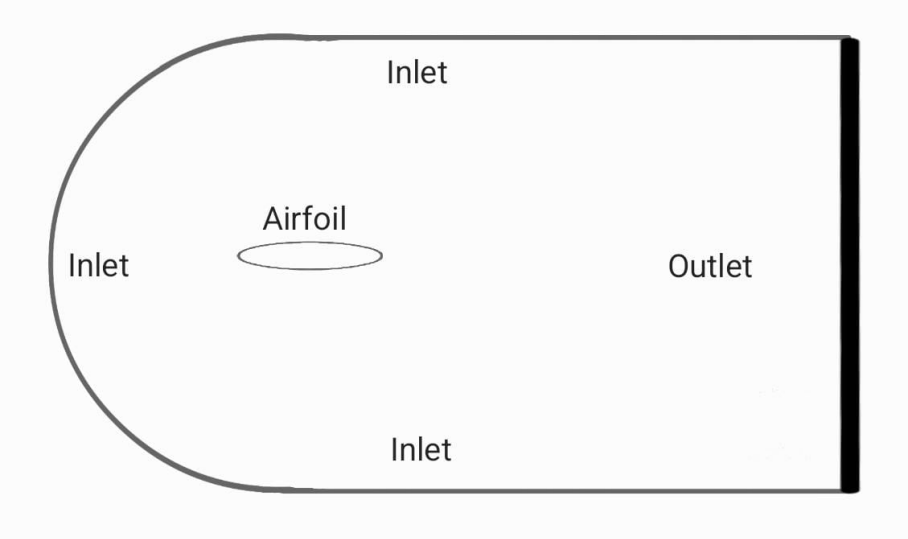

Fig. 10 – Domain C-mesh, out of scale.

The C-meshes are widely used in airfoil simulations as they contribute to the numerical stability of CFD solvers by minimizing numerical errors, ensuring consistent and reliable results throughout the simulation process (CHUNG, 2000). Additionally, these meshes aid in reducing numerical diffusion, preserving the physical characteristics of the flow field. This preservation leads to more precise and realistic simulation results (HIRSCH, 2007).

4.3.2 Validation

To ensure the accuracy of the results, a convergence analysis of the C_L as a function of the number of mesh cells was performed. The convergence study is essential to verify the stability and consistency of the results as the mesh is refined. A reference data from NASA's

validation case was considered. The Reynolds number is approximately $\mathrm{Re} \approx 6 \times 10^6$, where $V_\infty = 51.4815\,\mathrm{m/s}$, c is the local chord length of the airfoil (1.0 m), and ν_{fluid} is the kinematic viscosity of the fluid ($8.58 \times 10^{-6}\,\mathrm{m^2/s}$). Additionally, the free stream velocity in the flow direction is $V_\infty = 51.4815\,\mathrm{m/s}$, and the (Ma) is approximately 0.15. The turbulence model used is the Spalart-Allmaras (OPENFOAM,).

Figure 11 shows of C_L as a function of the number of cells for a simulation with 10 degrees of angle of attack. It is observed that the value of C_L approaches the reference value as the number of cells increases, indicating the convergence of the results.

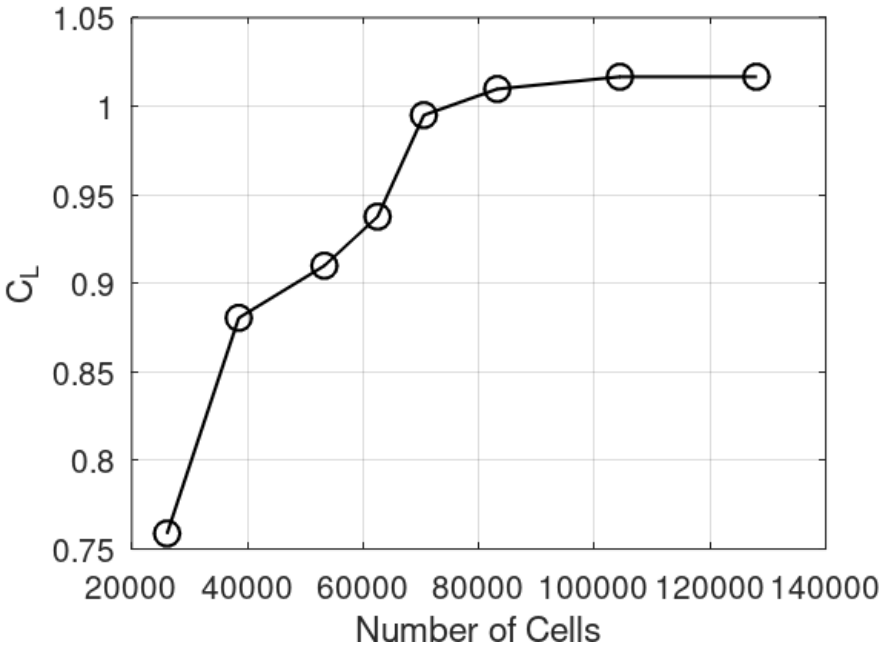

Fig. 11 – C_L as a function of the number of cells.

Figure 11 indicates that, after a certain number of cells, the value of the C_L stabilizes. This behavior is expected, as a more refined mesh provides a better resolution of the physical phenomena, leading to a more accurate solution.

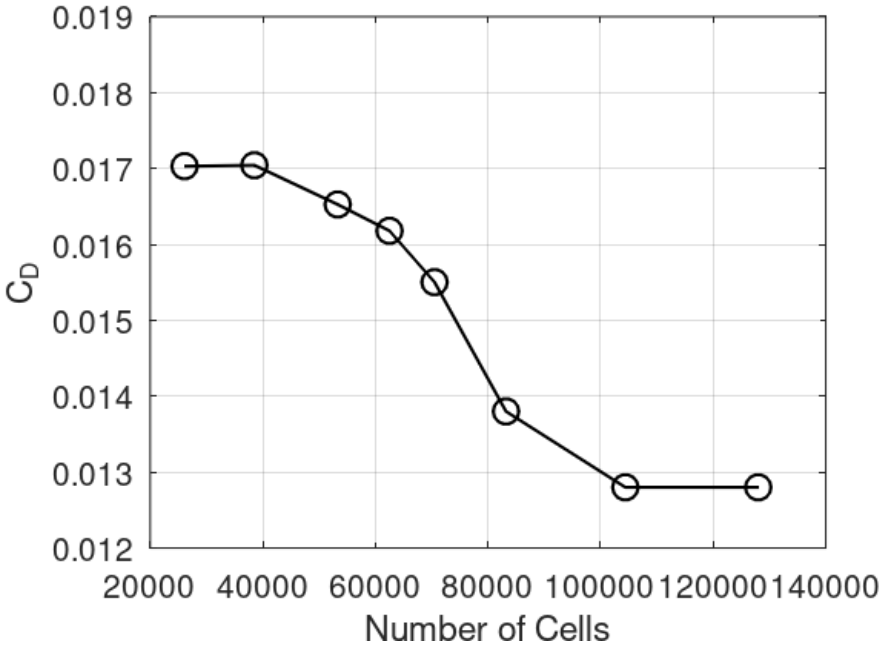

Fig. 12 – C_D as a function of the number of cells.

Figure 14 shows that the drag coefficient (C_D) also stabilizes as the number of cells increases. Similar to the lift coefficient, the stabilization of the drag coefficient indicates that the solution is converging, ensuring the accuracy and reliability of the numerical predictions.

The resulting mesh from this study has a distance from the profile to the inlet and outlet of 20 times the chord length. Along the profile, 300 divisions were made, with a variable density, starting with 100 divisions at the leading edge up to 25% of the chord. The boundary layer thickness is 2 millimeters, and the thickness of the first layer is 20 nanometers. The domain has the same number of divisions inside and outside the boundary layer, 100 divisions, to ensure the highest density near the profile. After the profile, 200 divisions were made within the boundary layer, with an expansion factor of 1.01.

The values of the variables adopted for the mesh, which are in the program, are: The cell size at the leading edge is 1×10^{-8}, at the trailing edge is 2×10^{-8}, and in the middle is 1.5×10^{-8}. The separating point position is 2.5×10^{-1} from the leading edge.

The boundary layer thickness is 2×10^{-1} and the thickness of the first layer is 2×10^{-10}. The expansion ratio used is 1.01. The maximum cell size at the inlet is 1×10^{-6}, at the outlet is 4×10^{-6}, and between the inlet and outlet is 1×10^{-5}.

The mesh in the boundary layer near the profile has 1×10^2 cells, in the layer furthest from the profile has 1×10^2 cells, in the tail has 2×10^2 cells, at the leading edge has 1×10^2 cells, and at the trailing edge has 2×10^2 cells. The expansion ratio at the inlet is 2.5×10^{-1}.

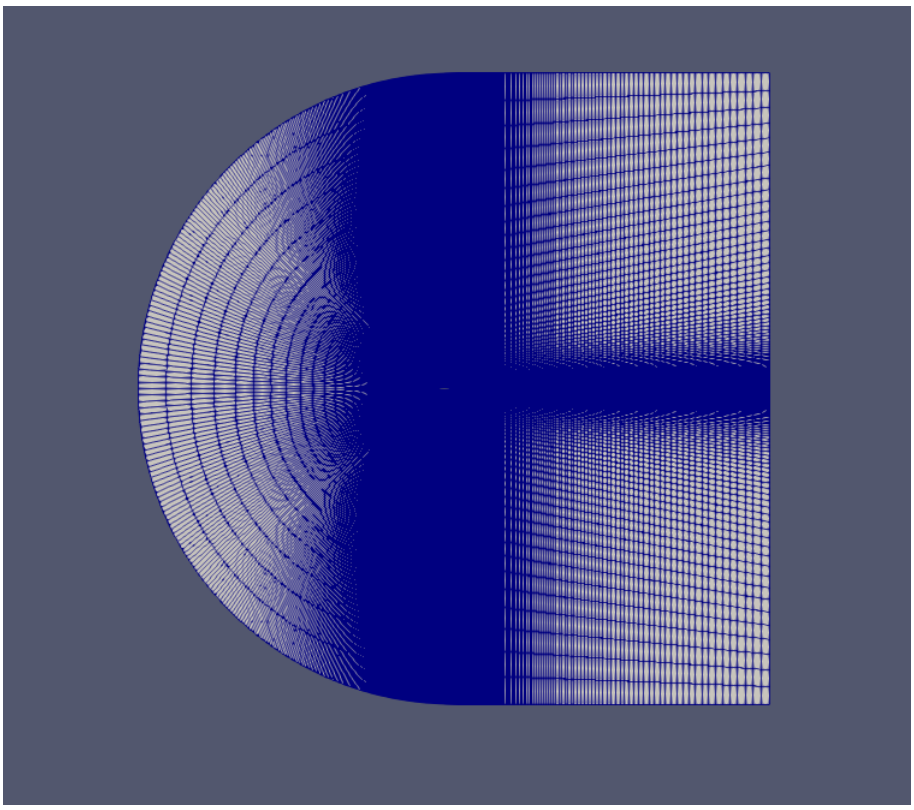

Fig. 13 – Mesh generated by the program.

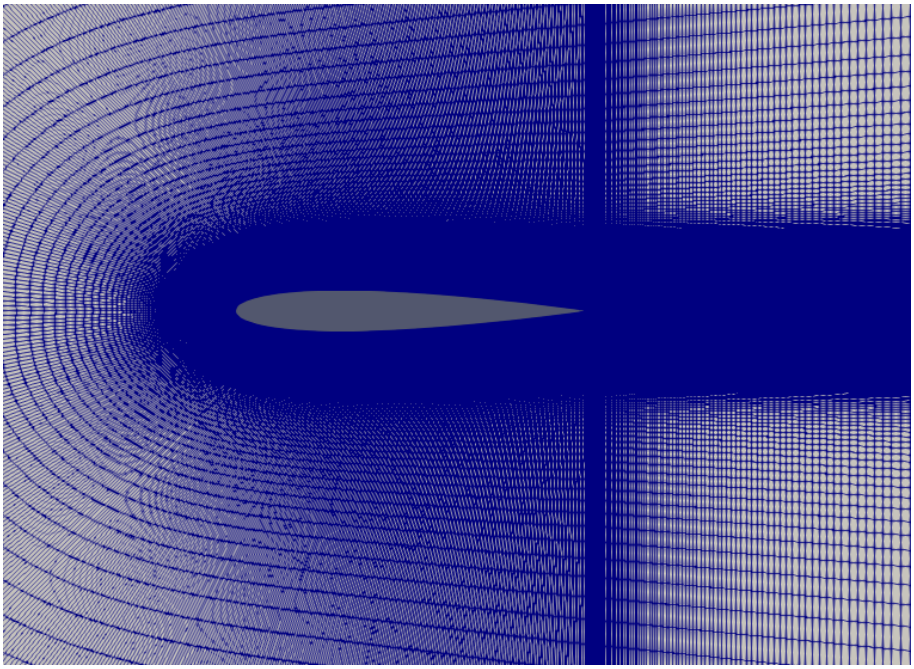

Fig. 14 – Zoom of the mesh in the profile region.

After convergence, to complete the mesh validation done by the software, a comparison was made with the data provided by the 2D NACA 0012 Airfoil Validation Case, comparing with the results presented below NASA (2021a):

The Mean Absolute Percentage Error (MAPE) is a common measure of model accuracy relative to actual values. It is calculated as the average of the absolute error proportions relative to actual values, expressed as a percentage:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{4.1}$$

Where:

- n is the total number of observations.
- y_i are the actual values.
- $\hat{y}_i$ are the values predicted by the method or model.

Tab. 1 – Comparison of C_L and MAPE for different methods

Method	C_L (0°)	C_L (10°)	C_L (15°)	MAPE (0°)	MAPE (10°)	MAPE (15°)
CFL3D	approx 0	1.0909	1.5461	-	0.89%	0.39%
FUN3D	approx 0	1.0983	1.5547	-	0.87%	0.37%
NTS	approx 0	1.0891	1.5461	-	0.88%	0.38%
JOE	approx 0	1.0918	1.5490	-	0.89%	0.36%
SUMB	approx 0	1.0904	1.5446	-	0.89%	0.36%
TURNS	approx 0	1.1000	1.5642	-	0.88%	0.37%
GGNS	approx 0	1.0941	1.5576	-	0.89%	0.36%
Results	approx 0	1.0097	1.5616	-	-	-

Tab. 2 – Comparison of C_D and MAPE for different methods

Method	C_D (0°)	C_D (10°)	C_D (15°)	MAPE (0°)	MAPE (10°)	MAPE (15°)
CFL3D	0.00819	0.01231	0.02124	1.12%	4.73%	1.35%
FUN3D	0.00812	0.01242	0.02159	1.07%	3.28%	0.85%
NTS	0.00813	0.01243	0.02105	1.17%	3.22%	1.35%
JOE	0.00812	0.01245	0.02148	1.11%	3.83%	1.02%
SUMB	0.00813	0.01233	0.02141	1.20%	3.27%	1.22%
TURNS	0.00830	0.01230	0.02140	1.17%	3.11%	1.36%
GGNS	0.00817	0.01225	0.02073	1.18%	3.55%	1.36%
Results	0.00826	0.01280	0.02117	-	-	-

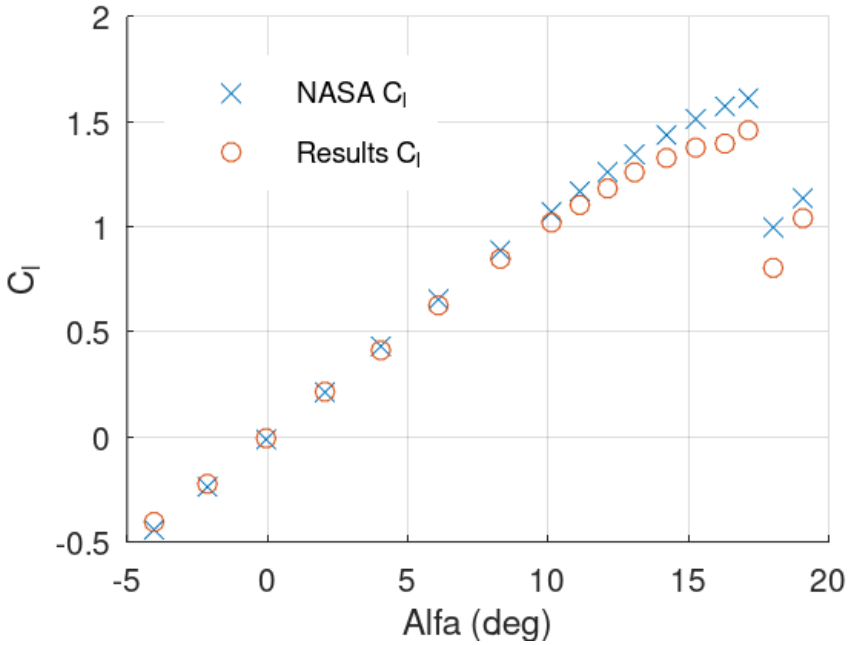

Fig. 15 – Comparison of values with data (NASA, 2021a)

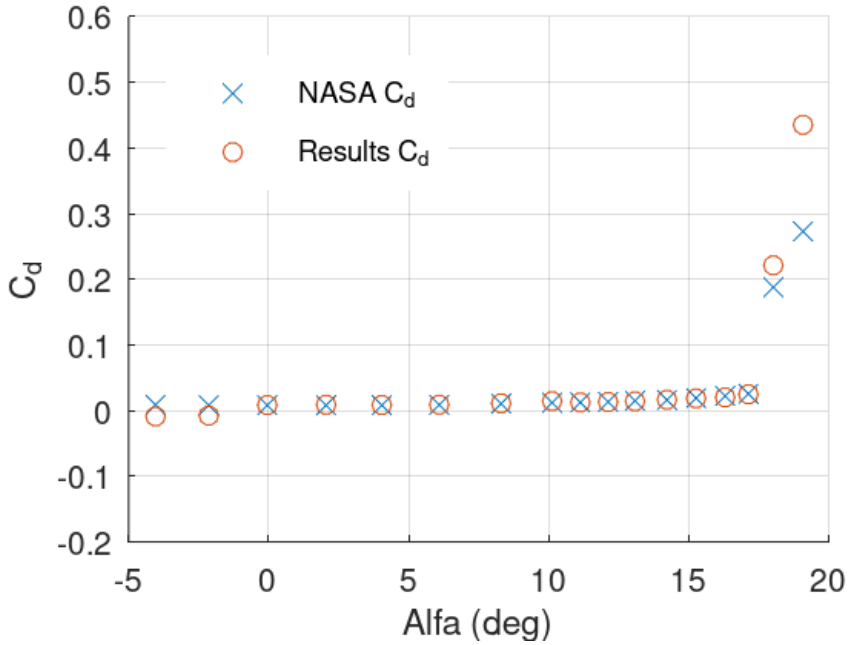

Fig. 16 – Comparison of values with data (NASA, 2021a)

The results of mesh validation, comparing the lift and drag coefficients obtained by different methods to the reference results, provide a view of the accuracy and performance of the analyzed models. Initially, observing the lift coefficient values, there is a general consistency among the methods, with minimal variations in results for different angles of attack. However, when examining the drag coefficient values, a slightly more significant variation among the methods is noted, especially at higher angles of attack.

The comparison of results also reveals that most methods present values very close to Results, indicating good agreement with the reference data. However, the calculation of (MAPE) reveals small differences between the methods and the reference values, especially for the drag coefficient at higher angles of attack. These differences, although small, can be significant in practical applications that require extremely high precision.

For the simulations, RANS equations in the steady-state regime were used with the simpleFoam solver and the SA turbulence model. During the simulations, the residuals remained below 10^{-6}.

The residual control was configured in the `FvSolution` section of OpenFOAM, for a maximum of 2000 iterations. The stopping criterion was set to interrupt the simulations when the residuals reached the specified values above or when the maximum number of iterations was reached.

The software is programmed to perform simulations of NACA airfoils with 4 digits, allowing the choice of simulations for a single angle of attack or obtaining polar curves, which are graphical representations of the variations of lift and drag coefficients as a function of angle of attack.

5 Results

The application of the theory regarding fluid dynamics and the operation of OpenFOAM, including mesh creation, definition of boundary conditions, initialization of the simulation, and presentation of results, was complemented by the development of a Python program. This program interacts with OpenFOAM, automating the creation of the mesh for NACA four-digit profiles or from a document of points provided by the user, as well as the definition of boundary conditions.

The following screens represent the result of this work, demonstrating the operation of the software after its validation. They illustrate the potential of the software to integrate with OpenFOAM, perform simulations, and extract relevant information for various applications.

5.1 Aerodynamic Simulation Automation Program

Upon starting the program, the initial screen, in Fig 17, will be displayed, where the user can define the study profile and the desired angles. The program allows the user to use the software's default mesh, which has been validated by comparison with data from studies of the NACA 0012 profile. However, for compressible cases, it is recommended to use a customized mesh, which enables the manipulation of the criteria that define the mesh

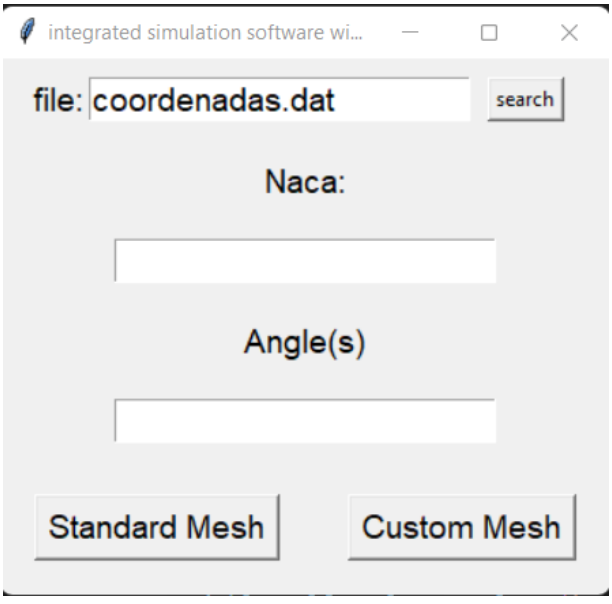

Fig. 17 – initial screen.

– **Option 1:** Selection of a Local File:

  The user can provide a file containing the coordinates of the aerodynamic profile points. These data should be entered in the first field, following the .dat file format as illustrated in Figure 17).

– **Option 2:** Coordinate definition:

  If the user prefers, simply enter the desired 4-digit NACA family profile name in the second field of the initial screen, Figure 17. For example, "0012". In this case, the program will generate the points according to the formula governing the profile domain based on the name values.

The software used offers extensive flexibility in modifying mesh parameters. Users can adjust the following criteria to optimize the mesh according to the specific needs of each simulation (see Figure 18):

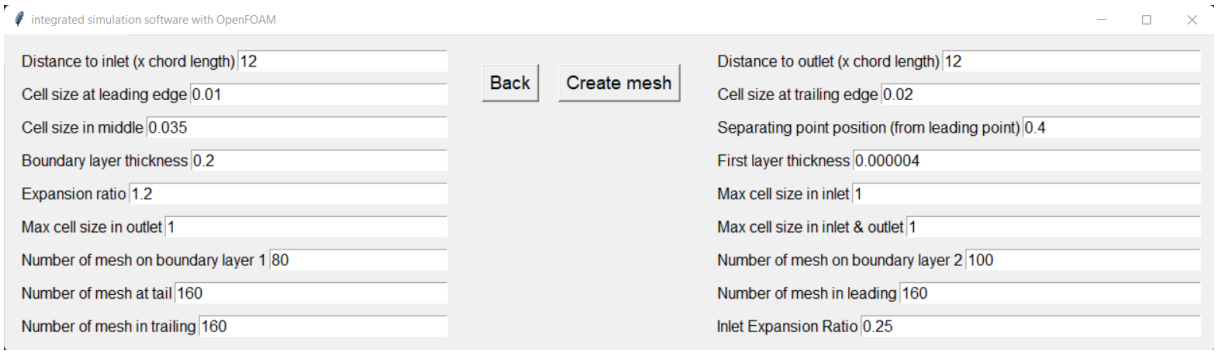

Fig. 18 – Custom mesh screen.

– Distances relative to the inlet and outlet points; (Fig.19)

– Sizes of cells at the leading edge, trailing edge, and middle; (Fig.20)

– Position of the separation point; (Fig.21)

– Thicknesses of layers at the boundary and the first layer; (Fig.21)

– Maximum cell sizes at the inlets, outlets, and the junction between inlet and outlet;(Fig.22)

– Number of divisions in and out the boundary layer, at the tail, at the leading edge, and the trailing edge. (Fig.23)

The tail angle is adjusted according to the angle of attack to better capture the effects downstream of the aerodynamic profile (Fig.24).

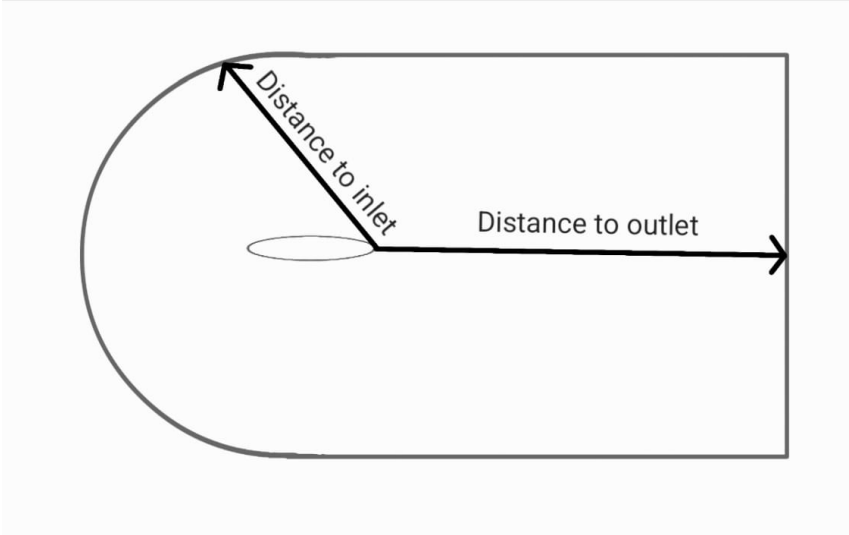

Fig. 19 – Distances to the inlet and outlet, out of scale.

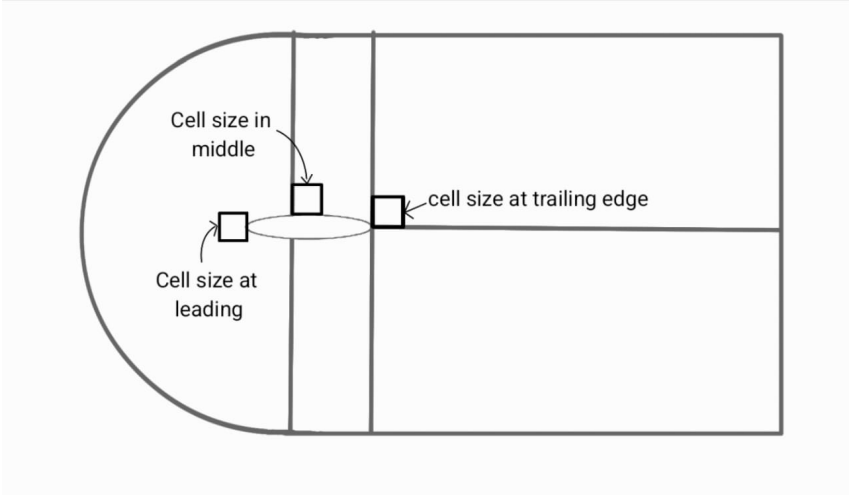

Fig. 20 – size of cells around the profile, out of scale.

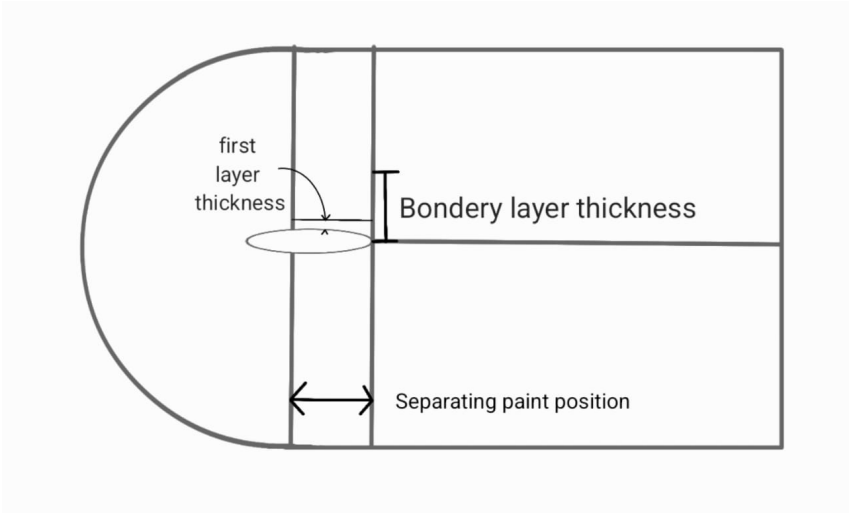

Fig. 21 – Position of the separation point, thicknesses of layers at the boundary and the first layer, out of scale.

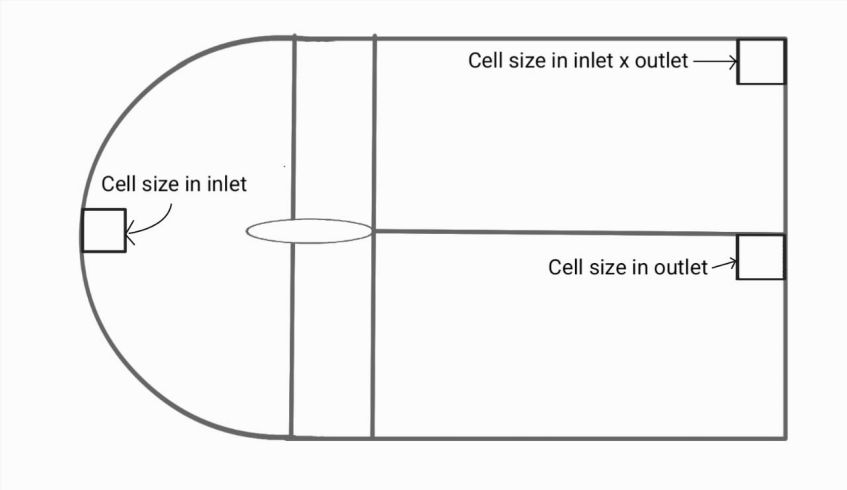

Fig. 22 – cell sizes on contours, out of scale.

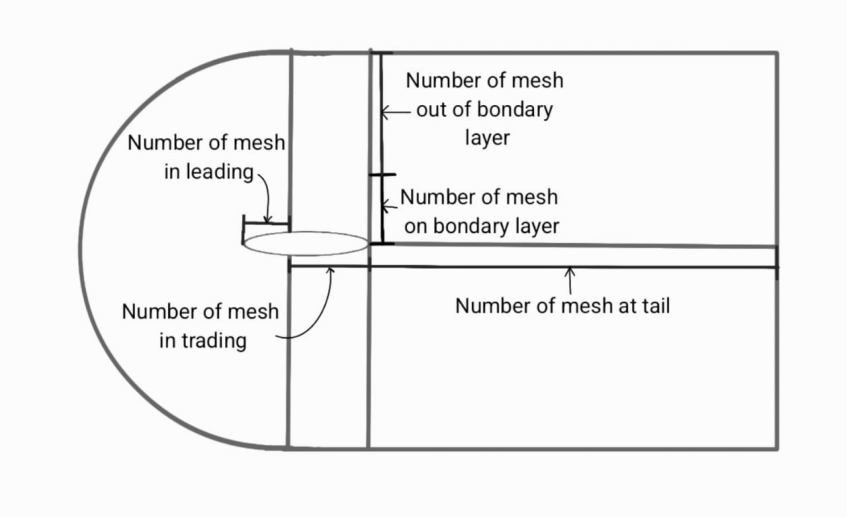

Fig. 23 – Number of divisions,out of scale.

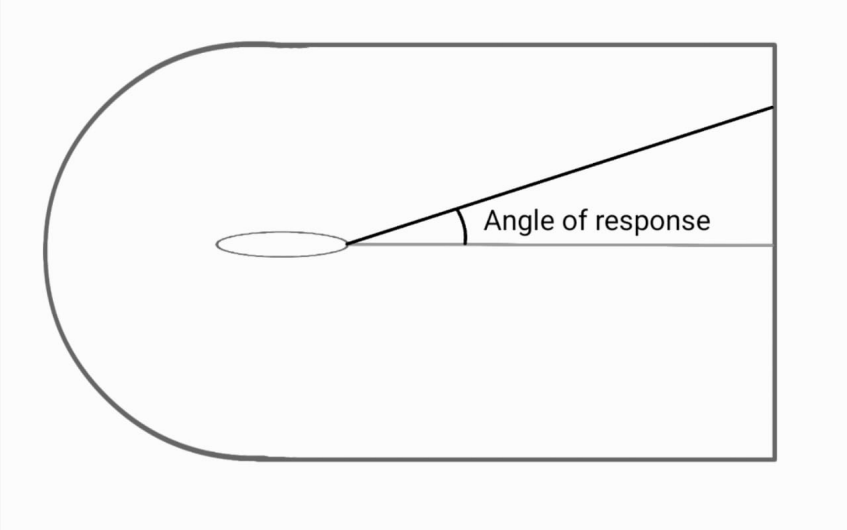

Fig. 24 – Tail angle, out of scale.

After defining the mesh, the user selects the type of simulation for their CFD study. Figure 25 illustrates the interface screen for this selection, where users can choose between compressible and incompressible simulations.

Compressible Simulation is ideal for flows with significant density variations, such as those encountered in high-speed scenarios.

Incompressible Simulation is suitable when density variations in the flow are negligible, commonly observed in low-speed situations.

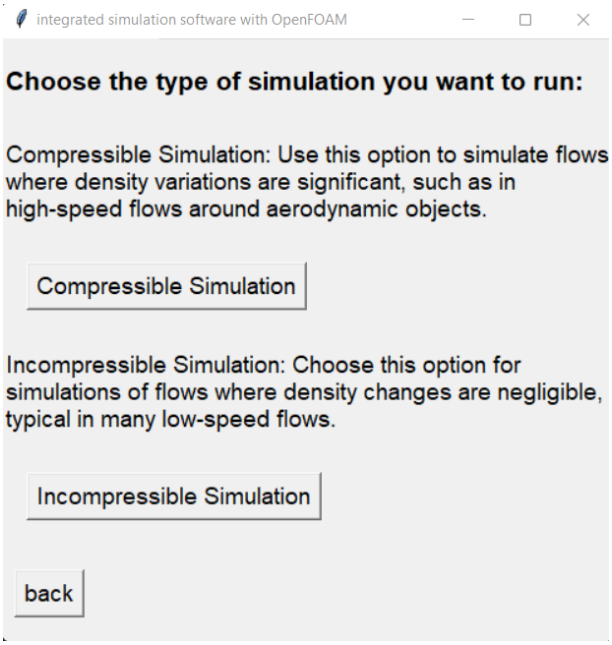

Fig. 25 – Simulation Type.

Choose the simulation type that best suits the characteristics of your study and click on the corresponding button to proceed with the simulation variables configuration.

For a more accurate simulation, the program initially configures the variables automatically according to the standards established in the OpenFOAM tutorial cases. However, it is essential to emphasize that the program provides the necessary flexibility for the user to adjust these parameters according to the specific requirements of their own study. These parameters play fundamental roles in ensuring accuracy and proper representation of fluid behavior during CFD simulations.

Among these parameters, particular attention is given to the kinematic viscosity (ν), which is especially relevant in incompressible simulations using SimpleFoam (Figure 26). In these simulations, SimpleFoam utilizes the Spalart-Allmaras turbulence model. In the case of compressible simulations using RhoSimpleFoam, variables such as alphat, nut, k, and, omega can be adjusted in addition to kinematic viscosity (Figure 27). The compressible solver RhoSimpleFoam employs the SST turbulence model for these simulations. The ability to adapt these variables according to the specific characteristics of the problem under study is crucial for ob-

taining precise and meaningful results across a wide range of simulation contexts.

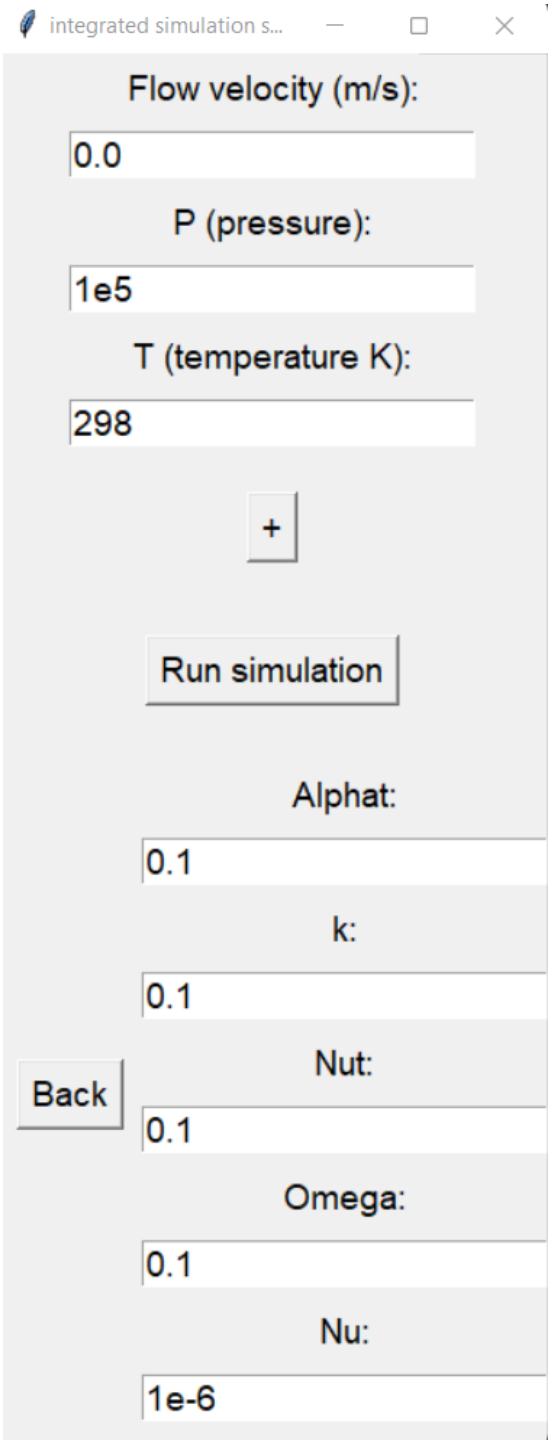

Fig. 26 – Compressible Simulation variables.

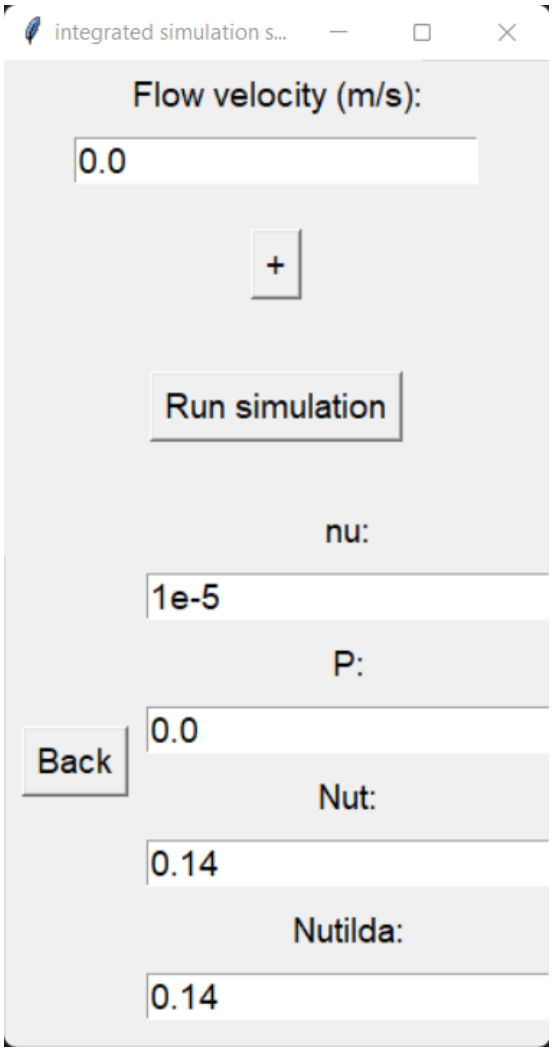

Fig. 27 – Incompressible Simulation variables.

Based on the provided parameters, the program creates a separate OpenFOAM folder for each requested angle. Each folder is individually configured for the simulation, with specific conditions and parameters for that angle of attack.

The simulation is then executed for each angle individually and sequentially using OpenFOAM until the results converge. This means that the program will run each simulation until the results are stabilized and within the defined convergence criteria.

Upon completion of all simulations, the program aggregates the results and generates figures depicting the values of C_L and C_D as functions of the angle of attack. These figures provide a comprehensive overview of how variations in the angle of attack influence the flow characteristics, specifically the lift and drag forces. In Figure 28, you can observe the output files generated after performing the polar curve simulation for an airfoil. The program provides information on the force coefficients as a function of the angle of attack. Additionally, there are two data files, one in .txt format and the other in .xlsx format, which contain the compiled data for manipulation.

Furthermore, each folder containing the simulation results is saved to allow visualization of the case in ParaView and for use in other more specific analyses or validations that may be necessary. This enables detailed analysis of the results and allows the user to explore different aspects of the problem.

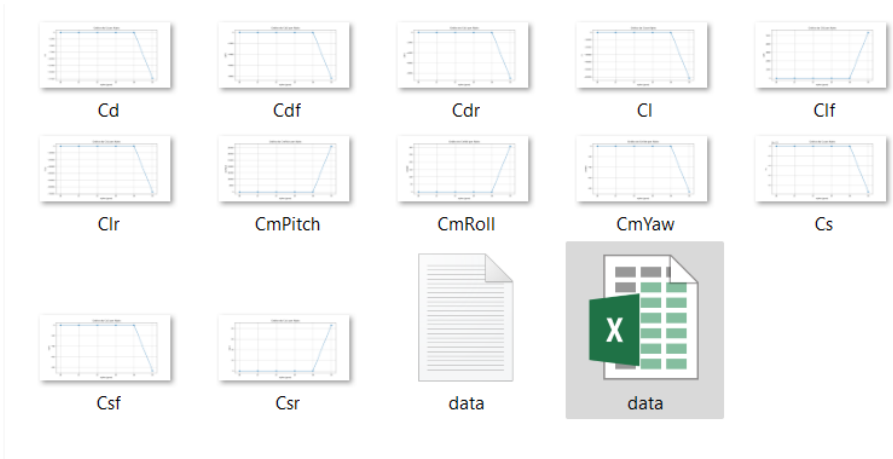

Fig. 28 – Folder with the results

The project material can be found on GitHub. The repository for the program called Aerodynamic_Simulation_Automation_Program acts as an interface with the OpenFOAM simulation. Additionally, the repository includes the necessary OpenFOAM case for the simulation, along with a user manual in Appendix B, an installation manual (README) in Appendix C, and the license description in Appendix D. The links to these resources are available in Appendix A.

5.2 Exploring Different Functionalities of the Program

With the software validated, additional simulations were conducted by modifying parameters to explore and verify the full functionality of the code. This process aimed to ensure that the software performs accurately under different conditions and scenarios, providing a comprehensive evaluation of its capabilities.

NACA 0018

Using the aerodynamic simulation software, simulations were conducted employing the NACA 0018 airfoil profile with an inlet velocity of 4.38 m/s, corresponding to a Reynolds number of $Re = 300{,}000$:

The results obtained were then compared with data available in the literature, including the study by Timmer (TIMMER, 2008) and the book by Anderson Jr. (ANDERSON, 2011). The tables below show the comparison of lift (C_L) and drag (C_D) coefficients between the simulation results and the referenced data:

Tab. 3 – Comparison of Lift and Drag Coefficients with Percentage Errors

Source	C_L	**Error C_L (%)**	C_D	**Error C_D (%)**
Results	0.86	-	0.029	-
Jacob	0.85	1.18	0.028	3.57
Timmer	0.93	7.53	0.027	7.41

The results reveal a satisfactory agreement with Timmer's data, although with a slight discrepancy, especially in the drag coefficient. The comparison with Jacob's data also shows good agreement in the values of C_L and C_D.

NACA 2412

A simulation of the NACA 2412 airfoil profile was conducted with a velocity of 30 m/s and a Reynolds number of approximately 2×10^5. The simulation results were compared with the reference values from the study "Study on effect of semi-circular dimple on aerodynamic characteristics of NACA 2412 airfoil"(SOWMYASHREE et al., 2020).

Table 4 and 4 present the comparison of the lift (C_L) and drag (C_D) coefficients obtained in the simulation with the reference values for the unmodified NACA 2412:

The simulation results show good agreement with the reference experimental data. The small discrepancies observed can be attributed to factors such as numerical modeling, discretization mesh, and turbulence models used. Overall, the simulation proved effective and can be used for further analyses.

Tab. 4 – Comparison of Lift Coefficients (C_L) between Reference Values and Results

Angle of Attack (°)	**C_L Reference Values**	**C_L Results**	**% Error C_L**
0	0.2183	0.1904	-12.77
2	0.4104	0.3984	-2.91
4	0.5895	0.5990	1.61
6	0.8100	0.8033	-0.83
8	0.9873	1.0011	1.40
10	1.1304	1.1882	5.11
12	1.2751	1.3466	5.61

Tab. 5 – Comparison of Drag Coefficients (C_D) between Reference Values and Results

Angle of Attack (°)	**C_D Reference Values**	**C_D Results**	**% Error C_D**
0	0.018	0.0175	-3.02
2	0.0226	0.0271	19.91
4	0.03545	0.0347	-2.25
6	0.048	0.0493	2.76
8	0.0705	0.0707	0.30
10	0.099	0.0992	0.17
12	0.11665	0.1094	-6.23

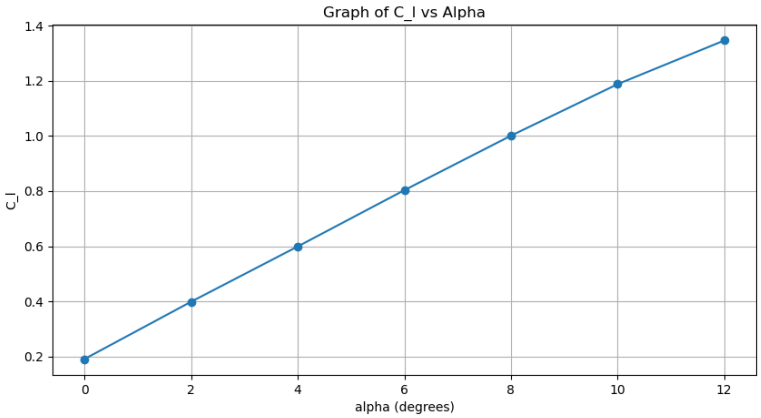

Fig. 29 – C_L generated by the NACA 2412 program

Compressible NACA 4415

Using the aerodynamic simulation software, compressible simulations were conducted employing the NACA 4415 airfoil profile, with a flow velocity of 50 m/s and a Reynolds number of $Re = 2.794 \times 10^6$. It was assumed that the air fluid had characteristics of $\rho = 1.225\,\text{kg/m}^3$ and $\mu = 1.7894 \times 10^{-5}\,\text{kg/(m·s)}$ at 20ºC.

A custom mesh was used for this purpose, introducing alterations to the mesh while maintaining the specified inlet and outlet distances. In comparison with the standard mesh, adjustments were made across various aspects. At the leading edge, the cell size was halved, while at the trailing edge, it remained unchanged. In the middle, there was a reduction in cell size by

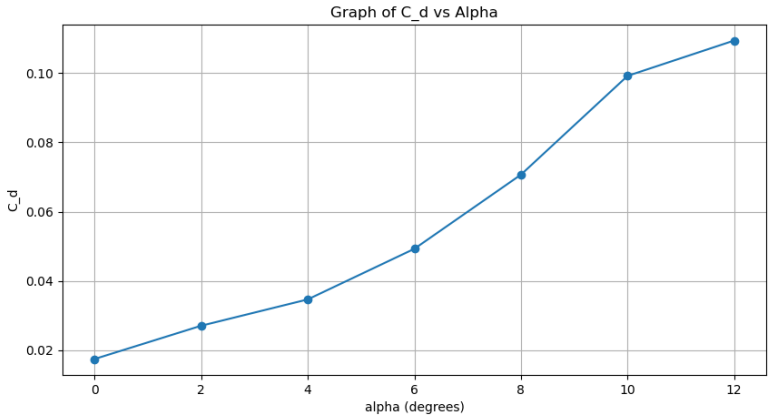

Fig. 30 – C_D generated by the NACA 2412 program

half. The boundary layer thickness was kept constant, but the thickness of the first layer was reduced twice. Additionally, the expansion ratio was slightly increased once. Significant changes were also observed in the maximum cell sizes at the inlet and outlet, with a halving of the latter. Regarding cell distribution, a denser arrangement was adopted near the profile in the boundary layer, twice. These modifications were undertaken to improve resolution in critical areas, such as the leading and trailing edges.

The results obtained were then compared with data available in the literature, specifically in the study "Numerical analysis of an aerodynamic profile model Naca 4415"(SANTOS; CARVALHO; SILVA, 2024). The tables below show the comparison of lift (C_L) and drag (C_D) coefficients between the simulation results and the reference data.

Tab. 6 – Comparison of Lift Coefficients (C_L) with Percentage Errors

Angle of Attack (º)	C_L **(Simulation)**	C_L **(Reference)**	**Error** C_L **(%)**
0	0.375	0.415	-9.64
3	0.699	0.6715	4.09
6	1.018	1.057	-3.69
9	1.299	1.374	-5.46
12	1.516	1.672	-9.33

Tab. 7 – Comparison of Drag Coefficients (C_D) with Percentage Errors

Angle of Attack (º)	C_D **(Simulation)**	C_D **(Reference)**	**Error** C_D **(%)**
0	0.0103	0.0118	-12.71
3	0.0174	0.0161	8.07
6	0.0179	0.0185	-3.24
9	0.0239	0.0252	-5.16
12	0.0342	0.0375	-8.80

The results reveal good agreement with the data from the reference study, although there are some discrepancies, especially in the drag coefficients. The comparison shows that the simu-

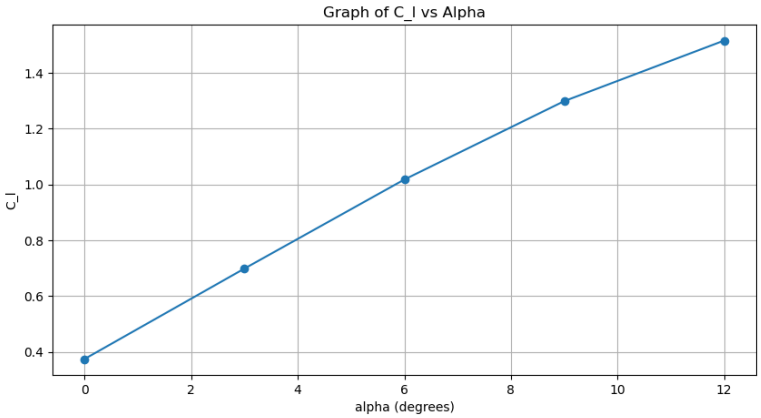

Fig. 31 – C_L generated by the NACA 4415 program

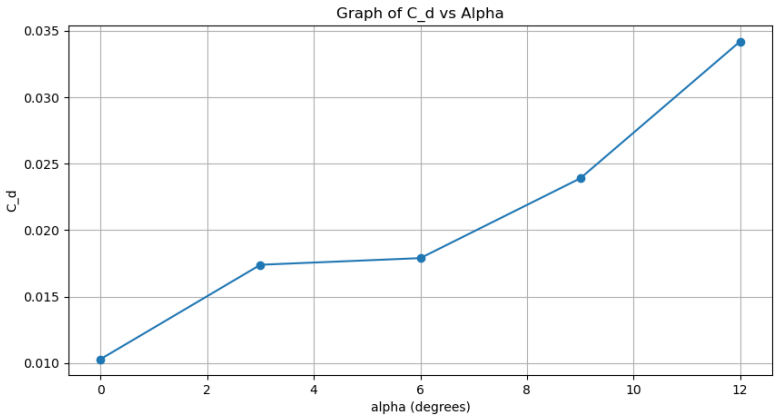

Fig. 32 – C_D generated by the NACA 4415 program

lated values of C_L and C_D follow the same trend as the reference values, with differences that can be attributed to the available computational power, variations in simulation conditions, or the experimental uncertainties in the reference studies.

NACA 0009

Using aerodynamic simulation software, simulations were performed employing the NACA 0009 airfoil profile, with a Reynolds number of $Re = 37{,}000$ and a speed of 7.6 m/s.

The results obtained were then compared with data available in the literature, specifically in the study "CFD Analysis of a Naca 0009 Aerofoil at a Low Reynolds Number"(GORGULU; SMITH; JOHNSON, 2021). The tables below show the comparison of drag (C_D) and lift (C_L) coefficients between the simulation results and the reference data, along with the calculated percentage errors.

The simulation results show good agreement with the reference data in terms of drag coefficients (C_D) and exhibit some variations in lift coefficients (C_L). However, the percentage of errors is generally low, indicating good agreement between the simulation results and the

Tab. 8 – Comparison of Drag Coefficients (C_D) with Percentage Errors

Angle of Attack (º)	C_D **(Simulation)**	C_D **(Reference)**	**Error** C_D **(%)**
0	0.02071334	0.01742	19.01
5	0.04762818	0.04597	3.52
10	0.104931	0.10172	3.14
15	0.2687369	0.24439	9.96

Tab. 9 – Comparison of Lift Coefficients (C_L) with Percentage Errors

Angle of Attack (º)	C_L **(Simulation)**	C_L **(Reference)**	**Error** C_L **(%)**
0	0.00359632	0	-
5	0.4984791	0.461	8.08
10	0.738343	0.748	-1.33
15	0.701543	0.661	6.05

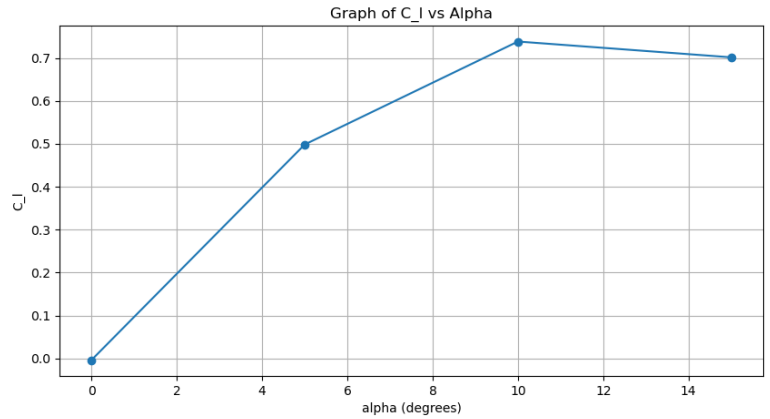

Fig. 33 – C_L generated by the NACA 0009 program

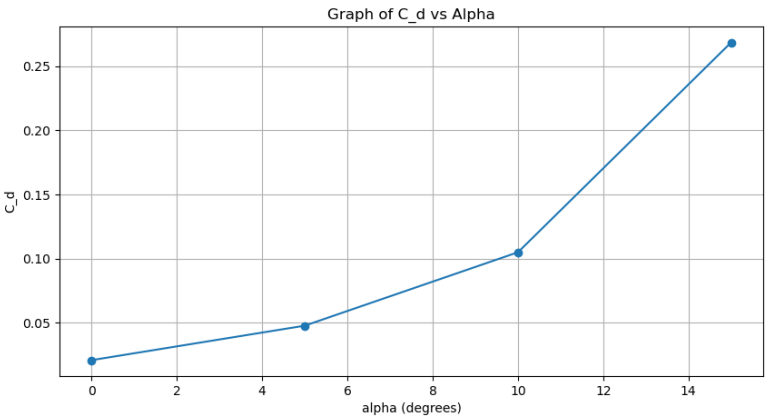

Fig. 34 – C_D generated by the NACA 0009 program

reference data.

5.3 Practical Use of the Software: Analysis of NACA 2412 and 4412 Airfoil Profiles

To exemplify situations in which the program can be useful, a brief study was conducted on the NACA 2412 and NACA 4412 aerodynamic profiles. However, it is not intended for an exhaustive analysis, as it is not the central focus of this work.

The software has proven useful in various applications, facilitating the study of airfoil profiles and their parameters by eliminating repetitive steps. This allows for a greater focus on the effects and comparisons of the results. A brief study was conducted simulating the NACA 2412 and 4412 airfoil profiles to compare and analyze the effects at 51.4815 m/s and a Reynolds number of 6×10^6. The results were used to validate the variation of the first digit of the NACA profiles, observing how it influences drag and lift at angles of 0, 10, and 15 degrees. The data for these profiles are presented in Tables 10 and 11.

For NACA 2412, the first digit '2' indicates a maximum curvature of 2% of the chord, while for NACA 4412, the first digit '4' indicates a maximum curvature of 4% of the chord.

Tab. 10 – Data for NACA 2412 airfoil

Angle (degrees)	C_D	C_L
0	0.0140	0.1998
10	0.0172	1.218
15	0.0753	1.526

Tab. 11 – Data for NACA 4412 airfoil

Angle (degrees)	C_D	C_L
0	0.0181	0.4037
10	0.0288	1.414
15	0.0858	1.7089

For both NACA 2412 and NACA 4412 airfoil profiles, the lift coefficient (C_L) increases with angle of attack. However, it is observed that the NACA 4412, which has a higher initial digit indicating greater maximum curvature, consistently presents higher C_L values compared to the NACA 2412 at the same angles of attack. This indicates that a higher maximum curvature results in a greater lift generation.

Similarly, the drag coefficient (C_D) increases with angle of attack for both airfoil profiles. The NACA 4412 exhibits higher C_D values than the NACA 2412 at all examined angles of attack. This suggests that although a higher maximum curvature increases lift, it also leads to higher drag.

Coefficient of pressure

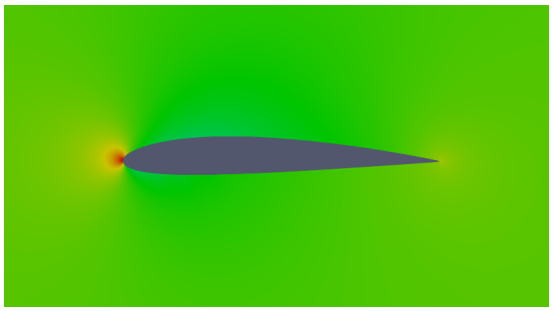

Fig. 35 – 2412 0º

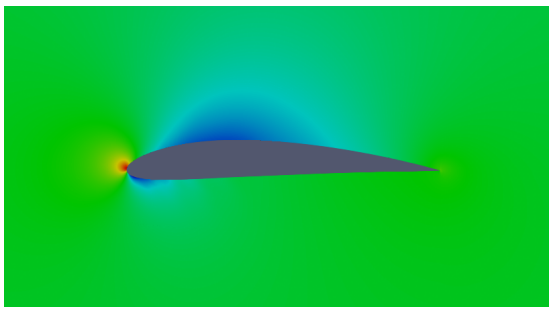

Fig. 36 – 4412 0º

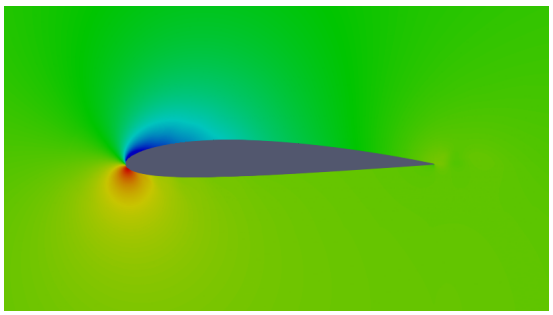

Fig. 37 – 2412 10º

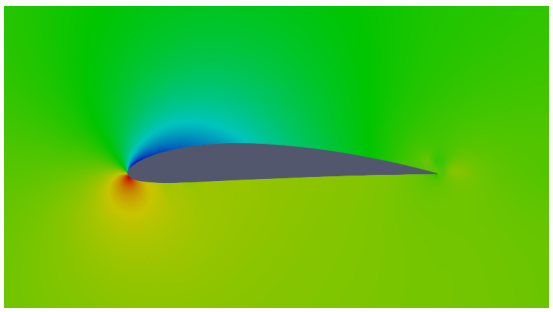

Fig. 38 – 4412 10º

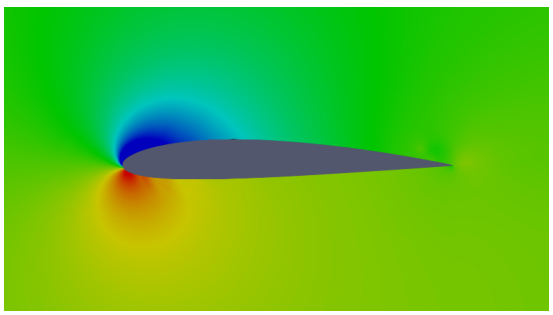

Fig. 39 – 2412 15º

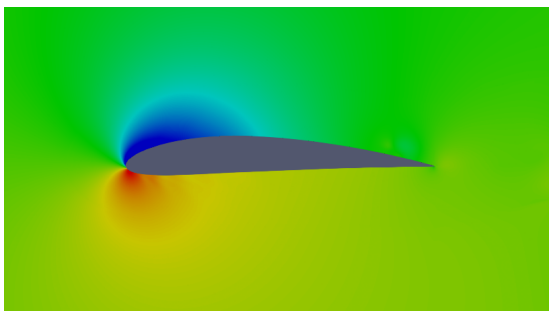

Fig. 40 – 4412 15º

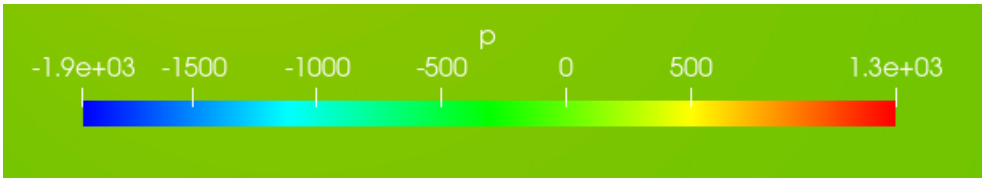

Fig. 41 – Pressure range

When analyzing pressure field images around the profiles, it is observed that as the angle of attack varies, there is a shift in the high-pressure point, primarily due to the higher percentage of curvature of the NACA 4412 profile compared to the NACA 2412. This difference results in higher pressure generated on the lower surface of the 4412 profile than on the 2412 profile at the same angle of attack, thereby leading to increased lift as there exists a pressure differential between the upper and lower surfaces. This is particularly observed at zero angle of attack for the 4412 profile due to its lesser symmetry, causing a greater difference between the pressure on the upper and lower surfaces of the profile.

The data in Tables 10 and 11 demonstrate that increasing the maximum curvature (as indicated by a higher initial digit) results in higher C_L values, indicating an increase in lift generation, and higher C_D values, indicating an increase in aerodynamic drag. Even though, the lift generated by the 4412 airfoil is higher than that by the 2412, but the drag remains virtually the same. Therefore, the choice between these airfoil profiles involves a trade-off between lift and drag. A higher curvature (e.g., NACA 4412) is beneficial for applications requiring higher lift but comes with the penalty of increased drag. On the other hand, a lower curvature (e.g., NACA 2412) offers lower drag but also lower lift.

6 Conclusion

Following the proposed objective, a tool was developed to perform aerodynamic simulations of NACA 4-digit airfoils. In the development of this tool, tests were conducted covering both incompressible and compressible flow. In the context of incompressible flow, simulations were carried out with a variety of aerodynamic profiles and different Reynolds numbers. In all tested configurations, the obtained results were consistent and close to data previously reported in the literature. However, regarding compressible flow simulations, fewer comparisons were made due to limitations in computational power and time constraints. This aspect can be further improved in future work, allowing for a more comprehensive and detailed analysis of results under these conditions. This tool was validated through comparison with reliable data, verifying the simulation process, its results, and the potential of this approach to simplify and expedite the aerodynamic simulation cycle. The methodology demonstrated efficiency and speed, particularly in the meshing and domain configuration phases, without significant setbacks.

The approach proved to be applicable in different contexts, eliminating the need for an in-depth prior knowledge of simulation software. This provided a more dynamic and simplified experience, allowing the focus to remain on the essential aspects of the study without the burden of complex and often repetitive technical details.

The agility of the process was notable in the domain configuration phase and mesh generation, areas that often require significant technical expertise in traditional simulations. Therefore, the proposed methodology stood out as a useful tool for researchers and engineers seeking simulation results without the complexity associated with traditional software.

The validation of the results through comparison with literature data, especially those already established for simulation validation, demonstrated the reliability of the obtained results. Being a study aimed at improving the process, this validation reinforces the program's credibility and demonstrates its ability to reproduce expected behaviors in broader contexts.

Furthermore, data compilation, often performed manually, was noted. The automation of the data collection and organization process significantly accelerated the analysis, creation of graphs, and comparisons, facilitating the researchers work.

In conclusion, the approach simplified the simulation process, indicating a more efficient and accessible direction for conducting aerodynamic studies. Moreover, the obtained results provide a solid platform for future work analysis and continuous methodology improvement.

For future work, improvements are highlighted in several areas of the project, including more options for easy customization of other criteria, boundary conditions, and more complex models, such as other fluids like water, more complex meshes, and the automatic development

of other types of profiles, such as NACA 5 or 6 digits, or even other models, which can assist in the interactive modification of profiles to achieve more specific models. Additionally, the simulation of wings in 3D domains is also a promising area for future developments, allowing for more detailed and realistic analyses.

Another crucial point for future development is the improvement of the program's presentation, enhancing the user interface to make it more intuitive and user-friendly. This will facilitate handling by users with different levels of experience in computational simulation, contributing to a more efficient and satisfying experience.

These planned improvements represent a continuous commitment to substantial enhancements in the functionality, usability, and reliability of the program over time.

Bibliography

ABBOTT, I. H.; DOENHOFF, A. E. V. *Theory of Wing Sections: Including a Summary of Airfoil Data*. [S.l.]: Dover Publications, 1959. mentioned on the page 4.

AEROSPACEWEB.ORG. *NACA Airfoil Series*. URL: https://www.aerospaceweb.org/question/airfoils/q0071.shtml. Accessed: 28/05/2024. mentioned on the page 12.

AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS. *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*. [S.l.], 1998. mentioned 2 times on pages 1 e 2.

ANDERSON, J. D. *Fundamentals of Aerodynamics*. [S.l.]: McGraw-Hill Education, 2001. mentioned 5 times on pages 6, 1, 8, 9 e 10.

ANDERSON, J. D. J. *Fundamentals of Aerodynamics*. 3rd. ed. [S.l.]: publisher, 2011. mentioned 4 times on pages 4, 14, 16 e 40.

AWRUCH, A. M.; BRAUN, A. L.; GRECO, M. Engenharia do vento computacional e suas aplicações na engenharia civil: Análise aerodinâmica e aeroelástica. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, v. 31, n. 1, p. 55–64, 2015. Disponível em: <https://www.scipedia.com/public/Awruch_et_al._2014a>. mentioned on the page 16.

BAKER, T. J. Mesh generation: Art or science? *Progress in Aerospace Sciences*, v. 41, n. 1, p. 29–63, 2005. ISSN 0376-0421. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0376042105000114>. mentioned on the page 2.

BRANDON, R. A.; SEIDEL, W. Experimental analysis of leading-edge vortices in oscillating naca 0012 airfoils. *AIAA Journal*, v. 26, n. 8, p. 1025–1032, 1988. mentioned on the page 5.

BROEREN, A. P.; BRAGG, M. B. Investigación de la variación del flujo de desprendimiento no estacionario a lo largo del perfil. *Journal of Aerodynamics*, v. 25, n. 3, p. 112–125, 2001. mentioned on the page 4.

CHUNG, T. J. *Computational fluid dynamics*. [S.l.]: Cambridge University Press, 2000. mentioned on the page 26.

COSTA, E. e. a. Simulação numérica de escoamento sobre perfil aerodinâmico eppler 423. 2019. Accessed in: 27/05/2024. mentioned on the page 26.

COURTNEY, M. Try before you fly. *Engineering Technology*, v. 9, n. 7, p. 43–48, 2014. mentioned on the page 1.

DOWNEY, A. B. *Think Python: How to Think Like a Computer Scientist*. 2nd. ed. [S.l.]: O'Reilly Media, 2015. ISBN 978-1-491-90102-6. mentioned on the page 24.

FERZIGER, J. H. *Computational Methods for Fluid Dynamics*. [S.l.]: Springer, 2002. mentioned on the page 18.

Fluent Inc. *FLUENT 6.3 User's Guide*. [S.l.: s.n.], 2006. mentioned on the page 18.

FOX, R. W. et al. *Introdução à Mecânica dos Fluidos*. 9. ed. [S.l.]: LTC Atlas, 2018. mentioned 2 times on pages 14 e 15.

GENC, M. S.; KARASU, I.; KAYNAK, U. Numerical simulation of flow around naca 0012 airfoil with and without leading-edge slat. *International Journal of Mechanical Sciences*, v. 52, n. 2, p. 210–219, 2010. mentioned on the page 6.

GONZáLEZ, D. C. *Tesis: Estudio experimental y numérico de la separación de flujo turbulento alrededor de un perfil aerodinámico*. 2011. Instituto Politécnico Nacional, México. mentioned on the page 18.

GORGULU, K.; SMITH, S.; JOHNSON, M. Cfd analysis of a naca 0009 aerofoil at a low reynolds number. *Journal of Fluids Engineering*, v. 143, n. 9, p. 091101, 2021. mentioned on the page 43.

HIRSCH, C. *Numerical methods in fluid dynamics: A practical introduction*. [S.l.]: Butterworth-Heinemann, 2007. mentioned on the page 26.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, v. 9, n. 3, p. 90–95, 2007. mentioned on the page 24.

JACOBS, E.; WARD, K.; PINKERTON, R. *The Characteristics of 78 Related Airfoil Sections from Tests in the Variable-Density Wind Tunnel*. [S.l.], 1933. mentioned on the page 12.

JAVAHERIAN, H.; SAHIN, M. Numerical simulation of flow over naca 0012 airfoil using openfoam. *International Journal of Computational Fluid Dynamics*, v. 26, n. 6, p. 403–415, 2012. mentioned on the page 6.

KHALID, M. et al. Numerical simulations of unsteady flow over a naca 0012 airfoil using high-fidelity methods. *Computers & Fluids*, v. 122, p. 161–175, 2015. mentioned 2 times on pages 5 e 6.

LADSON, C. L. *Effects of independent variation of Mach and Reynolds numbers on the low-speed aerodynamic characteristics of the NACA 0012 airfoil section*. [S.l.], 1988. mentioned on the page 4.

LADSON, C. L.; JR, C. W. B. *Development of a Computer Program to Obtain Ordinates for NACA 4-digit, 4-digit Modified, 5-digit, and 16-series Airfoils*. USA, 1975. mentioned on the page 12.

LAKHAN, S. E.; JHUNJHUNWALA, K. Open source software in education. *Educause Quarterly*, v. 31, n. 2, p. 32, 2008. mentioned on the page 1.

LAUNDER, B. E.; SPALDING, D. B. *Mathematical Models of Turbulence*. London: Academic Press, 1972. mentioned on the page 5.

LEE, J.; SU, C. Estudio de la separación del flujo en condiciones de bajo número de reynolds. *Journal of Fluid Mechanics*, v. 203, p. 456–468, 2010. mentioned on the page 5.

LOPEZ, M.; RODRIGUEZ, L.; GARCIA, J. Effects of surface roughness on the aerodynamic behavior of naca 0012 airfoil. *AIAA Journal*, v. 34, n. 7, p. 891–905, 2016. mentioned on the page 7.

LUNDH, F. *Python Standard Library*. [S.l.]: O'Reilly Media, 1999. ISBN 978-1-56592-529-3. mentioned on the page 24.

LUTZ, M. *Learning Python*. 5th. ed. [S.l.]: O'Reilly Media, 2013. ISBN 978-1-449-35573-9. mentioned on the page 23.

MANWELL, J. F.; MCGOWAN, J. G.; ROGERS, A. L. *Wind Energy Explained – Theory, Design and Application*. 2. ed. [S.l.]: Wiley, 2009. mentioned 2 times on pages 6 e 8.

MARTELLI, A. *Python in a Nutshell*. 2nd. ed. [S.l.]: O'Reilly Media, 2005. ISBN 978-0-596-10046-9. mentioned on the page 24.

MARZOCCA, D. *The NACA Airfoil Series*. Accessed: 28/05/2024. Disponível em: <https://www.clarkson.edu/>. mentioned on the page 13.

MCALISTER, K. W.; TAKAHASHI, R. K. *NACA 0012 airfoil in oscillating subsonic and transonic flows*. [S.l.], 1991. mentioned on the page 4.

MCKINNEY, W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 2nd. ed. [S.l.]: O'Reilly Media, 2017. ISBN 978-1-491-95761-0. mentioned on the page 24.

MENTER, F. R. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal*, v. 32, n. 8, p. 1598–1605, 1994. mentioned on the page 5.

MENTER, F. R. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, v. 32, n. 8, p. 1598–1605, 1994. mentioned on the page 19.

MENTER, F. R.; KUNTZ, M.; LANGTRY, R. Ten years of industrial experience with the sst turbulence model. *Turbulence, Heat and Mass Transfer*, v. 4, 2003. mentioned on the page 6.

MIRANDA, L. R. Application of computational aerodynamics to airplane design. *Journal of Aircraft*, v. 21, n. 6, p. 355–370, 1984. Disponível em: <https://doi.org/10.2514/3.44974>. mentioned on the page 1.

NACA. *NACA0012 Grids*. Available in: <https://turbmodels.larc.nasa.gov/naca0012_grids.html>. Accessed in: 27/05/2024. mentioned on the page 26.

NASA. *NACA0012 Airfoil with Synthetic Jets and Coanda Effector*. 2021. <https://turbmodels.larc.nasa.gov/naca0012_val_sa.html>. Accessed: 27/05/2024. mentioned 3 times on pages 6, 30 e 31.

NASA. *The Spalart-Allmaras Turbulence Model*. 2021. [Accessed: October 31, 2021]. Disponível em: <https://turbmodels.larc.nasa.gov/spalart.html#sa>. mentioned 4 times on pages 6, 11, 18 e 19.

NASA. *Turbulence Modeling Resource*. [S.l.], 2021. Disponível em: <https://turbmodels.larc.nasa.gov/>. mentioned 2 times on pages 19 e 20.

NETO, A. S.; SOARES, R. F. *Turbulência dos Fluidos*. Dissertação (Mestrado) — Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Mecânica, Uberlândia, 2014. mentioned on the page 17.

NILSSON, E. et al. Assessment of turbulence models for prediction of flow around naca 0012 airfoil. *Journal of Aerospace Engineering*, v. 228, n. 3, p. 417–430, 2015. mentioned on the page 6.

OPENFOAM. *OpenFOAM Guide, k-omega SST*. Accessed: 28/05/2024. Disponível em: <https://www.openfoam.com/documentation/guides/latest/doc/guide-turbulence-ras-k-omega-sst.html>. mentioned on the page 20.

openfoam. *OpenFOAM Guide, Spalart-Allmaras*. [Accessed: 28/05/2024]. Disponível em: <https://www.openfoam.com/documentation/guides/latest/doc/guide-turbulence-ras-spalart-allmaras.html>. mentioned on the page 18.

OPENFOAM. *validation naca0012 Openfoam*. Available in: <https://www.openfoam.com/documentation/guides/latest/doc/verification-validation-naca0012-airfoil-2d.html>. Accessed in: 27/05/2024. mentioned on the page 27.

OPENFOAM. *OpenFOAM. The open source computational fluid dynamics toolbox*. 2013. <http://www.openfoam.com/>. mentioned 4 times on pages 6, 21, 22 e 23.

PIRES, P. S. M.; ROGERS, D. A. Free/open source software: An alternative for engineering students. In: IEEE. *Frontiers in Education, 2002. FIE 2002. 32nd Annual*. [S.l.], 2002. v. 1, p. T3G–T3G. mentioned on the page 1.

POPE, S. B. *Turbulent Flows*. [S.l.]: Cambridge University Press, 2000. mentioned on the page 18.

RINOIE, K.; TAKEMURA, F. Aplicaciones de fluorescencia y fotografía para el análisis del comportamiento del flujo. *Journal of Aerodynamics*, v. 32, n. 5, p. 275–287, 2004. mentioned on the page 5.

ROACHE, P. J. *Verification and Validation in Computational Science and Engineering*. [S.l.]: Hermosa Publishers, 2009. mentioned on the page 2.

ROSSUM, G. van; JR., F. L. D. *Python 3 Reference Manual*. [S.l.]: CreateSpace, 2009. ISBN 978-1-441-43470-6. mentioned on the page 23.

RUIZ, P.; SMITH, J.; JOHNSON, E. Aerodynamic characteristics of airfoils at low reynolds numbers. *Journal of Aircraft*, v. 48, n. 4, p. 1235–1240, 2011. mentioned on the page 11.

RUMSEY, C. L.; GATSKI, T. B.; SELLERS, W. L. Summary of the 2001 cfd validation workshop on synthetic jets and turbulent separation control. In: *AIAA Paper*. [S.l.: s.n.], 2001. p. 2001–2820. mentioned on the page 6.

SAE Brasil. *26ª Competição SAE Brasil Aerodesign 2024 Regulations*. 2024. Appendix 3. mentioned on the page 3.

SANTOS, F. D. d.; CARVALHO, N. B.; SILVA, D. d. N. Análise numérica de um perfil aerodinâmico modelo naca 4415. *Bacharel em Ciências e Tecnologia*, Campus Universitário, Lagoa Nova, Natal - RN, CEP: 59078-970, 2024. mentioned on the page 42.

SOWMYASHREE, Y. et al. Study on effect of semi-circular dimple on aerodynamic characteristics of naca 2412 airfoil. *AIP Conference Proceedings*, v. 2204, p. 030009, 2020. mentioned on the page 40.

SPALART, P. R.; ALLMARAS, S. R. A one-equation turbulence model for aerodynamic flows. *AIAA journal*, v. 30, n. 11, p. 289–306, 1992. mentioned 2 times on pages 5 e 6.

SUNDNES, J. *Programming for Everyone: Learning with Python*. [S.l.]: Tech University Press, 2020. mentioned 2 times on pages 23 e 24.

TABOR, G.; SMITH, J.; BROWN, D. Openfoam validation in simulations of naca 0012 airfoil. *Journal of Computational Physics*, v. 285, n. 2, p. 345–358, 2013. mentioned on the page 7.

TANGLER, J. L.; SOMERS, D. M. *NREL Airfoil Families for HAWTs*. [S.l.], 1995. mentioned on the page 8.

THILEEPANRAGU, R. et al. *Passive flow control over NACA0012 airfoil using vortex generators*. 2010. ResearchGate. mentioned 2 times on pages 6 e 11.

TIMMER, W. Two-dimensional low-reynolds number wind tunnel results for airfoil naca 0018. *Wind Engineering*, v. 32, n. 6, p. 525–537, 2008. mentioned on the page 40.

TOOLS, A. *Airfoil Tools*. Accessed on 2023-11-25. Disponível em: <http://airfoiltools.com/search/index>. mentioned on the page 13.

VERSTEEG, H. K.; MALALASEKERA, W. *An Introduction to Computational Fluid Dynamics–The Finite Volume Method*. [S.l.]: Pearson, 2007. mentioned 4 times on pages 15, 16, 17 e 18.

WILCOX, D. C. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA journal*, v. 26, n. 11, p. 1299–1310, 1988. mentioned on the page 5.

ZHANG, L.; LI, J.; ZHU, J. Large eddy simulation of unsteady flow around naca 0012 airfoil using openfoam. *Journal of Fluids Engineering*, v. 36, n. 4, p. 456–468, 2014. mentioned on the page 6.

A Github link

The source code for the aerodynamic simulation automation program developed in this thesis is available at:
<https://github.com/Thiago-Vinicius-Costa/Aerodynamic_Simulation_Automation_Program.git>.

B Manual for Aerodynamic Simulation Automation Program

Introduction

The Aerodynamic Simulation Automation Program is a tool developed in Python that allows interaction with OpenFOAM to perform aerodynamic simulations of 4-digit Naca profiles. With an intuitive graphical interface, the program simplifies the process of configuring and running simulations, making it ideal for aeromodeling competition teams, profile studies, and beginners with OpenFOAM.

Process

The program's initial screen presents fields for entering essential data to start a simulation:

- **Naca Points File:** This option allows the user to load a file containing the Naca profile points. The file should include coordinates that define the shape of the Naca airfoil, typically in a TXT format. Each line in the file represents a point on the airfoil, with columns indicating the x and y coordinates. It is recommended to have a minimum of 100 points for a better definition of the profile.
- **Naca and Study Angles:** Users can input the Naca profile code, such as "0012", specifying the desired airfoil geometry. Additionally, users can specify the angles at which they want to conduct aerodynamic studies. Angles should be provided in degrees, separated by commas, e.g., "0, 1, 2.5".
- **Mesh Options:** The user has the flexibility to choose between a standard mesh validated for Naca 0012 or a custom mesh adapted for compressible cases. The mesh plays a critical role in accurately capturing the flow behavior around the airfoil. The choice between a standard and custom mesh depends on the specific requirements of the simulation, such as resolution and boundary conditions.
- **Simulation Configuration:** Users can configure various parameters essential for the simulation process. This includes selecting between compressible or incompressible simulation models, which determine the behavior of the fluid flow. Additionally, users can set variables such as velocity (in meters per second), kinematic viscosity (ν), nut (recommended to be equal to the ν value), and nutilda (recommended to be four times the ν value).

Simulation Execution

After configuring all necessary parameters, the user can click on "Run Simulation"to start the simulation process. The program will create a file for each specified angle and initialize them sequentially. The results will be grouped at the end of the process.

Error Messages

The program provides error messages to assist users in case of invalid inputs or problems during execution:

- **Error in Angles:** If the specified angles are incorrect or not entered correctly, the program will display an error message indicating the necessary correction.
- **OpenFOAM in Use:** If OpenFOAM is in use by another process, the system itself will notify the user about the impossibility of starting a new simulation until OpenFOAM is available.
- **Open Excel File:** When performing simulations sequentially and generating results files in Excel format, it is important to ensure that these files are not open during the compilation of results, as this may cause errors in the operation.

Mesh Modification

The program offers extensive flexibility in modifying mesh parameters. Users can adjust the following criteria to optimize the mesh according to the specific needs of each simulation:

- Distances relative to the inlet and outlet points
- Sizes of cells at the leading edge, trailing edge, and middle
- Position of the separation point
- Thicknesses of layers at the boundary and the first layer
- Maximum cell sizes at the inlets, outlets, and the junction between inlet and outlet
- Number of divisions in and out the boundary layer, at the tail, at the leading edge, and the trailing edge

The tail angle is adjusted according to the angle of attack to better capture the effects downstream of the aerodynamic profile.

Installation

The program installation can be done through GitHub at:

<https://github.com/Thiago-Vinicius-Costa/Aerodynamic_Simulation_Automation_Program.git>.

where the source code and a README file explaining the installation process and system requirements are available. The GitHub repository link will be provided to facilitate user access.

Documentation

The code of the Aerodynamic Simulation Automation Program was developed as part of the developer's undergraduate thesis. In the GitHub repository, users will find the final document of the thesis, which includes detailed information about the methodology used, software documentation, and obtained results. This document serves as an additional source of information for users interested in understanding the operation and objectives of the program.

C README

Aerodynamic Simulation Automation Program

Overview

This Python program aims to automate the process of aerodynamic simulation using the OpenFOAM software. It provides a user-friendly interface for generating and refining mesh grids, guiding users through simulation steps from pre-processing to post-processing. By leveraging Python's libraries and a clear structure, the program enhances accessibility to aerodynamic simulation techniques, making it suitable for both beginners and experts in the field.

Requirements

To run this program, you need to have the following installed:

- OpenFOAM 2.2.12: Ensure that you have OpenFOAM version 2.2.12 installed on your system. If you're using a different version, you'll need to modify the version in the standard/incompressible/run_simulation directory to match the version you're using.
- Python: The program is written in Python, so you need to have Python installed on your system. Python 3.0 is recommended.
- tkinter: This program's graphical interface is built using the tkinter module, which is included in standard Python installations.
- numpy: Install numpy using the following command: `pip install numpy`
- matplotlib: Install matplotlib using the following command: `pip install matplotlib`
- pandas: Install pandas using the following command: `pip install pandas`
- openpyxl: Install openpyxl using the following command: `pip install openpyxl`
- tqdm: Install tqdm using the following command: `pip install tqdm`

Installation

Install OpenFOAM

Follow the installation instructions for OpenFOAM 2.2.12 specific to your operating system. You can download OpenFOAM 2.2.12 from openfoam.org

<https://github.com/Thiago-Vinicius-Costa/Aerodynamic_Simulation_Automation_Program.git>.

Clone Repository

Clone this repository to your local machine using the following command:

```
git clone https://github.com/Thiago-Vinicius-
Costa/Aerodynamic_Simulation_Automation_Program.git
```

Replace your-username with your GitHub username.

Recommended: Visual Studio Code It is recommended to use Visual Studio Code (VSCode) for running and modifying this program. VSCode provides an integrated terminal, powerful debugging tools, and extensive extensions for Python development. You can download VSCode from <https://code.visualstudio.com/>.

Usage

Navigate to Directory: Open a terminal or command prompt and navigate to the directory where you cloned the repository.

Run the Program: Run the Python script `aero_sim.py` using the following command:

```
python Run.py
```

Follow Instructions

The program will prompt you with instructions for each step of the simulation process. Follow the on-screen prompts to generate and refine mesh grids, perform simulations, and analyze results.

Select Coordinate Document or Specify NACA Profile: After launching the program, you will be prompted to either select a document containing coordinates (default is 100 points) or specify a desired NACA profile with 4 digits.

Mesh Refinement: While the default mesh can be used for simulation, it is recommended to refine the mesh based on the complexity of the simulation, such as high velocities, compressible behavior, etc.

Provide Simulation Properties: Input simulation properties such as velocity, temperature, pressure, etc.

Note: When specifying multiple angles, the program may take longer to complete due to increased computational complexity.

Note: To alter the number of iterations, navigate to one of the standard models in the system folder, open the `controlDict` document, and modify `endTime` or `deltaT`.

Note: The simulation results will be saved in the "results"folder, and the plots will be saved in the "graphics"folder.

Contributions

Contributions to this project are welcome. If you find any bugs or have suggestions for improvements, please open an issue or submit a pull request on the GitHub repository.

License

This program is licensed under the MIT License. See the `LICENSE` file for details.

Contact

For any questions or inquiries, feel free to contact the project maintainer at <180078330@aluno.unb.br>.

D License