Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Aerospace Engineering (37.th cycle)

Machine learning for transitional and turbulent flows

Luca Muscarà

* * * * *

Supervisors
Prof. Ferrero Andrea, Supervisor
Prof. Larocca Francesco Co-supervisor

Politecnico di Torino

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Luca Muscarà

Turin, September 3, 2025

Summary

The purpose of this work is to study and apply data-driven models within the field of Computational Fluid Dynamics (CFD), focusing on enhancing current turbulence models with these advanced techniques. Modern CFD faces significant challenges in accurately predicting complex phenomena such as fluid separation and the transition from laminar to turbulent flow. Failures in these predictions can lead to incorrect evaluations of critical parameters like force coefficients, which are vital in the design of aerospace components. Traditional approaches, like the Reynolds-Averaged Navier-Stokes (RANS) equations, offer a simplified model of the Navier-Stokes equations but often result in low accuracy, particularly in predicting separation and transition phenomena.

This research explores the use of data-driven models, with a particular focus on field inversion and machine learning (ML) methods, to improve the predictive capabilities of existing RANS models. A significant aspect of the work has been the implementation of these models almost from scratch, including the Spalart-Allmaras model and the adjoint method to compute the gradient of the target function required to perform the field inversion. A pseudo-time version of the adjoint solver was also implemented.

This study involved a range of test cases, including transient flow over a flat plate, a backward-facing step, and low Reynolds number simulations around airfoils such as the SD7003 and the NACA0021. A multi-objective analysis was specifically carried out for the NACA0021 test case, offering useful insights into the optimization strategy. Both traditional neural networks and more advanced architectures, like the U-Net model, were applied to improve predictive capability. Furthermore, the impact of different objective functions within the field inversion process was closely examined to understand their influence on the enhancement of the RANS model. This work demonstrates that by integrating data-driven approaches, particularly through the use of ML and field inversion techniques, it is possible to enhance the accuracy of turbulence models in CFD, potentially reducing the need for costly high-fidelity simulations during the design of new components.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Professor Francesco Larocca and Professor Andrea Ferrero, for accepting me as their student and for the guidance they have provided throughout this journey. I am especially grateful to Professor Ferrero, who has supported me since my master's thesis. Whatever I know today is largely thanks to his mentorship and patience.

I am also grateful to my colleagues at Optimad, whose support was fundamental for the development of this thesis. I thank them for giving me the opportunity to pursue a PhD within the company and for everything they taught me along the way.

My deepest thanks go to my parents, who have always believed in me and made it possible for me to follow this path. I hope I can make you proud.

Finally, but by no means least, I would like to thank my sweetheart, Laura. Without even realizing it, she helped me release stress and stay balanced during the hardest times. I must also thank this PhD itself, because it was along this journey that I had the fortune to meet her.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

Nowadays, we are living in the era of artificial intelligence, where scientists are striving to harness the vast amounts of data collected over the years. By leveraging sophisticated algorithms, they aim to extract meaningful relationships from this data—often in a black-box manner. The algorithms I am referring to are the foundation of data-driven models.

Artificial intelligence is a broad and general concept; we can think of it as an overarching set. A more precise term for the algorithms that I am discussing is machine learning algorithms. As the name suggests, these algorithms enable machines—specifically computers—to learn intrinsic patterns and correlations within data without being explicitly programmed to do so.

As I write this PhD dissertation, machine learning algorithms are gaining immense popularity. However, they are not new. Their development dates back to the 1950s, but at that time, computational power was insufficient to fully explore their potential. Today, the exponential growth in computing capabilities has propelled these algorithms into widespread use across numerous fields, including finance, surveillance, engineering, healthcare, robotics, and many others.

This thesis focuses on the application of data-driven models in a specific area of aerospace engineering: fluid dynamics. The study reported in this thesis has been dedicated to leveraging existing techniques to identify limitations and enhance methodologies.

1.1.1 Thesis objective

In aerospace engineering, one of the most intriguing and challenging topics is the study of turbulence. This widespread phenomenon is present in everyday life—from the swirling smoke of a cigarette to the airflow around objects—and plays a crucial role in more complex scenarios such as airplane wing design and the analysis of secondary

flows in turbine blades within aircraft propulsion systems. Turbulence also significantly impacts combustion processes, making its understanding essential for numerous applications in aerodynamics, propulsion, and fluid dynamics.

Turbulent flows are characterized by their irregular, chaotic motion and an enhanced ability to transport and mix properties such as thermal energy, momentum, and chemical species. Despite being one of the oldest studied physical phenomena, turbulence remains an active area of research due to the inherent challenges in modeling and predicting its behavior. Its complexity arises from its chaotic nature and the vast range of interacting scales, making it difficult to analyze using traditional methods.

This thesis focuses on turbulence modeling and the use of data-driven approaches to enhance existing models. Extensive research has been conducted in this direction, with particular attention to the persistent challenge of balancing accuracy and computational cost. Due to the complexity of fluid dynamics problems involving turbulence and intricate transition phenomena, the Reynolds-Averaged Navier–Stokes (RANS) equations are widely used as an approximate formulation of the Navier–Stokes (N-S) equations.

Industrial design processes require evaluating numerous geometries while operating within computational limitations. As a result, high-fidelity methods such as Direct Numerical Simulations (DNS) and Large Eddy Simulations (LES), which provide detailed turbulence information, are often impractical for preliminary design stages. Instead, RANS models are widely used due to their lower computational cost and are expected to remain integral in aerospace engineering for years to come [11]. Various aerospace applications involve performance analysis in flow regimes characterized by low Reynolds numbers. In such regimes, a consistent portion of the boundary layer remains laminar, which increases the likelihood of flow separation. Examples of these kind of flows include those present in propellers for unmanned aerial vehicles [110, 108, 102, 32, 105, 49] or low-pressure gas turbines in turbofan engines [58, 24], as well as for small-scale wind turbines [41, 13]. Flow separation phenomena also occur in rocket nozzles [90], and on thick airfoils, where short to long bubble separation on the suction side can degrade aerodynamic performance and affect stall behavior [15]. Accurate simulation of these flow phenomena is essential for improving the design process.

Although RANS models provide reasonable estimates of the average flow field, their accuracy diminishes in the presence of transition or separation. These limitations stem from the underlying simplifications, which are only approximately valid in fully turbulent, attached flows. Recent efforts have focused on enhancing turbulence models to overcome these constraints. Two key advancements support this progress. First, increasing computational power has enabled high-fidelity simulations, offering valuable datasets for turbulence research. Second, the rise of machine learning techniques has introduced new opportunities for improving turbulence predictions, particularly in separated flows. Various machine learning strategies have been explored [20], including field inversion and machine learning (FIML) [65, 84], corrections to the Boussinesq approximation in Reynolds stress computations[47, 97], subgrid stress modeling in LES [71, 83], and RANS modeling of vortex breakdown [18]. As previously mentioned, the objective of

this thesis is to explore data-driven models, assessing both their potential and limitations in improving turbulence modeling. Specifically, my PhD research has focused on the FIML framework, aiming to understand its capabilities, identify its constraints, and develop novel approaches to enhance its applicability.

1.1.2 Outline

This research project was supported by Optimad S.r.l., a company that provided invaluable support throughout the three years of my research. Their extensive expertise in CFD and reduced-order models (ROM) greatly contributed to the progress of this work. In particular, Optimad granted me access to their CFD software, Immerflow, which was instrumental in carrying out my research.

The structure of this thesis follows the chronological development of the research over these three years. It is organized as follows:

- Chapter 2 provides a brief introduction to turbulence physics, including transition phenomena and the evolution of various models developed to describe these complex and fascinating behaviors. Additionally, it presents an overview of existing applications of data-driven models in turbulence research, summarizing different approaches explored over the years.

- Chapter 3 details the turbulence model used in this study, the Spalart-Allmaras model, along with some of its variations. This chapter also introduces the first set of results, as the research is structured into three main stages: (1) analysis of the original model, (2) enhancement of the model using field inversion, and (3) generalization through machine learning algorithms. The initial results, focusing on the original model, are presented here.

- Chapter 4 discusses the field inversion and machine learning framework, providing details on the governing equations, strategies to improve convergence, and the overall potential of this approach. The same test cases from Chapter 3 are revisited here, but with results obtained using the optimization framework. Additionally, an example of multi-objective optimization is presented.

- Chapter 5 introduces the fundamentals of one of the key machine learning algorithms, neural networks, followed by their application to leverage the results obtained through field inversion. This chapter also explores the selection of input parameters for this architecture and strategies for defining an appropriate subset of the database. The chapter concludes with a comparison of results between the original model, the augmented model, and the machine learning-enhanced approach.

- Chapter 6 presents a novel approach based on a modified combination of existing machine learning algorithms. Specifically, it explores the use of convolutional

neural networks structured in a U-Net architecture. This method was developed after identifying limitations in the traditional neural network approach and is further supported by the findings of my recent publication [62]. The chapter concludes with results from the final two test cases.

- Chapter 7 summarizes the key findings of this research and provides concluding remarks.

Chapter 2

The Challenge of Turbulence: Modeling, Simulation, and Transition in Fluid Flows

The Navier-Stokes equations provide a comprehensive description of fluid flow behavior. However, these equations do not have a general analytical solution due to their complexity and non-linearity. Over the years, scientists have developed various methods to approximate their solutions. Before the advent of modern computers, researchers were able to derive analytical solutions for highly simplified cases, such as Couette flow or Stokes flow. In other cases, they used approximations like potential flow theory, which assumes the flow is inviscid and irrotational, greatly simplifying the equations.

Scientists also employed numerical methods, such as finite difference techniques, to discretize and solve the Navier-Stokes equations numerically. This was a labor-intensive and time-consuming process since it required performing calculations step-by-step, often relying on mechanical calculators or plotting results by hand. Despite these challenges, early researchers made significant advances in fluid dynamics using these approaches.

Today, with the immense computational power available, even on devices as small as laptops, solving complex fluid flow problems has become much more efficient. Modern CFD tools can simulate highly intricate flows, leveraging the power of computers to handle millions of calculations in seconds, a task that once required entire rooms of equipment and days or weeks of manual effort.

Even with the great power of our modern computers, some assumption has to be made before starting fluid flow simulation. The main reason lies in the fact that several numerical methods and models are available to approximate the N-S equations when dealing with specific fluid regimes.

As far as the laminar regime is considered, things are greatly simplified and the full N-S can be directly solved. The problem arises when turbulence or transitional regimes are considered.

2.1 The Physics and Modeling of Turbulence Phenomena

Turbulent flow is a complex phenomenon characterized by chaotic and irregular motion, often arising when a fluid flows past solid surfaces or when adjacent fluid streams interact.

Figure 2.1: Free water jet issuing from a square hole into a pool, Leonardo Da Vinci. [34]

One of the earliest studies of turbulence can be attributed to Leonardo Da Vinci. Figure 2.1 illustrates his observation of a stream of water falling from a sluice into a pool, a

remarkably detailed and insightful study. In this depiction, multiple layers of vortices extend far beneath the water surface, with each swirling current creating concentric rings of bubbles. These bubbles expand and overlap, showcasing Da Vinci's keen understanding of fluid dynamics centuries ahead of modern turbulence theory.

Figure 2.2: The Great Wave off Kanagawa, Katsushika Hokusai. [79]

Another remarkable piece of art is Hokusai's iconic painting from the early 19th century, The Great Wave off Kanagawa (see Figure 2.2). In this masterpiece, Hokusai captures the drama of cargo ships struggling to navigate through a colossal, stormy sea. The image is renowned for its vivid portrayal of turbulence, with towering waves resembling a tsunami, perfectly encapsulating the overwhelming force and unpredictability of nature. Although seldom highlighted in major fluid dynamics texts, Hagen first described the transition from laminar to turbulent flow in 1854 [33]. By 1869, he used dark amber to visually illustrate these flow patterns, though the specific conditions for this transition were not well understood. The more widely recognized work on this subject came from Osborne Reynolds at the University of Manchester in 1883 [72]. Reynolds conducted pivotal experiments by injecting a colored jet into water flowing through a large glass pipe. He observed that at low velocities, the dye remained intact, while at higher velocities it dispersed, indicating the onset of turbulent flow. Reynolds introduced the dimensionless Reynolds number to predict this transition, which considers factors such as fluid viscosity and pipe diameter. This quantity, named Reynolds number by later

scientists such as Sommerfeld (1908) and Prandtl (1910) [76], is defined as follows:

$$Re = \frac{\rho V D}{\mu} \tag{2.1}$$

The Reynolds number is defined as the ratio of inertial forces to viscous forces in a fluid. Indeed at high Reynolds numbers, the inertial forces become dominant over the stabilizing effects of viscosity, leading the flow to transition from laminar to turbulent motion. To express this concept mathematically, consider a 3D flow described by the compressible N-S equations, neglecting pressure and gravitational effects. By non-dimensionalizing these equations, we obtain the following momentum equations:

$$\frac{\partial \rho^* u_i^*}{\partial t^*} = \frac{1}{Re} \frac{\partial {\tau_{ij}}^*}{\partial x_j^*} - \frac{\partial \rho^* u_i^* u_j^*}{\partial x_j^*} \tag{2.2}$$

In Eq. 2.2, the superscript '$*$' indicates the dimensionless variable. The time and spatial coordinates are represented by t and x_j, respectively. The fluid density is represented by ρ, and u_i denotes the velocity components, where the subscripts i and j refer to the spatial directions. The shear stress tensor, τ_{ij}, describes the viscous forces acting on the fluid. Therefore, Eq. 2.2 suggests that the transition is driven by the balance between the convective and viscous terms. Viscous forces act as a damping mechanism, smoothing out disturbances. However, as the Reynolds number increases, the influence of the convective nonlinearity becomes more pronounced. When this nonlinear convective term exceeds the damping effect of the viscous forces, flow instabilities develop, leading to the formation of three-dimensional structures and turbulent eddies. These eddies, driven by the energy transfer from the main flow, generate high levels of vorticity and contribute to the random, swirling motions characteristic of turbulence [73].

Unlike laminar flow, where fluid moves in smooth, parallel layers, turbulent flow is inherently three-dimensional and unsteady. This chaotic nature results in rapid fluctuations in velocity and pressure, which can be observed both in compressible and incompressible flows. Turbulence causes significant mixing, leading to much higher rates of heat transfer, mass diffusion, and species mixing compared to laminar flow [73, 100].

From an engineering perspective, the most critical aspect of turbulence is its enhanced diffusivity. Large eddies within turbulent flow dominate the energy transfer process, stretching smaller vortices and transferring energy down to the smallest scales, where it is eventually dissipated as heat. These large structures are influenced by the shape and properties of the body which is immersed in the flow, making it difficult to predict purely from local flow properties.

This non-local nature of turbulence means that it cannot be easily modeled or approximated in two dimensions, as the stretching and interaction of vortex lines require a full three-dimensional treatment [100].

In practical applications, such as in aerodynamics or geophysics, turbulent flow plays a significant role. For example, it enhances the mixing of air in the planetary boundary layer or influences the drag forces on vehicles. Even in biological systems, turbulence can be significant—such as in blood flow, where its presence may indicate health issues like defective heart valves. Despite its chaotic appearance, turbulence can be statistically analyzed, with time-averaged properties providing valuable insights into flow behavior.

As turbulent eddies decay, they transfer their kinetic energy down the cascade until it is dissipated by viscous forces at the smallest scales. This continual interplay between large-scale structures and fine-scale dissipation makes turbulence a challenging yet crucial phenomenon in fluid dynamics.

Turbulence is treated as a continuum phenomenon because the smallest turbulent scales are substantially larger than molecular scales. In turbulent flows, energy cascades from larger eddies to progressively smaller ones. Consequently, at the smallest scales, the energy inflow from larger eddies matches the rate at which these small eddies dissipate energy as heat. This principle is central to Kolmogorov's (1941) universal equilibrium theory. According to this theory, the dynamics at the smallest scales are primarily influenced by the energy transfer rate $\epsilon \approx -dk/dt$, where k is the turbulent kinetic energy, and the kinematic viscosity ν [100, 69]. To gain insight into turbulence, it's crucial to understand the role of eddies, which are coherent structures within the flow. Turbulent flows are essentially composed of a spectrum of eddies, ranging from large to medium and the smallest, also referred as Integral eddies, Taylor eddies and Kolmogorov eddies. These eddies interact dynamically across the flow, creating a complex and layered turbulence structure.

For a more quantitative analysis, turbulence is often examined using its energy spectrum. This involves decomposing the turbulence into wavenumber κ or wavelength $\lambda = 2\pi/\kappa$. The spectrum $E(\kappa)$ is affected by the size of the larger eddies l and the mean strain rate tensor $\boldsymbol{S}$. Additionally, because turbulence involves dissipation, $E(\kappa)$ also depends on viscosity ν and the energy transfer rate ϵ. At high Reynolds numbers, dimensional analysis indicates that the turbulent kinetic energy k can be described in terms of ϵ and a characteristic length scale l, which is fundamental to many turbulence models. A description of the Kolmogorov's energy spectrum is depicted in figure 2.3.

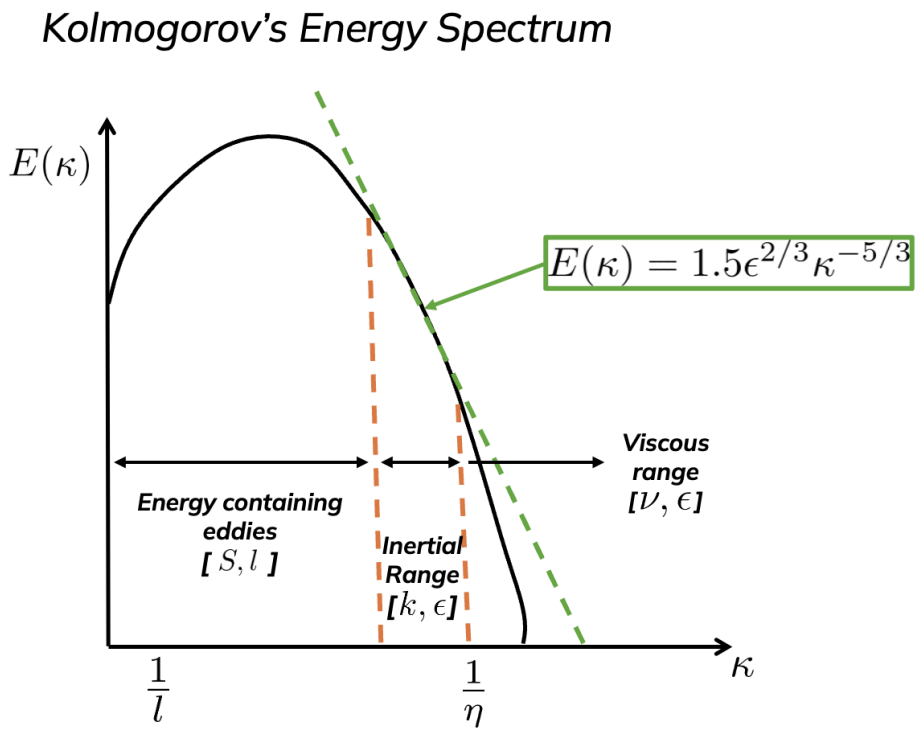


Figure 2.3: Turbulent energy spectrum - log-log scale

Despite the broad range of eddy sizes, their collective behavior can be described using characteristic scales length, time, and velocity that are useful for engineering estimates. Understanding these categories allows analysts to estimate key turbulence metrics such as eddy size, lifetime, and velocity.

Large eddies are inherently unstable due to their high energy, significant size, and the interactions with surrounding eddies that stretch and distort them in various directions. This instability causes large eddies to fragment into smaller eddies, which continues the cascade process. The intermediate-sized eddies, fall between the large integral eddies and the smaller Kolmogorov eddies. They display a hybrid behavior that bridges the characteristics of integral and Kolmogorov eddies.

Finally, the Kolmogorov eddies represent the smallest scale in the turbulence cascade. As these eddies become smaller, they eventually reach a scale where their size is comparable to or smaller than the viscous forces that dampen their rotational motion. Consequently, Kolmogorov eddies are at the end of the energy cascade. Their energy is

dissipated into the fluid through viscous shear, returning to its original source. Due to their small size and high vorticity, Kolmogorov eddies have a very short lifespan compared to Integral and Taylor eddies. Moreover, their timescale is much smaller than the mean flow timescale, making their motion largely independent of both the mean flow and larger eddies. In an equilibrium state of turbulence, the energy supplied to large eddies is balanced by the heat dissipated by Kolmogorov eddies.

Another key feature of turbulent flow is that, as we move perpendicularly from the wall toward the turbulent core, the flow can be divided into distinct regions, each with unique characteristics. Specifically, it can be separated into the viscous sublayer, buffer layer, log layer, and defect layer. These layers are typically distinguished using two dimensionless variables. The first is analogous to the Reynolds number and is defined as follows:

$$y^+ = \frac{yu^*}{\nu} \tag{2.3}$$

In equation 2.3, u^* is known as the shear velocity, and it depends on the wall shear stress tensor, defined as follows:

$$u^* \equiv \sqrt{\frac{\tau_w}{\rho}} \tag{2.4}$$

The second variables is the dimensionless velocity profile defined as the ratio between the mean velocity and the shear velocity, as follows:

$$u^+ = \frac{\overline{u}}{u^*} \tag{2.5}$$

Laminar flow persists in the viscous sublayer, where viscous forces dominate and the Reynolds number remains below the critical value. As y^+ increases, the flow enters the buffer layer, spanning $7 \leq y^+ < 30$. This region exhibits both laminar and turbulent characteristics, with a steep velocity gradient that maximizes turbulence production and generates the largest eddies, resulting in high turbulence intensity. Beyond the buffer layer, the log layer ($30 \leq y^+ \leq 700$) follows, characterized by a logarithmic velocity profile, commonly referred to as the "law of the wall". While the turbulence intensity here is lower than in the buffer layer, the log layer still significantly contributes to eddy formation. For $y+ > 700$, the turbulent velocity exceeds predictions from the log law, marking the defect layer. Together, the log and defect layers form the turbulent core of the flow.

This classification is crucial for turbulent flow simulations. To achieve accurate resolution of turbulent behavior, grids must be designed with small y^+ values. Additionally, this layer-based distinction informs the development of wall functions, which allow engineers to approximate flow behavior near the wall without the computational cost of highly refined grids.

2.2 RANS Modelling

All the complex structures and behaviors of turbulent flow can be predicted using the 3D Navier-Stokes equations, solved on a sufficiently fine mesh ($y^+ \ll 1$). This approach, known as Direct Numerical Simulation (DNS), resolves all scales of turbulence. However, as the Reynolds number increases, the computational cost becomes prohibitively expensive. Scientists sought alternative methods to reduce this cost, one of which involves solving only the larger scales of turbulence while modeling the smaller scales. This is the core concept of Large Eddy Simulations (LES), where the governing equations are derived by applying a filter to the N-S equations. Typically, the filter's cut-off length is proportional to the mesh element size. As a result, new terms, called subgrid-scale terms, appear in the equations and are a function of the mesh size. When the mesh size approaches zero, the subgrid terms also vanish, recovering the N-S equations as an asymptotic limit. Although LES is less computationally demanding than DNS, it remains quite expensive and is still mostly used for research purposes, with limited application in industry.

Turbulence is characterized by random fluctuations, which allows it to be treated statistically. This idea was first proposed by Osborne Reynolds in 1895, who suggested that fluid flow properties, such as velocity, could be viewed as the sum of a mean and a fluctuating component.

$$u(x,t) = \overline{u}(x) + u^{'}(x,t) \tag{2.6}$$

By applying the decomposition presented in Eq. 2.6 to the Navier-Stokes equations, one can derive the so-called Reynolds-Averaged Navier-Stokes (RANS) equations. Eq. 2.6 describes a time-averaged mean, which is suitable for turbulent flows that exhibit statistically steady behavior. Other forms of averaging, such as spatial or ensemble averaging, are also possible; however, the choice of averaging method does not affect the mathematical structure of the RANS equations. In this work, a compressible flow solver is used, making it more appropriate to apply Favre averaging, or density-weighted time averaging. The Favre-averaged quantities are denoted with a tilde, and the mean component is defined as follows:

$$\tilde{u} = \frac{\overline{\rho u}}{\overline{\rho}} \tag{2.7}$$

Introducing the Reynolds decomposition into the Navier-Stokes equations gives rise to what is known as the closure problem, which is directly related to the nonlinearity of the governing equations. The nonlinearity results in the emergence of momentum fluxes, which act as apparent stresses within the flow, named Reynolds stresses. These momentum fluxes, however, are not known a priori. When deriving equations for these stresses, additional unknowns are introduced, further complicating the system. The closure problem, therefore, involves generating enough equations to account for all these unknown variables.

The RANS equations (continuity and momentum equations) for a compressible flow read:

$$\frac{\partial \overline{\rho}}{\partial t} + \nabla \cdot (\overline{\rho}\tilde{\boldsymbol{u}}) = 0 \tag{2.8}$$

$$\frac{\partial \overline{\rho}\tilde{\boldsymbol{u}}}{\partial t} + \nabla \cdot (\overline{\rho}\tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}}) = -\nabla \overline{P} + \nabla \cdot (\overline{\boldsymbol{\tau}} - \overline{\rho \boldsymbol{u}' \boldsymbol{u}'}) \tag{2.9}$$

The unknown tensor $-\overline{\rho \boldsymbol{u}' \boldsymbol{u}'}$ that appears in the momentum equation 2.9 is referred to as the Reynolds stress tensor. It possesses several important properties, such as positive definiteness, and since it is symmetric, only six of its nine components are independent. These six unknowns pose a challenge in solving the equations governing turbulent flow. To address this, the Boussinesq hypothesis—proposed by Jacques Boussinesq in the late 19th century—provides a crucial simplification. This hypothesis suggests that turbulent shear stresses can be related to the mean velocity gradients, in much the same way molecular viscosity accounts for momentum transfer in laminar flow.

$$-\overline{\rho \boldsymbol{u}' \boldsymbol{u}'} \propto \nu_t S \tag{2.10}$$

where ν_t is the turbulent eddy viscosity. This formulation simplifies the complex interactions within turbulence, reducing them to an effective viscosity term, which allows for easier analysis and computation of turbulent flows. While the Boussinesq hypothesis facilitates bridging the macroscopic behavior of turbulent flows with the microscopic eddy interactions, it introduces some limitations. For instance, modeling turbulent viscosity as an isotropic quantity eliminates the capacity to capture the intricate structures of turbulence, especially in non-homogeneous or anisotropic flows. This isotropic assumption leads to errors in simulations of complex flow environments, such as turbomachinery, where secondary flows and directional variations in turbulence are significant. Models based on Pope's theory, for example, attempt to reintroduce some of the nonlinear behavior of turbulence to improve accuracy in these situations [68]. Despite these shortcomings, RANS models based on the Boussinesq hypothesis remain widely used in industrial applications. This is due not only to the reduced computational cost compared to more detailed turbulence models but also because in many practical engineering problems, only the mean flow characteristics are required. These models typically provide sufficient accuracy for calculating integral quantities like lift and drag coefficients in turbulent flows, which are of primary interest in industrial settings.

The next challenge lies in selecting an appropriate model for the turbulent eddy viscosity. Over the years, several models have been developed, which can be categorized into zero, one, and two-equation models.

2.2.1 Zero-equation models

The 0-equation models, also known as algebraic models, like Prandtl's mixing length model, are the simplest. In this approach, the turbulent eddy viscosity is determined based on a mixing length and the gradient of the mean flow, particularly in the wall-normal direction. While straightforward, these models rely heavily on case-specific assumptions and require prior knowledge of the flow characteristics, which limits their general applicability. Another example of an algebraic model is the one of Baldwin-Lomax [4] which can be applied only to attached, thin boundary layers.

2.2.2 One-equation models

To address the limitations of algebraic turbulence models, researchers began developing one-equation models. The key idea behind these models is to introduce a transport equation for a single turbulent quantity, typically the turbulent kinetic energy. One of the earliest and most famous examples of this approach is Prandtl's 1945 model, which solves for the turbulent kinetic energy. However, since the model only provides one turbulent characteristic, an additional assumption is required to calculate the turbulent viscosity. In Prandtl's model, a linear relationship is used for this purpose, which constitutes its primary limitation.

Other well-known one-equation models include the Baldwin-Barth model [3], and the Spalart-Allmaras model [86]. Both of these models simplify the turbulence modeling by introducing a transport equation for the turbulent viscosity ν_t directly, rather than relying on turbulent kinetic energy or other turbulent quantities. The Spalart-Allmaras model, in particular, has gained widespread use in aerodynamics and it is the model employed in this thesis. Therefore, a more detailed discussion of this model will be provided in the following chapter.

2.2.3 Two-equation models

Two-equation turbulence models are based on solving two transport equations for two key variables, which together are used to model the turbulent viscosity ν_t. Kolmogorov's original model from 1942 laid the foundation for these models by using two key variables: turbulent kinetic energy k and specific dissipation rate ω. In Kolmogorov's view, ω represents the rate at which turbulence dissipates in a unit volume over time. He considered ω to be a fundamental quantity applicable across all scales of turbulence, assuming equilibrium energy transfer among eddies.

All two-equation models share a common feature: one of the transport equations governs turbulent kinetic energy k, while the second equation differs, leading to two main categories of models: the $k-\epsilon$ and the $k-\omega$ models. The Standard $k-\epsilon$ (SKE) model of Launder and Sharma [45], is a widely used two-equation turbulence model. It is valued for its simplicity and robustness but has several limitations. It tends to be overly

dissipative, has difficulty accurately modeling flow separation, and exhibits issues with the correct representation of eddy scales and closure inconsistencies. To address some of these shortcomings, alternative versions of the $k-\epsilon$ model have been developed. The Renormalization Group (RNG) $k-\epsilon$ model [103] introduces a term that reduces dissipation, particularly improving performance in low-Reynolds number flows and flows with swirling motions. The Realizable $k-\epsilon$ (RKE) model [81] incorporates a realizability constraint, which ensures that the predicted stresses remain physically realistic, making it particularly effective for flows involving separation or swirling. Despite these improvements, both the RNG and RKE models share some of the same drawbacks as the SKE model, including challenges with accurately capturing eddy scales and closure inconsistencies.

The most notable $k-\omega$ models are those developed by Wilcox, with three versions existing: 1988, 1998, and 2006 [99, 101, 100].

Wilcox 1988: This version demonstrated good performance in predicting flows with adverse pressure gradients and separated flows. It tends to be more accurate near walls due to its use of the specific dissipation rate, ω, instead of the turbulent dissipation rate, ϵ, as seen in the $k-\epsilon$ model. However, it was sensitive to free-stream turbulence, often over-predicting turbulence levels in regions away from walls.

Wilcox 1998: The 1998 version introduced modifications to address the free-stream sensitivity that plagued the 1988 model, enhancing its robustness and accuracy in high-Reynolds-number flows. A significant addition was a cross-diffusion term that improved the interaction handling between turbulence and flow properties in complex boundary layer scenarios. While it represented an improvement, the 1998 model still exhibited some sensitivity to free-stream conditions, albeit significantly reduced.

Wilcox 2006: This model further refined the traditional $k-\omega$ approach by blending the $k-\omega$ behavior near the wall with a $k-\epsilon$-like approach in the outer boundary layer. This strategy aimed to mitigate free-stream sensitivity, a persistent issue in earlier $k-\omega$ formulations. A key feature of the 2006 model is the introduction of a limiter on the turbulent shear stress, which helps to prevent the unphysical growth of turbulent viscosity and excessive turbulent production in separated flows. Overall, Wilcox's 2006 model focuses on enhancing free-stream performance while effectively managing boundary layer flows, and maintaining accuracy near walls and in adverse pressure gradients. It requires a fine mesh near the wall, as it does not utilize wall functions, which can be seen as an advantage, with the first node of the mesh typically having $y^+ < 5$.

In 2003, Menter et al. [55] developed the Shear Stress Transport (SST) $k-\omega$ model. This model similarly blends $k-\epsilon$ and $k-\omega$ behavior, leveraging the strengths of both: the $k-\epsilon$ model performs well at high Reynolds numbers away from walls, while the $k-\omega$ model excels in low Reynolds numbers near boundary layers but is more sensitive to free-stream conditions. In this approach, the boundary layer is computed using the 1988 $k-\omega$ model, while the $k-\epsilon$ model is applied in the free-stream region. A blending function computes the asymptotic turbulent behavior between the two regions. Additionally, a stress limiter is implemented to account for the influence of mean strain rates on turbulent

kinematic viscosity, akin to the stress limiter in the 2006 Wilcox model.

Overall, while the SST $k - \omega$ model is slightly more complex than the 2006 Wilcox model, both are among the most widely used two-equation turbulence models today due to their excellent performance in solving turbulent flow characteristics.

2.2.4 Reynolds Stress Transport models

It's important to note that there are more sophisticated models that bypass the Boussinesq approximation, namely, the Reynolds Stress Transport models (RSM) [88, 38, 59]. These models offer an advanced approach to turbulence modeling by directly solving transport equations for the components of the Reynolds stress tensor. This allows for a more nuanced representation of turbulent flows compared to simpler models that rely on isotropic assumptions.

However, one of the main challenges with RSM models is the closure problem. Because they introduce more unknowns than equations, various closure assumptions must be employed to relate these stresses to mean flow quantities.

Additionally, the complexity of RSM models typically requires greater computational resources than two-equation models, making them less commonly used in practical applications due to their computational intensity.

2.3 Transition phenomena

Turbulence modeling is inherently complex due to the various approximations that must be made to capture the chaotic nature of turbulent flows. Among the most commonly used approaches are RANS models, which provide a time-averaged solution to the N-S equations. However, RANS models introduce the closure problem, where additional equations are needed to account for the unknowns introduced by averaging the turbulent fluctuations. To address this, many RANS models have been developed, each offering different levels of accuracy depending on the flow conditions. While these models provide good approximations, they still introduce errors, though typically small enough to capture the essential physical properties required for engineering design purposes. The most challenging scenarios to model, however, are the transition regimes, where the flow shifts from laminar to turbulent, and situations involving flow separation. In these cases, fully turbulent RANS models often fall short, as they are designed for fully developed turbulent flows and struggle to capture the intricate dynamics of transitional flows. Finding new models or improving existing ones to handle these scenarios remains an active area of research, as simulating transitional flows with both accuracy and efficiency is crucial but extremely difficult. The transition from laminar to turbulent flow can occur under several conditions, including natural transition, bypass transition, separated-induced transition, and reverse transition, each with distinct characteristics. The laminar boundary

layer can become destabilized by viscous instability waves, a phenomenon initially predicted by Prandtl and later confirmed mathematically by Tollmien. These disturbances are known as Tollmien-Schlichting (T-S) waves. Schlichting [80] offers a comprehensive explanation of this process, which is termed natural transition. As the Reynolds number based on momentum thickness reaches a critical threshold, the laminar boundary layer becomes susceptible to small disturbances in the flow. These disturbances evolve into two-dimensional T-S waves. As the instability intensifies, three-dimensional perturbations arise, forming loop vortices that generate large fluctuations in the flow. These fluctuations lead to the formation of turbulent spots, which grow and propagate downstream, eventually merging to form a fully developed turbulent boundary layer. This transition process has been thoroughly studied in low-turbulence environments, such as the flow over aircraft wings, where the laminar boundary layer gradually transitions into turbulence. However, the transition does not occur until the T-S waves become nonlinear, and inviscid mechanisms begin to amplify the disturbances, leading to three-dimensional flow structures. The turbulent spots are formed and expand within the surrounding laminar flow and eventually coalesce, transforming the entire boundary layer into a turbulent one. The figure 2.4 represents this process.

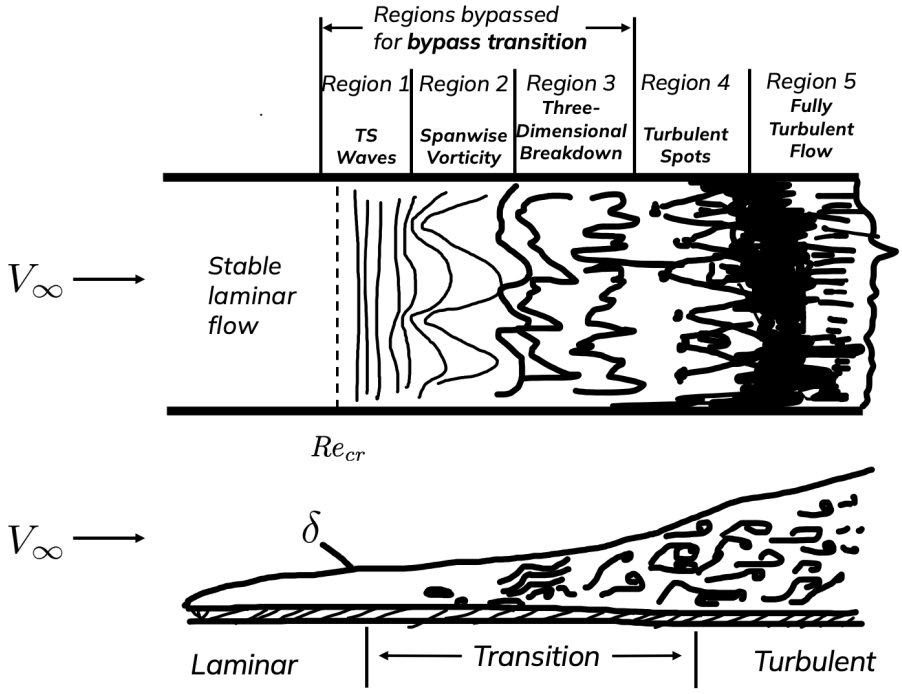


Figure 2.4: Natural transition process

Natural transition typically occurs when the freestream turbulence level is low (below 1%). However, when the turbulence level in the freestream is higher (above 1%), the initial stages of natural transition (as shown in figure 2.4) are bypassed, leading to a different mechanism known as bypass transition. Bypass transition can also occur in the presence of surface roughness. In this case, turbulent spots form directly within the boundary layer, with external disturbances driving the process. Unlike natural transition, where small disturbances grow gradually, bypass transition bypasses linear growth, focusing instead on the generation, growth, and downstream propagation of turbulent spots. From a practical perspective, bypass transition is often considered to begin when the skin friction deviates from the expected laminar values, which has been identified as the point where the first turbulent spots appear [52]. Before these spots form, the boundary layer remains laminar, though measurements indicate that this pre-transitional flow exhibits significant velocity fluctuations, often referred to as laminar fluctuations [52]. These fluctuations are thought to result from pressure waves in the turbulent freestream, which propagate into the boundary layer and, similar to T-S waves, cause the velocity fluctuations to grow. Once these fluctuations reach a critical level, turbulent spots form and propagate downstream, marking the onset of transition.

Separated-flow transition is a significant mechanism in various engineering applications, especially in gas turbines. When a laminar boundary layer detaches from a surface, the resulting free shear layer can undergo a transition to turbulence. Due to the increased mixing in the turbulent flow, this shear layer can reattach to the surface, forming what is known as a laminar separation and turbulent reattachment "bubble". This phenomenon often occurs in the presence of boundary layer disturbances like trip wires or in regions with adverse pressure gradients. In gas turbines, such transition is particularly common in overspeed regions, notably near the leading edge of an airfoil, both on the suction and pressure sides, or around the point of minimum pressure on the suction side. The length of these separation bubbles varies depending on the transition processes within the free shear layer. They are classified as long or short, based on their influence on the airfoil's pressure distribution. Long bubbles, particularly in low-turbulence environments, tend to maintain laminar flow for an extended duration, during which T-S instabilities have been detected. From an aerodynamic performance standpoint, long separation bubbles are undesirable, as they lead to increased flow losses and significant deviations in exit flow angles. In contrast, shorter bubbles promote earlier transition to turbulence, which can be strategically used to enhance flow control and improve overall performance. However, predicting whether a separation bubble will be long or short is challenging. Small changes in Reynolds number or the airfoil's angle of attack can cause the bubble to drastically change from short to long, a phenomenon known as "bursting" [52]. Separation-induced transition can also occur around an airfoil's leading edge, especially if the leading edge radius is small. The size of the leading-edge separation bubble is influenced by factors such as freestream turbulence intensity, leading-edge geometry, angle of attack, and the Reynolds number. Larger leading-edge separation bubbles result in a thicker downstream boundary layer, increasing the likelihood of flow separation due

to an adverse pressure gradient. A depiction of this bubble-like separation is shown in Figure 2.5.

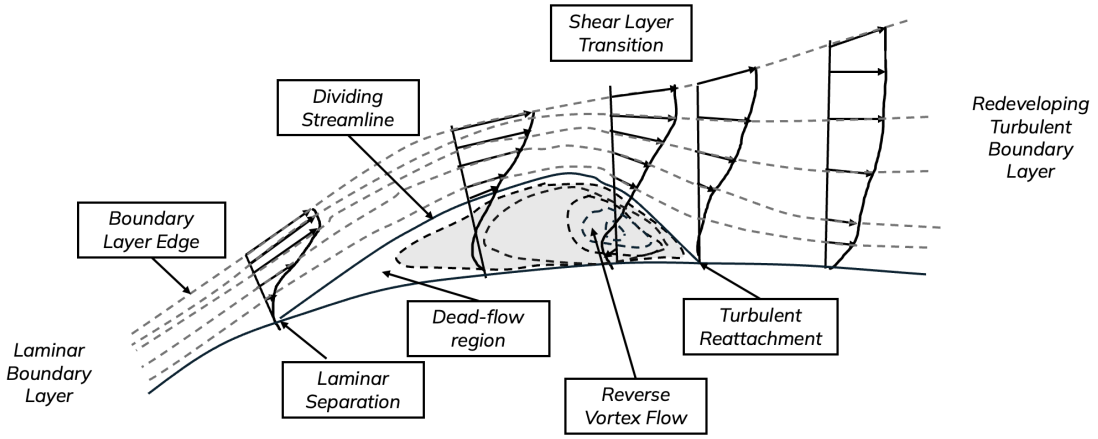


Figure 2.5: Separation-induced transition mechanism

The transition process can also occur in the reverse direction, where turbulent flow reverts to laminar flow, a phenomenon known as reverse transition or re-laminarization. This typically happens when there is significant acceleration, such as on the trailing edge pressure side of most airfoils or on the leading edge suction side of many turbine blades. The acceleration in these regions can be large enough to cause the turbulent boundary layer to relaminarize. However, if the acceleration decreases sufficiently, the flow can undergo re-transition, reverting back to turbulence. This dual process of re-laminarization followed by re-transition can occur as the flow adapts to changing conditions.

2.4 Transition Modelling

To address the transition phenomena, scientists have developed transitional RANS models that attempt to capture transition phenomena more accurately. One key aspect of transition is its intermittency, the flow does not smoothly transition from laminar to turbulent but rather switches back and forth depending on local conditions. For example, natural transition is driven by flow instability, where the flow evolves gradually as the Reynolds number increases, reaching a critical value that stabilizes the turbulent regime. However, transition can also occur abruptly due to external factors, such as an adverse

pressure gradient that forces laminar flow to become turbulent or a favorable pressure gradient that returns turbulent flow to a laminar state. Additionally, the geometry of the object can significantly influence transition. A sharp shape may induce flow separation, triggering the transition to turbulence. However, despite the knowledge on the phenomena and the advancements in the RANS model, transitional RANS models often fail to accurately predict such complex behaviors, resulting in erroneous estimates for key design parameters like drag or lift coefficients. The complexity of capturing transitional flows, especially under varying conditions, remains a significant challenge in turbulence modeling.

2.4.1 e^N Semi-Empirical Method

Initially, models for predicting transition were based on boundary layer linear stability theory. This allows for the prediction of transition by monitoring the amplification of small disturbances, the T-S waves. The linear stability theory decomposes the flow into a mean flow and a small disturbance superimposed on it. This approach assumes that the disturbances are small compared to the main flow quantities. The primary objective of stability theory is to assess whether disturbances will grow or diminish when subjected to a given mean flow. The flow is considered unstable if disturbances amplify and stable if they decay [80]. A significant application of linear stability theory is the semi-empirical e^N method, developed by van Ingen [92] and further refined by Smith and Gamberoni [85] in 1956. This method extends stability theory by concentrating on the energy amplification of disturbances within the flow. The fundamental concept involves assessing how these disturbances develop and amplify along the flow, represented by the factor e^N, where N denotes the amplification factor. The procedure consists of three sequential steps. The initial step involves calculating the laminar velocity and temperature profiles along the object of interest. In the second step, the local growth rates of unstable waves are determined for each of these profiles, which can be achieved by solving the local stability equations. Finally, the local growth rates are integrated along each streamline to ascertain the N factor. Transition is assumed to begin once the disturbance amplitude ratio e^N surpasses the critical N factor. Typical values can range from 7 to 9. A significant limitation of this method is that the initial amplitude of disturbances in the boundary layer is influenced by the external disturbance environment, which is typically governed by an unknown receptivity process. Consequently, the N factor at the onset of transition is not universal and must be calibrated through wind tunnel or flight tests. This method encounters challenges when applied to separated flow transitions, where the transition onset leads to boundary layer reattachment. In such cases, the laminar solution can be significantly separated and exhibit unsteady vortex shedding. Because a steady-state solution is often not available, local growth rates cannot be calculated using the local stability equations. In this scenario, the e^N method, which is based on linear stability theory, fails to account for transitions driven by nonlinear effects, such as bypass transition. In contemporary N-S simulations, the transition is generally modeled by activating

a turbulence model at the predicted transition point, as indicated by the e^N method. This approach facilitates a gradual increase in eddy viscosity, resulting in a smooth rather than abrupt transition in the properties of the viscous layer and creating a small computational transitional zone. However, this point transition method can introduce local disturbances due to significant reductions in displacement thickness near the transition point, leading to pronounced interactions between viscous and inviscid flows, which may notably influence flow characteristics upstream of the transition region. While these methods provide valuable insights into transitional phenomena, they often rely on linear approximations that may not fully capture the complex behaviors observed in practical applications. Furthermore, they tend to be incompatible with standard CFD techniques employed for complex geometries, as they necessitate prior knowledge of geometry and grid topology and involve numerous non-local operations, such as tracking disturbance growth along each streamline. These complexities pose significant challenges for integrating such methods into modern CFD approaches [44].

2.4.2 Correlation-based Method

Another approach to modeling transition involves using correlated empirical formulas. A notable investigation into transition was conducted by Abu-Ghannam and Shaw in 1980 at the University of Liverpool's Aerodynamic Laboratory [1]. Their experiments carried out in a low-speed wind tunnel, used six turbulence grids made from steel or wooden bars to generate turbulence levels between 0.5% and 5% at the leading edge of a flat plate. They developed a model that links boundary layer parameters to external factors, such as turbulence intensity and both adverse and favorable pressure gradients. Specifically, they established a correlation for the onset of transition based on the turbulence level and pressure gradient data. Transition initiation was analyzed by calculating the momentum thickness Reynolds number, using the following equations:

$$Re_{\theta S} = 163 + \exp\left(F(\lambda_\theta) - \frac{F(\lambda_\theta)}{6.91}\tau\right) \tag{2.11}$$

$$F(\lambda_\theta) = \begin{cases} 6.91 + 12.75\lambda_\theta + 63.64(\lambda_\theta)^2 & : \lambda_\theta < 0 \\ 6.91 + 2.48\lambda_\theta + 12.27(\lambda_\theta)^2 & : \lambda_\theta > 0 \end{cases} \tag{2.12}$$

where λ_θ represents the pressure gradient parameter and it is defined as follows:

$$\lambda_\theta = \frac{\theta^2}{\nu}\frac{dU}{dx} \tag{2.13}$$

In Equation 2.13, θ is the momentum thickness, whereas ν represents the kinematic viscosity and dU/dx is the streamwise velocity gradient.

They also derived a correlation to predict the end of transition. Building on the work of Dhawan and Narasimha (1958) [17], who defined a transition length Reynolds number

Re_{L_γ} as a function of the Reynolds number Re_{xs} at the start of transition for a zero pressure gradient scenario, the following equation was used:

$$Re_{L_\gamma} = 5 \cdot Re_{xs}^{0.8} \tag{2.14}$$

Here, Re_{L_γ} is based on the transition length L_γ, which corresponds to the region where the intermittency factor γ increases from 0.25 to 0.75. Moreover, Dunham's (1972) [19] research showed that the total transition length L is related to L_γ ($L = 3.36L_\gamma$), leading to the final equation for the Reynolds number at the end of transition:

$$Re_{xe} = Re_{xs} + 16.8 \cdot Re_{xs}^{0.8} \tag{2.15}$$

Additionally, they provided a relation for the intermittency factor γ, which was measured under different conditions. Notably, no significant effect of pressure gradient was observed. From these findings, along with other experiments involving various pressure gradients and turbulence levels, they derived the following relationship between γ and η:

$$\gamma = 1 - \exp(-5\eta^3) \tag{2.16}$$

where η represents the non-dimensional distance from the start of the transition over the transition length. It is defined as follows:

$$\eta = \frac{X - X_S}{X_E - X_S} \tag{2.17}$$

In Equation 2.17, X_S is the starting point for transition, while X_E is the end point. Finally, X is an arbitrary point in the transition region. The correlations developed by these researchers were able to predict the evolution of key boundary layer parameters, such as the momentum thickness θ, shape factor H, skin-friction coefficient CF, and intermittency factor γ. These quantities were found to be related to the non-dimensional distance from the start of transition.

2.4.3 $\gamma - Re_\theta$ Model

The methods described above address the limitations of RANS models, which are unable to capture the linear disturbance growth characteristic of transition phenomena. However, these methods are not fully compatible with general-purpose CFD approaches. The primary challenge lies in the need for prior knowledge of the geometry and grid topology required by these models. Additionally, another limitation is the use of non-local operations, which, while possible to implement in CFD codes, are highly inefficient and time-consuming. The complexity arises from the fact that CFD codes often employ mixed elements and are typically executed in parallel, meaning boundary layers may be divided and computed across different processors, making search or integration algorithms impractical. For a transition model to be fully compatible with CFD codes, it must be formulated locally. One of the most well-known and widely used transition models

is the $\gamma - Re_\theta$ model developed by Langtry and Menter [44]. The authors emphasized that their model does not aim to replicate the detailed physics of the transition process. Instead, its purpose is to provide a framework for incorporating correlation-based models into general-purpose CFD methods, a technique referred to as local correlation-based transition modeling. The key innovation of this model is that it eliminates the need to integrate the boundary-layer velocity profile to determine the onset of transition [56]. Instead, it introduces the concept of the vorticity Reynolds number to establish a connection between the transition onset Reynolds number, derived from empirical correlations, and local boundary-layer parameters. The vorticity Reynolds number, or the strain-rate Reynolds number, used in this model is defined as follows:

$$Re_v = \frac{\rho y^2}{\mu} \left| \frac{\partial u}{\partial y} \right| \tag{2.18}$$

while the momentum thickness Reynolds is defined as follows:

$$Re_\theta = \frac{max(Re_v)}{2.193} \tag{2.19}$$

In the presence of strong pressure gradients, the relationship between the momentum-thickness Reynolds number and the vorticity Reynolds number, as described by Eq. 2.19, is altered due to changes in the boundary-layer profile shape. This variation affects the accuracy of the transition prediction. Indeed, Langtry and Menter in [44] present a plot showing the relative error between the maximum vorticity Reynolds number and the momentum-thickness Reynolds number as a function of the boundary layer shape factor H.

The result shows that the vorticity Reynolds number increases with the shape factor, providing a useful indicator to predict separation-induced transition. This feature makes these models superior to conventional low-Reynolds-number models, as they inherently contain information about the boundary layer thickness.

As the name suggests, the $\gamma - Re_\theta$ transition model is based on a transport equation for the intermittency factor γ, which is used to locally trigger transition by modifying the production term of the underlying RANS model. In their work [44], Langtry and Menter coupled this transport equation with the $k - \omega$-SST model. A brief description of the transport equation is provided below:

$$\frac{\partial \rho \gamma}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \gamma) = P_\gamma - E_\gamma + \nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_f} \right) \nabla \gamma \right] \tag{2.20}$$

For simplicity, I will not provide a detailed description of each term, just the most important one. The production term P_γ, which governs the generation of the intermittency factor, depends on the strain-rate tensor, as well as two empirical correlations: one controlling the length of the transition and the other determining the onset of transition. The production term is defined as follows:

$$P_\gamma = F_{length} c_{a1} \rho S \left[\gamma F_{onset}\right]^{0.5} (1 - c_{e1}\gamma) \tag{2.21}$$

The destruction or relaminarization source term is instead described below:

$$E_\gamma = c_{a2} \rho \Omega \gamma F_{turb} (c_{e2}\gamma - 1) \tag{2.22}$$

This destruction term ensures that the intermittency remains close to zero in the laminar boundary layer. When the F_{onset} function is no longer satisfied, the eq. 2.22 allows the model to predict re-laminarization. Instead of the strain rate S, the vorticity Ω is used in the destruction term to avoid the destruction of intermittency in the freestream due to freestream strain rates.

In addition to the transport equation for intermittency, the $\gamma - Re_\theta$ model solves a second transport equation for the transition onset momentum-thickness Reynolds number. The authors' idea was to use an empirical correlation to calculate $Re_{\theta t}$ in the freestream and to allow the freestream value to diffuse into the boundary layer. Finally, the transport equation takes a non-local empirical correlation and transforms it into a local quantity. The local quantity is then used to predict the transition length, as well as the critical Reynolds number. The transport equation for the transition onset momentum-thickness Reynolds number is presented below:

$$\frac{\partial \rho \tilde{Re}_{\theta t}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \tilde{Re}_{\theta t}) = P_{\theta t} + \nabla \cdot \left[\sigma_{\theta t} (\mu + \mu_t) \nabla \tilde{Re}_{\theta t}\right] \tag{2.23}$$

In Equation 2.23, the source term $P_{\theta t}$ is built to force the transported scalar $\tilde{Re}_{\theta t}$ to match the local value of $Re_{\theta t}$, which is evaluated with an empirical correlation. The source term is defined as follows:

$$P_{\theta t} = c_{\theta t} \frac{\rho}{t} (Re_{\theta t} - \tilde{Re}_{\theta t})(1 - F_{\theta t}) \tag{2.24}$$

The blending function $F_{\theta t}$ in Equation 2.24, allows to turn off the source term inside the boundary layer which forces the $\tilde{Re}_{\theta t}$ to diffuse in from the freestream. It is worth noting that the authors adjusted the local intermittency value γ to account for separation-induced phenomena. The original model struggled to predict the correct position of turbulent reattachment, often placing it too far downstream. To address this issue, the key idea was to allow the local intermittency to exceed 1 when the laminar boundary layer separates. This adjustment leads to a significant increase in the production of turbulent kinetic energy, promoting earlier reattachment of the flow.

The $\gamma - Re_\theta$ model is one of the most widely used models for simulating turbulent transition, known for delivering accurate results and being compatible with CFD computations. It is based on transport equations. However, the authors emphasize that this model is not designed to capture the underlying physics of the transition process. Instead, it should be seen as a framework for incorporating transition correlations into general-purpose CFD methods.

2.4.4 Laminar Kinetic Energy Model

An alternative approach to modeling transition, distinct from the widely-used $\gamma - Re_\theta$ model, is the laminar kinetic energy (LKE) model, which focuses on capturing the underlying physics of the transition process rather than relying on empirical correlations typically embedded in transport equations.

The core idea behind the LKE model is to represent the pre-transitional rise of fluctuations within transitional boundary layers and their subsequent breakdown into turbulence, as first emphasized by Mayle and Schulz [52]. To describe this phenomenon, the authors introduced an additional equation to model the evolution of the energy associated with these pre-transitional fluctuations, termed the laminar kinetic energy equation. They identified a mechanism that establishes the effective frequency and turbulence level in the freestream, both critical for generating and amplifying the fluctuations. These fluctuations increase skin friction and heat transfer in the pre-transitional region, eventually leading to bypass transition as the streamwise fluctuations deteriorate into turbulence. It is crucial to note that these streamwise fluctuations differ from conventional turbulence.

Pacciani et al. [64] also applied this LKE approach to investigate transitional flows in low-pressure gas turbines, particularly focusing on transition induced by laminar separation. Their model was based on the low-Reynolds-number Wilcox (1998) $k - \omega$ model, augmented with an additional laminar kinetic energy equation. This formulation includes coupling terms between the laminar and turbulent kinetic energy equations, which play a key role in the transition process by facilitating energy transfer from pre-transitional fluctuations to the turbulent field.

Moreover, Ferrero [23] found that the 1998 Wilcox $k - \omega$ model performs well when combined with the LKE model. Specifically, the 1998 version underpredicts turbulent kinetic energy, and the introduction of the LKE equation compensates for this missing contribution. In contrast, the 1988 and 2006 versions of the $k - \omega$ model tend to overpredict turbulent kinetic energy, which, when further augmented by the LKE equation, leads to poorer agreement with experimental data.

The turbulent and laminar kinetic energy equations for a compressible flow are expressed as follows:

$$\frac{\partial \rho k_l}{\partial t} + \nabla \cdot (\rho k_l \boldsymbol{u}) = \rho \nu_l S^2 - 2\mu \frac{k_l}{d^2} + \nabla \cdot (\mu \nabla k_l) - \rho R \tag{2.25}$$

$$\frac{\partial \rho k}{\partial t} + \nabla \cdot (\rho k \boldsymbol{u}) = P_k - \rho \beta^* k \omega + \nabla \cdot [(\mu + \sigma_k \mu_t) \nabla k] + \rho R \tag{2.26}$$

In equation 2.25, the production term is governed by the "laminar eddy viscosity" ν_l, which itself is a function of the laminar kinetic energy k_l and the vorticity thickness δ_Ω. The laminar eddy viscosity ν_l plays a crucial role in describing how energy is transferred within the laminar boundary layer during the pre-transitional phase, with the vorticity thickness δ_Ω helping to characterize the spatial variation of the velocity field in this region. Together, these parameters enable the model to account for the production and

evolution of laminar kinetic energy during the early stages of transition.

$$\nu_l = C_1 f_1 \sqrt{k_l} \delta_\Omega \tag{2.27}$$

To enhance the transfer of laminar kinetic energy to the turbulent field, the term R is introduced. This term appears in both the laminar and turbulent kinetic energy equations but with opposite signs, representing the exchange of energy between the two fields. The term R is a function of k_l, the laminar kinetic energy, and is defined as follows:

$$R = C_2 f_2 \omega k_l \tag{2.28}$$

In equation 2.28, the term f_2 acts as a damping function that controls the onset of the energy transfer process from laminar kinetic energy to the turbulent field. This function modulates the influence of R, ensuring that the transfer only occurs when certain conditions, such as the presence of significant laminar kinetic energy fluctuations, are met. By adjusting f_2, the model can regulate the initiation and magnitude of this energy exchange, preventing premature or excessive energy transfer during the transition process.

$$f_2 = 1 - e^{-\Psi/C_3} \quad \Psi = \max(0, R_y - C_4) \tag{2.29}$$

From equation 2.29, it can be observed that the damping function f_2 depends on R_y, which is defined as a local Reynolds number based on the turbulent kinetic energy ($R_y = \sqrt{k} d/\nu$). Once R_y surpasses a certain threshold, denoted by C_4, the transfer of laminar kinetic energy to the turbulent flow begins. This threshold mechanism ensures that the energy transfer only occurs when the local turbulence has developed sufficiently, as indicated by the value of R_y.

2.5 Enhancing Turbulence Model with Machine Learning

The need for multiple turbulence models stems from the fact that each is designed to perform well under specific scenarios, such as boundary layer flows, separated flows, or transitional flows. The widely-used two-equation models, such as the $k-\epsilon$ and $k-\omega$ models, incorporate several empirical coefficients that need to be calibrated for each specific study case. This calibration process adds another layer of complexity, requiring expert knowledge and often extensive trial and error. When it comes to predicting the transition from laminar to turbulent flow, the modeling challenge becomes even more intricate. In previous paragraphs, traditional approaches to modeling transition have been outlined, starting from empirical correlation-based methods to those that attempt to mimic the intermittency behavior associated with transition using transport equations. More complex models also exist, which try to incorporate the underlying physics of the transition phenomenon. Each of these methods comes with its limitations, such as

difficulties in predicting the exact location of transition onset or handling flow separation and reattachment. An example is the study by Choudhry et al. [15] where the $\gamma - Re_\theta$ model and the LKE model are compared on the low Reynolds NACA0021 airfoil.

Recently, turbulence modeling has increasingly embraced data-driven methods, leveraging machine learning algorithms to enhance the accuracy of conventional models. This need arises due to the limitations of traditional models like RANS, which, while computationally efficient, rely on empirical approximations and struggle to capture the complexity of turbulent flows, particularly in transitional regimes. Machine Learning (ML) offers a promising approach to address these challenges by using high-fidelity data to improve predictions. These advancements have gained attention in the CFD community, as they offer potential cost reductions and enhanced prediction capabilities [10, 30, 94].

According to Beck et al. [5], data-driven turbulence modeling generally follows three key approaches:

- **Parameter Estimation:** Machine learning algorithms are employed to fine-tune parameters within existing turbulence models, particularly targeting the Reynolds stress tensor and the Boussinesq hypothesis [68]. This method incorporates neural networks enhanced with Galilean invariance to improve flexibility when addressing complex fluid flows [47, 36, 109, 95].

- **Replacement of Closure Models:** Traditional closure models are replaced with fully connected neural networks (FCNNs) or architectures using gated recurrent units (GRUs). These networks are specifically designed to resolve stability issues in long-term simulations [7, 54, 43].

- **Substitution of PDE Models:** In some cases, the entire system of Partial Differential Equations (PDEs) governing turbulence is substituted with machine learning techniques, such as autoencoders or Physics-Informed Neural Networks (PINNs) [60, 70, 22, 37]. These methods aim to directly model the fluid dynamics without relying on traditional PDE formulations.

A recent and comprehensive review by Paola Cinnella explores the growing use of data-driven models to enhance turbulence modeling in CFD [16]. Cinnella categorizes the applications of ML in turbulence modeling into three main approaches:

- **Data-assimilation models:** These models identify corrective fields for RANS transport equations or Reynolds stresses using ML algorithms. The corrective fields are parameterized as functions of flow features, allowing more accurate predictions in new flow conditions. This approach adjusts RANS models to improve their predictions while preserving the model's overall structure [21, 27].

- **Direct learning of Reynolds stress corrections:** In this approach, corrections to the Reynolds stress field are learned directly from high-fidelity DNS or LES data. This allows RANS models to leverage detailed simulations to improve their accuracy, bypassing the need for empirical adjustments [47, 14].

- **Model-consistent approaches:** Here, the corrective fields are learned directly within the RANS framework through backpropagation or other optimization algorithms. This ensures that the corrections are learned in a way that is consistent with the model's physics. These methods use either gradient-based or gradient-free optimization techniques, depending on the complexity of the problem. [89, 25]

The review also distinguishes between different types of ML turbulence models based on their interpretability:

- **Black-box models:** These models use machine learning algorithms such as Neural Networks (NN), Random Forests (RF), or Gaussian Processes (GP) to learn unclosed terms like Reynolds stresses. As the name suggests, these models act as black boxes, offering no direct insight into the flow physics but providing high flexibility and predictive power. The lack of interpretability is a key limitation of these models.

- **Grey-box models:** In this category, machine learning is employed to refine specific components of the RANS model, including aspects like eddy viscosity or Reynolds stresses. This approach enhances accuracy while maintaining a degree of physical interpretability. A notable framework within this category is the Field Inversion and Machine Learning (FIML), which will be discussed in detail later, as it forms a foundational aspect of this PhD work.

- **Open-box models:** These models prioritize physical transparency by formulating corrections in easily interpretable mathematical expressions. Genetic Expression Programming (GEP) is commonly employed in this context. To ensure that the corrections remain physically meaningful and straightforward, techniques such as feature engineering and sparsity constraints are frequently applied. While these models offer greater interpretability, they tend to be less flexible than black-box models and may face challenges when scaling to high-dimensional problems.

Cinnella further classifies ML-driven turbulence models based on their training strategies:

- **A priori training:** This approach involves training machine learning models on high-fidelity data obtained from DNS or LES. The models are designed to predict corrections for RANS models by leveraging fully resolved turbulent properties, such as Reynolds stresses. However, this method may encounter issues related to inconsistencies between DNS and RANS fields, a challenge often referred to as the feature mismatch problem. Since the machine learning model is trained exclusively on DNS data, its predictions might not correspond well to the features present in RANS simulations, particularly because RANS models utilize approximations for turbulent quantities that do not have a direct comparison with DNS results.

- **A posteriori or model-consistent training:** This approach involves training the machine learning model directly within the RANS framework. The unclosed terms in the turbulence model are optimized through a "CFD-in-the-loop" process, where the output of the RANS model is compared to high-fidelity data. During this iterative process, the model parameters are adjusted to minimize discrepancies between the model predictions and the reference data. However, due to the highly non-linear nature of the RANS equations and the non-convex landscape of the loss function, optimization techniques, especially gradient-based ones, can become stuck in local minima, complicating the pursuit of global convergence. While gradient-free algorithms can circumvent this problem, they tend to be slower and more computationally intensive.

2.5.1 Field Inversion and Machine Learning

The literature review of various machine learning algorithms applied to turbulence modeling highlights their respective advantages and disadvantages. Among these approaches, the FIML method stands out as particularly intriguing due to its model-consistent nature. Moreover, as pointed out in the previous paragraph, FIML can be seen as a grey-box model. Unlike other methods, FIML does not require specific high-fidelity data; it can utilize any available high-fidelity or experimental data.

The initial phase of the algorithm focuses on an optimization problem aimed at determining a "correction field". This correction field is used as an additional parameter intended to enhance the existing turbulence model. The subsequent phase employs machine learning techniques to identify a suitable correlation between this correction field and fluid flow characteristics.

The FIML paradigm has shown great promise as a data-driven approach. Various studies underscore its effectiveness across different applications. For instance, Fidkowski [28] utilized FIML in gradient-based shape optimization to replace traditional RANS models, while Yang and Xiao [104] enhanced a four-equation transition model. Furthermore, Ferrero et al. [24, 27] applied FIML to improve RANS modeling for turbomachinery flows, and Brenner et al. [9] employed it in incompressible turbulent flow to directly adjust turbulent viscosity. The primary objective of this PhD work was to implement the FIML technique within a compressible CFD solver and to analyze its advantages and disadvantages. This approach was specifically applied to augment the RANS model, with a focus on the Spalart-Allmaras model due to its simplicity and single-equation formulation, which minimizes the number of parameters involved. This approach was tested on various scenarios, including a transitional flat plate, a backward-facing step, and two low-Reynolds number airfoils (details to be provided later). An analysis of different goal functions was conducted, alongside an investigation into multi-objective strategies. Additionally, a method was explored to bypass the identification of input variables in the second phase by employing an alternative ML algorithm. Further details will be elaborated in the upcoming chapters.

Chapter 3

Spalart-Allmaras

3.1 Original Spalart Allmaras Model

To leverage the FIML framework, the Spalart-Allmaras turbulence model was selected. Algebraic models were ruled out due to their limitations, as they require prior knowledge of the test case for calibration, making them inadequate for flexible simulations. On the other hand, two-equation models require the introduction of an additional differential equation, which can lead to increased numerical complexity [100]. Moreover, there are cases in which one-equation models yield comparable or even superior accuracy compared to two-equation models. A notable example is discussed in [90]. The objective here, however, is to develop a CFD-data-driven framework that can predict transitional phenomena without the limitations of overly specialized models. The Spalart-Allmaras model strikes the right balance between simplicity and adaptability for this purpose.

The original Spalart-Allmaras model was introduced in 1992 [86], where Spalart et al. provided a detailed explanation of the development and rationale behind each term in the equation. In this section, I will provide a concise summary of their reasoning while preserving the essential meaning of each component. The development of the Spalart-Allmaras one-equation model began by addressing some key disadvantages of two-equation models. First, the authors pointed out that two-equation models typically require finer grids near the wall, which increases computational cost and complexity. These models also involve strong source terms that can degrade convergence, and they require non-trivial upstream and freestream conditions for turbulence variables. Additionally, near-wall issues often necessitate the use of wall functions, which are cumbersome and lose their validity in critical scenarios like flow separation. The Spalart-Allmaras model was constructed by adhering to several fundamental principles: Galilean invariance, the use of mean flow field variables, and dimensional analysis to accurately model the turbulent eddy-viscosity ν_t. In turbulence modeling, the governing equations are characterized by the presence of source terms, specifically production and destruction terms. These source terms play critical roles in capturing the complex dynamics of

turbulence. The production term is responsible for the generation of turbulence, while the destruction term accounts for its dissipation. Together, they describe how turbulent kinetic energy is created, sustained, and eventually dissipated within the flow. The production term in turbulence equations typically represents the conversion of mean flow energy into turbulent energy. It quantifies how instabilities in the flow, such as velocity gradients (related to shear stresses), lead to the growth of turbulence. For example, near the boundary layer of an object, strong shear gradients generate turbulent eddies, which is reflected in the production term. Spalart et al. formulated the production term based on vorticity to effectively represent the generation of turbulence. Since the turbulent viscosity ν_t is as a scalar quantity, the production term utilizes the magnitude of the vorticity. This formulation links the turbulence production directly to the shear present in the flow. The resulting production term is expressed as follows:

$$P = c_{b1} S \nu_t \tag{3.1}$$

In Equation 3.1, S denotes the vorticity magnitude and the constant c_{b1} was calibrated on in-homogeneous flows yielding values between 0.13 and 0.14, specifically $c_{b1} = 0.1355$. In eddy viscosity models, a diffusion term is also included to represent the spreading of turbulent quantities from regions of high concentration to regions of low concentration. This diffusion process accounts for the movement of turbulent energy, and it typically depends on the sum of material viscosity and eddy viscosity, reflecting both molecular and turbulent transport effects. The diffusion term in these models is generally written as the spatial variation of the transport of the turbulent variable. Mathematically, it is expressed in the following form:

$$\frac{1}{\sigma} \nabla \cdot (\nu + \nu_t) \nabla \nu_t \tag{3.2}$$

However, Spalart et al. also introduced a non-linear term involving the square of the gradient of the turbulent eddy viscosity to account for additional effects that are not captured by the linear diffusion alone. The non-linear term is written as follows:

$$\frac{c_{b2}}{\sigma} (\nabla \nu_t)^2 \tag{3.3}$$

This non-linear term was introduced to control the spreading of the wake at the edge of the turbulent region. Specifically, it helps to manage the extent of turbulence diffusion into the surrounding flow, ensuring that the wake region is accurately represented without excessive spreading. To calibrate this term with experimental data, Spalart et al. identified a value for the model constant c_{b2} equals 0.622, which governs the influence of this additional non-linear contribution. Importantly, this term is split from the previous diffusion term and is treated as a source term in the governing equations. Finally, the destruction term in the Spalart-Allmaras model is crucial for modeling the damping of turbulence near solid boundaries. This damping occurs primarily due to two mechanisms. Near the wall, pressure fluctuations in the turbulent flow are restricted, or

"blocked," which results in reduced turbulence intensity. This effect, known as inviscid blocking, occurs because the wall imposes a no-penetration condition, preventing the large-scale turbulent structures from fully developing in the wall-normal direction. In addition to inviscid blocking, viscous damping further diminishes turbulence near the wall. The high viscosity in the boundary layer close to the wall acts to dissipate the smaller-scale turbulent structures effectively, reducing the turbulent kinetic energy. The destruction term is responsible for the inviscid block and it is written as follows:

$$D = c_{w1} f_w \left(\frac{\nu_t}{d}\right)^2 \tag{3.4}$$

In the equation 3.4, the variable d represents the distance from the wall, indicating that the destruction term becomes increasingly significant as the distance to the wall decreases. This means that turbulence dissipation is more intense near the wall, where the interaction between turbulent eddies and the wall's no-slip condition leads to enhanced damping effects. The constant c_{w1} is determined to ensure a balance between the production and diffusion terms ($c_{w1} = c_{b1}/k^2 + (1 + c_{b2})/\sigma$). The function f_w is incorporated to modulate the decay of the destruction term specifically in the outer region of the boundary layer. This function helps to ensure that the turbulence dissipation reduces appropriately as we move away from the wall, thereby reflecting the different dynamics at play in the free stream compared to the near-wall region. Figure 3.1 shows the behavior of this function.

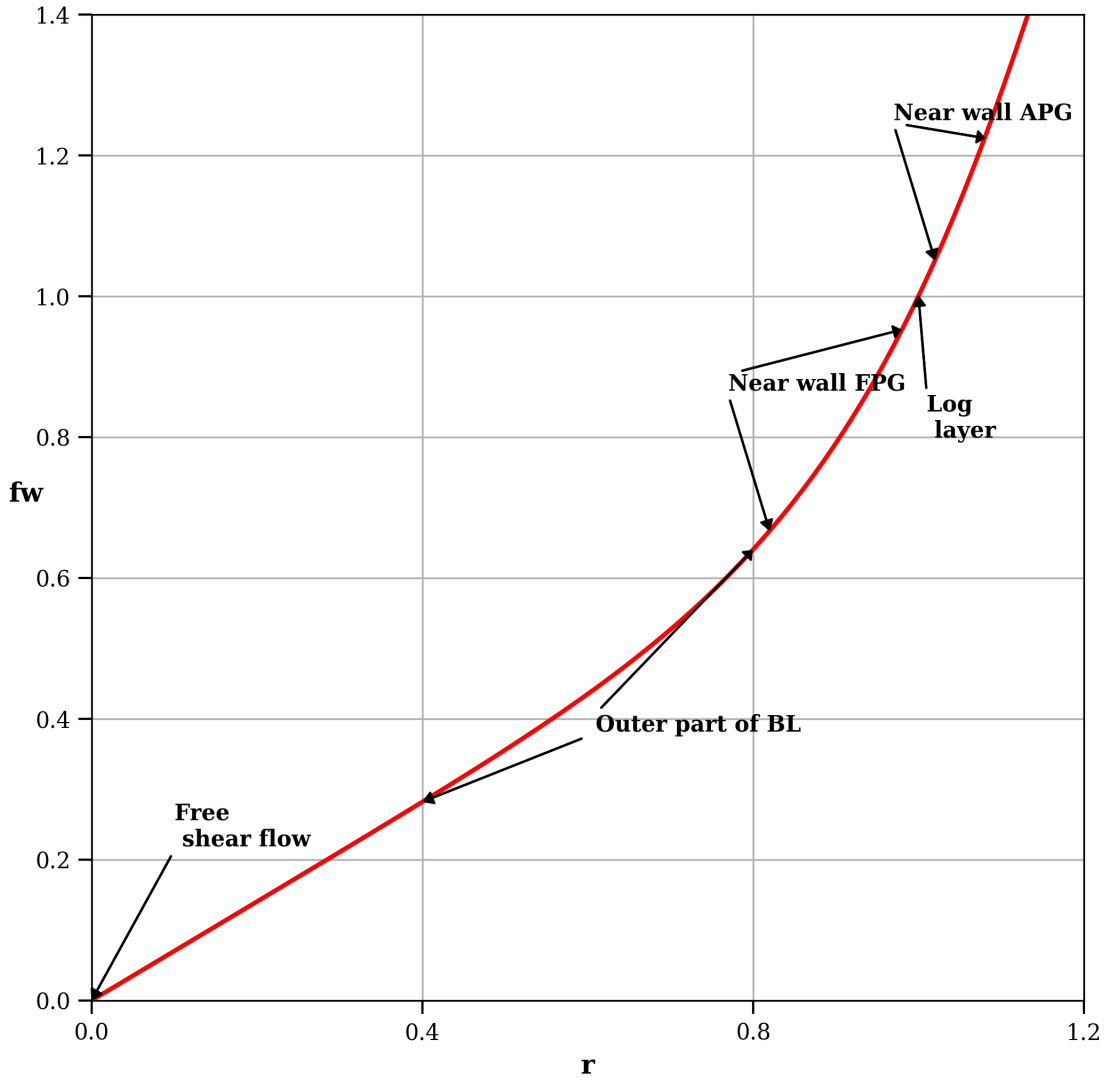


Figure 3.1: f_w function with respect to r. APG: adverse pressure gradient; FPG: favorable pressure gradient.

The formulation of the function f_w was influenced by traditional algebraic turbulence models, where the concept of mixing length is crucial, particularly near the wall. In algebraic models, the mixing length characterizes the scale of turbulence and varies significantly with distance from the wall being smaller near the wall and increasing farther away. The equations needed to build f_w are shown in the following:

$$f_w = g\left(\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right)^{1/6}, \quad g = r + c_{w2}(r^6 - r), \quad r = min\left(\frac{\tilde{\nu}}{\tilde{S}^2 k^2 d^2}, 10\right) \tag{3.5}$$

The authors calibrated c_{w2} to ensure that the skin-friction coefficient in a flat-plate

boundary layer matched empirical data, resulting in a value of $c_{w2} = 0.3$ for $\sigma = 2/3$. They also selected a reasonable value for c_{w3}, setting it to 2.

For the transported quantity, the authors chose a variable that behaves linearly near the wall. Instead of using the turbulent eddy viscosity directly, they used $\tilde{\nu}$ to represent the turbulent viscosity-like variable. This choice provides significant benefits for numerical solutions, as it allows the model to achieve accurate results without requiring a finer computational grid compared to what an algebraic model would demand. To achieve this desired behavior near the wall, they considered the classical law of the wall and formulated near-wall damping functions that are consistent with known empirical results. The new variable $\tilde{\nu}$ maintains a value of kyu_τ all the way to the wall, unlike the eddy viscosity ν_t, which only equals kyu_τ in the logarithmic layer. This approach allows the model to maintain more consistent near-wall behavior, simplifying the treatment of the boundary layer and improving the overall robustness of the numerical solution. To represent this behavior the authors build a function f_{v1} to retrieve the turbulent eddy-viscosity as follows:

$$\nu_t = \tilde{\nu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \tag{3.6}$$

where $\chi = \tilde{\nu}/\nu$ and $c_{v1} = 7.1$. The production term also needs attention. The vorticity magnitude S is replaced with $\tilde{S}$.

$$\tilde{S} = S + \frac{\tilde{\nu}}{k^2 d^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{3.7}$$

The function f_{v2} constructed, just like f_{v1}, so that $\tilde{S}$ maintains its log-layer behavior ($\tilde{S} = u_\tau/(ky)$) al the way to the wall. Finally, taking compressible flow into account, the transport equation is modified accordingly to accommodate the variations in density resulting in:

$$\frac{\partial \rho \tilde{\nu}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \tilde{\nu}) = \rho (P - D) + \frac{1}{\sigma} \nabla \cdot (\rho(\nu + \tilde{\nu}) \nabla \tilde{\nu}) + \frac{c b_2}{\sigma} \rho (\nabla \tilde{\nu})^2 - \frac{1}{\sigma} (\nu + \tilde{\nu}) \nabla \rho \cdot \nabla \tilde{\nu} \tag{3.8}$$

The final term in equation 3.8 represents a cross-diffusion term, which arises from incorporating the density variation within the diffusion term of the equation. The authors suggest that the version without considering density variation can be applied to both incompressible and compressible flows. However, for the sake of maintaining consistency within the code, I opted to implement the compressible formulation, as shown in equation 3.8.

3.2 Negative Spalart-Allmaras Model

In July 2012, at the Seventh International Conference on Computational Fluid Dynamics (ICCFD7) held in Big Island, Hawaii, Allmaras et al. [2] presented some valuable

modifications to the original Spalart-Allmaras model. The original model can produce undesirable outcomes in cases involving under-resolved grids and unphysical transients. One such issue is the appearance of undershoots at the edge of boundary layers and in wake regions. Another critical problem occurs when the modified vorticity becomes negative. This situation arises because it is possible for $\tilde{S}$ to reach zero or negative values, as the function f_{v2} itself becomes negative over a certain range of χ. When $\tilde{S}$ turns negative, it disrupts other Spalart-Allmaras correlation functions, leading to non-physical behavior. To address this, the authors revised the definition of the modified vorticity. They proposed that the modified vorticity should always be positive, with a minimum value that does not drop below $0.3S$, where S represents the vorticity magnitude. This ensures that the modified vorticity remains stable and physically realistic, preventing unwanted numerical artifacts. The new modified version is described in the equation below:

$$\tilde{S} = \begin{cases} S + \overline{S} & : \overline{S} \geq -c_{v2}S \\ S + \frac{S(c_{v2}{}^2 S + c_{v3}\overline{S})}{(c_{v3} - 2c_{v2})S - S} & : \overline{S} < -c_{v2}S \end{cases} \tag{3.9}$$

In the Equation 3.9, $\overline{S} = \frac{\tilde{\nu}}{k^2 d^2} f_{v2}$ while the constants c_{v2} and c_{v3} are set, respectively, equal to 0.7 and 0.9. Another important modification involves extending the Spalart-Allmaras model to handle negative values of $\tilde{\nu}$. The original version of the Spalart-Allmaras model is designed to admit only non-negative solutions, provided that non-negative boundary and initial conditions are used. However, the turbulence eddy-viscosity can become negative in certain cases involving coarse grids and transient states. This situation often arises at the edges of boundary layers and wakes, where the turbulence solution involves ramp-like transitions to the constant outer/freestream values over a short region. The rapid transition from relatively large values in the inner region to smaller levels in the outer region can lead to undershoots in the numerical solution. These undershoots may cross zero, necessitating a response to ensure continuity and stability. A common practice to address this has been to clip the updates, effectively eliminating any negative values that appear in the solution. However, the authors modified the original model to maintain both the original behavior for positive $\tilde{\nu}$ and ensure energy stability when negative $\tilde{\nu}$ appears. To accomplish this, a new version of the Spalart-Allmaras equation is used specifically in scenarios where $\tilde{\nu}$ becomes negative. This modified version allows the model to continue providing physically meaningful and stable results even under these conditions. The revised equation is presented as follows:

$$\frac{\partial \rho\tilde{\nu}}{\partial t} + \nabla\cdot(\rho \boldsymbol{u}\tilde{\nu}) = \rho(P_n - D_n) + \frac{1}{\sigma}\nabla\cdot(\rho(\nu + \tilde{\nu}f_n)\nabla\tilde{\nu}) + \frac{cb_2}{\sigma}\rho(\nabla\tilde{\nu})^2 - \frac{1}{\sigma}(\nu + \tilde{\nu}f_n)\nabla\rho\cdot\nabla\tilde{\nu} \tag{3.10}$$

In equation 3.10, the production P_n and destruction D_n term are written as follows:

$$P_n = c_{b1}(1 - c_{t3})S\tilde{\nu} \tag{3.11}$$

$$D_n = -c_{w1}\left(\frac{\tilde{\nu}}{d}\right)^2 \tag{3.12}$$

and f_n is a modification to the diffusion coefficient defined as follows:

$$f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3} \tag{3.13}$$

where the constant c_{n1} is set equals to 16. Differently from the positive SA equation 3.8, the production term depends on the vorticity magnitude S instead of the modified vorticity $\tilde{S}$. The constant c_{t3} is greater than 1, so that the production term is always positive and equals to 1.2. The destruction term, described in the equation 3.12 has a negative sign in comparison with the positive SA equation.

This version of the Spalart-Allmaras model was implemented during the first year of my PhD within the CFD code utilized for my research. Finally, the complete set of mass-averaged RANS equations that have been implemented in the code are presented below. These equations are formulated using Favre averaging; however, the overbar $\bar{\cdot}$ and tilde $\tilde{\cdot}$ notation are omitted, particularly to avoid confusion with the turbulent viscosity variable $\tilde{\nu}$.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{3.14}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} \tag{3.15}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (\boldsymbol{u}(E + p)) = \nabla \cdot (\boldsymbol{\tau} \cdot \boldsymbol{u} - \boldsymbol{q}) \tag{3.16}$$

$$\frac{\partial \rho \tilde{\nu}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \tilde{\nu}) = \rho\,(P - D) + \frac{1}{\sigma}\nabla \cdot (\rho(\nu + \tilde{\nu})\nabla\tilde{\nu}) + \frac{cb_2}{\sigma}\rho(\nabla\tilde{\nu})^2 - \frac{1}{\sigma}(\nu + \tilde{\nu})\nabla\rho \cdot \nabla\tilde{\nu} \tag{3.17}$$

The equation for the energy is given by:

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho|\boldsymbol{u}|^2 \tag{3.18}$$

The viscous stress tensor $\boldsymbol{\tau}$ includes contributions from both the molecular and eddy viscosity. Its components are given by:

$$\tau_{ij} = 2\rho(\nu + \tilde{\nu} f_{v1})\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right) \tag{3.19}$$

Finally, the heat flux $\boldsymbol{q}$ is described by Fourier's law:

$$\boldsymbol{q} = -\left(\frac{c_p \mu}{\mathrm{Pr}} + \frac{c_p \rho \tilde{\nu}}{\mathrm{Pr}_t}\right)\nabla T \tag{3.20}$$

For the test cases considered, corresponding to experiments with air, the following values are assumed: $\gamma = 1.4$, $Pr = 0.72$, and $Pr_t = 0.9$.

3.3 Results of the Negative Spalart-Allmaras Model

The objective of this thesis is to assess the limitations of fully turbulent models in scenarios involving transition-separation phenomena and explore ways to enhance these models using data-driven approaches. Specifically, this study focuses on the Field Inversion and Machine Learning framework, which will be discussed in detail in Chapter 4. One could argue that such problems can be addressed using high-fidelity models like LES or transitional RANS models, as described in Section 2.4. However, high-fidelity methods often come with significant computational costs, sometimes making them impractical, especially in the case of LES simulations.

Beyond addressing these challenges, this work also aims to evaluate the potential of data-driven models in this context. To this end, the Negative Spalart-Allmaras turbulence model (in this thesis, it is often called the original Spalart-Allmaras model) was selected, and four test cases were chosen to apply the proposed methodology: the transitional flat plate, the backward-facing step, the SD7003 airfoil, and the NACA0021 airfoil.

In the following subsection, the test cases are presented in detail, and the results obtained with the original Spalart–Allmaras model are compared against experimental data or high-fidelity simulations, where available. All simulations employ the SLAU (Simple Low-dissipation AUSM) scheme for convective fluxes [82], combined with second-order MUSCL (Monotonic Upwind Scheme for Conservation Laws) reconstruction and the modified Venkatakrishnan slope limiter [93]. Diffusion terms are discretized using a hybrid least-squares gradient reconstruction, while time integration is carried out using a first-order implicit scheme.

3.3.1 Transitional Flat Plate

The T3A test case is a benchmark test case in fluid dynamics, commonly used to investigate the transition from laminar to turbulent flow over a flat plate under zero pressure gradient conditions. It is part of the ERCOFTAC Classic Database, providing experimental data for validating computational fluid dynamics models. To set up the case, a turbulence level is selected, commonly expressed as the turbulence intensity. This is defined as the ratio of the velocity fluctuation to the mean velocity: $Tu = \frac{u'}{\overline{u}}$. Turbulence intensity values between 1% and 5% typically indicate a low Reynolds number flow. In this configuration, the flat plate is exposed to a freestream turbulence intensity of approximately $Tu = 3\%$, enabling researchers to examine the effects of moderate freestream turbulence on boundary layer transition.

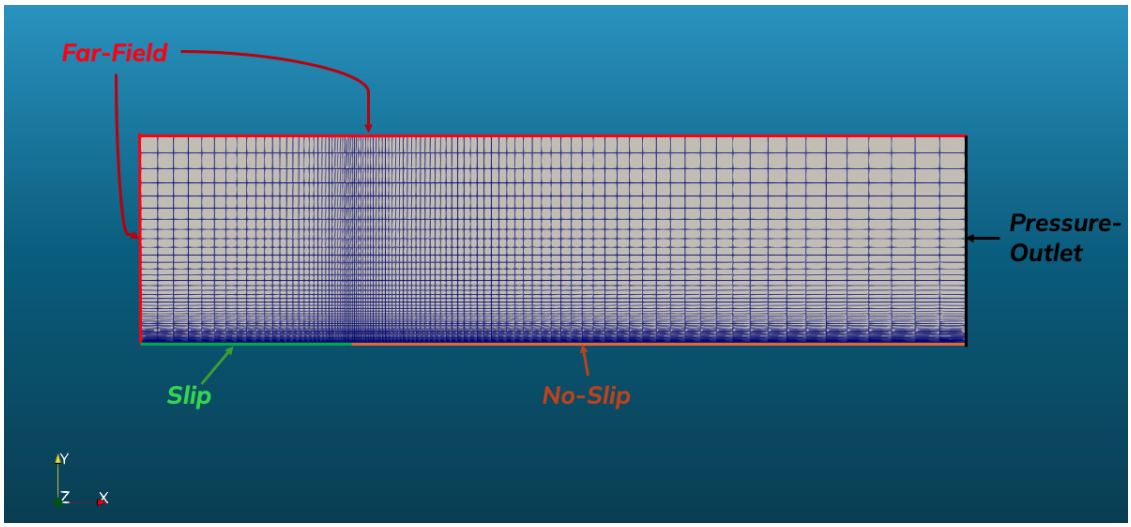


Figure 3.2: T3A Computational mesh with boundary conditions.

Figure 3.2 shows the mesh used for this case, consisting of 8,722 fully structured elements. The mesh is refine near the wall to accurately capture boundary layer characteristics. The figure also illustrates the applied boundary conditions.

This test case serves as a simple yet effective framework to evaluate the field inversion methodology and assess its capability to modify the turbulence equation to align with experimental data. To begin, the results for the skin friction coefficient are presented and compared against experimental measurements. The skin friction coefficient is defined as:

$$C_f = \frac{\tau_w}{0.5\rho_\infty u_\infty^2} \tag{3.21}$$

where τ_w is the local wall shear stress, while ρ_∞ and u_∞ are the density and velocity of the fluid in the freestream, respectively.

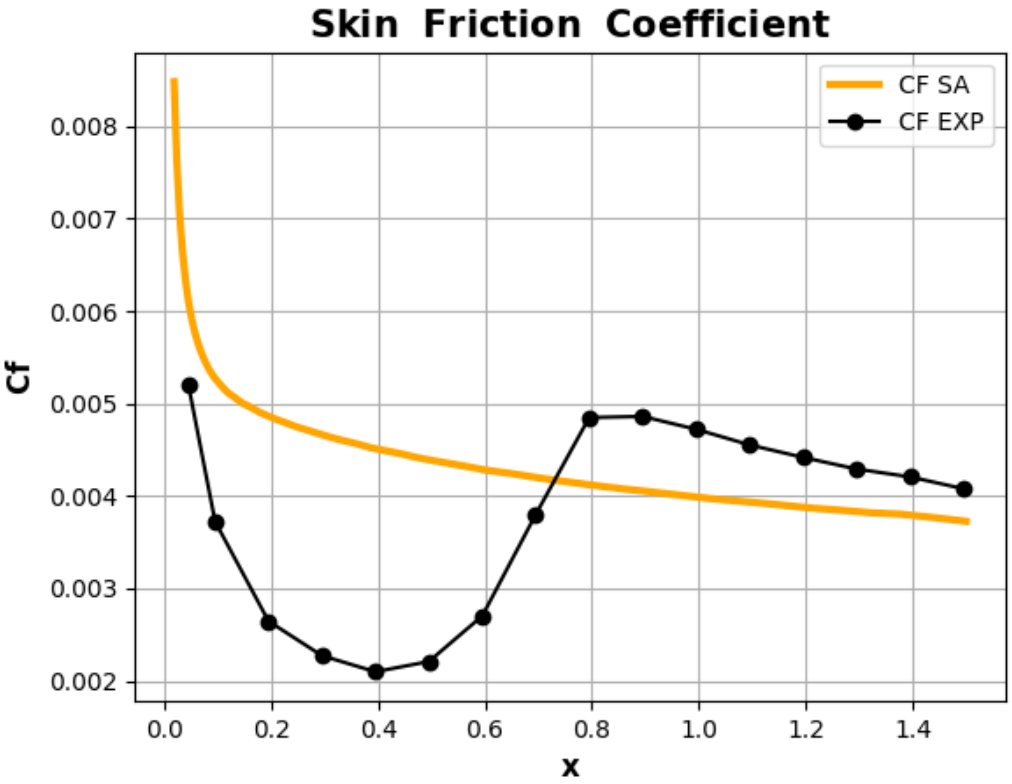


Figure 3.3: T3A skin friction coefficient comparison, Spalart–Allmaras model against experimental results.

As illustrated in Figure 3.3, the experimental results shown in black display the typical transition behavior of the flow. In particular, the skin friction coefficient begins to increase noticeably between $x = 0.4$ and $x = 0.8$, marking the onset of transition. This rise is attributed to the development of a turbulent boundary layer, which is characterized by a steeper velocity gradient near the wall and consequently a higher wall shear stress. However, the figure also shows that the Spalart–Allmaras (SA) model, being a fully turbulent model, is unable to capture this transition phenomenon. In the following section (3.4), an alternative formulation, the Spalart–Allmaras Bas-Cakmakcioglu Mura model, will be introduced and applied to the same test case for comparison. Additionally, subsequent analysis will present the results obtained using the augmented SA model with the field inversion technique.

3.3.2 Backward-Facing step

The backward-facing step (BFS) turbulent test case is a fundamental benchmark in fluid dynamics, widely used to study flow separation, reattachment, and recirculation

phenomena. In this setup, a fluid flows through a channel that experiences a sudden expansion due to a step on one of its walls. This abrupt change causes the boundary layer to separate at the step edge, forming a shear layer that eventually reattaches downstream. The region between the step and the reattachment point is characterized by a recirculation zone, where the flow reverses direction.

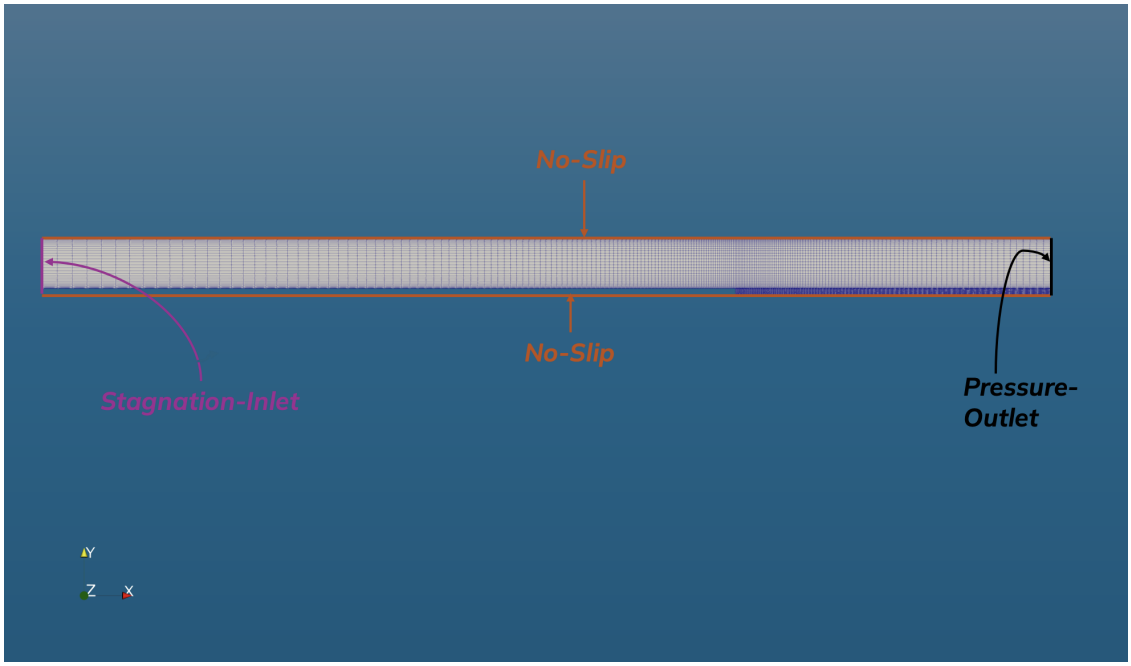


Figure 3.4: BFS Computational mesh with boundary conditions.

Figure 3.4 shows the computational mesh and boundary conditions used for the BFS test case. Refinement near the wall ensures accurate resolution of the boundary layer, especially in the region following the step. The mesh contains a total of 45,941 elements. The BFS flow is particularly challenging due to the complex interactions between the separated shear layer and the recirculating flow. Accurately predicting the length of the recirculation zone and the reattachment point is a stringent test for turbulence models. Understanding BFS flows has practical implications in various engineering applications where flow separation and reattachment occur, such as in diffusers, combustors, and aerodynamic surfaces. On NASA's Turbulence Modeling Resource website, several validation and verification results for CFD codes are published, along with the boundary conditions used to run the tests. In these studies, simulations were performed at a Reynolds number of $Re = 36{,}000H$, where H represents the step height and for this test case it is equal to 1, and a Mach number of $M = 0.128$. This test case is particularly suitable for applying the field inversion technique to the SA model. The BFS scenario involves flow separation and reattachment, which are challenging for fully turbulent models. Additionally, it allows for testing the field inversion framework's ability

to work with different objective functions, specifically skin friction and pressure coefficient. The pressure coefficient is defined as follows:

$$Cp = \frac{p_w - p_\infty}{0.5\rho_\infty u_\infty} \tag{3.22}$$

where p_w is the wall pressure and p_∞ is the pressure in the free-stream. Below, Figure 3.5 presents the results obtained using the Negative Spalart–Allmaras model, alongside experimental data retrieved from the NASA website for comparison. The figure focuses on the region of interest, extending from $x = 0$ downstream, where $x = 0$ denotes the location of the step.

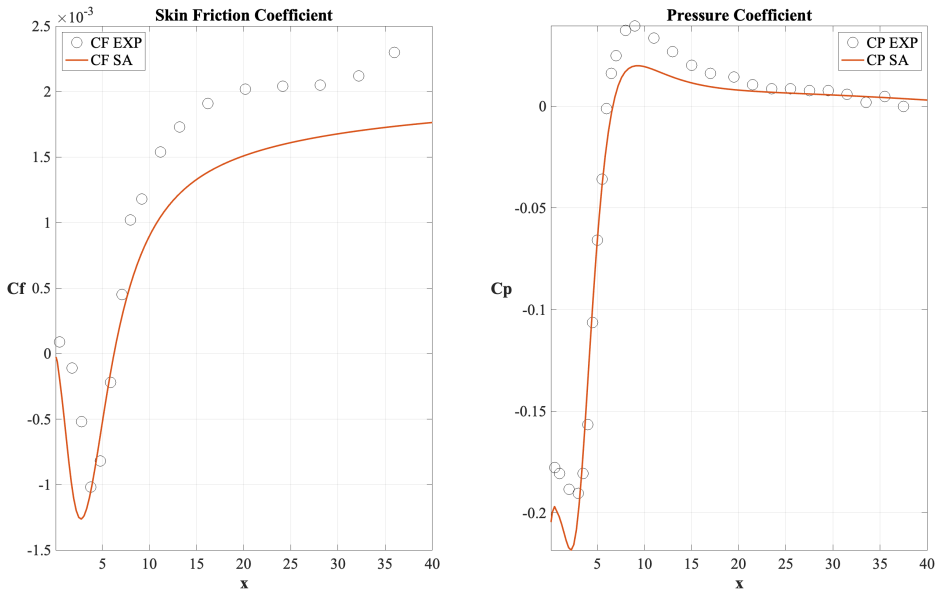


Figure 3.5: Skin friction and pressure coefficients predicted for the Backward-facing step test case by the Negative Spalart–Allmaras model.

Figure 3.5 shows the results in the step downstream region, where the flow is expected to separate and a recirculation region is presented. A discrepancy can be seen between the SA results and the experimental results, particularly for the skin friction coefficient.

3.3.3 SD7003 Airfoil

The SD7003 airfoil, designed by Selig and Donovan, is a well-known airfoil optimized for low Reynolds number applications, typically ranging from 10^4 to 10^5. It is widely employed in small-scale aircraft, unmanned aerial vehicles (UAVs), and model aircraft due to its favorable aerodynamic performance at low speeds. Given its suitability

for low Reynolds number regimes, the SD7003 airfoil is often selected for applications where low-speed aerodynamics play a critical role.

A distinctive feature of the SD7003 airfoil is its tendency to form laminar separation bubbles. These bubbles arise when laminar flow detaches from the suction side of the airfoil, undergoes transition to turbulence, and subsequently reattaches. The presence of laminar separation bubbles has a significant influence on the airfoil's performance, particularly affecting lift and drag characteristics.

In this thesis, the SD7003 airfoil is used to analyze separation behavior through simulations conducted with the SA turbulence model.

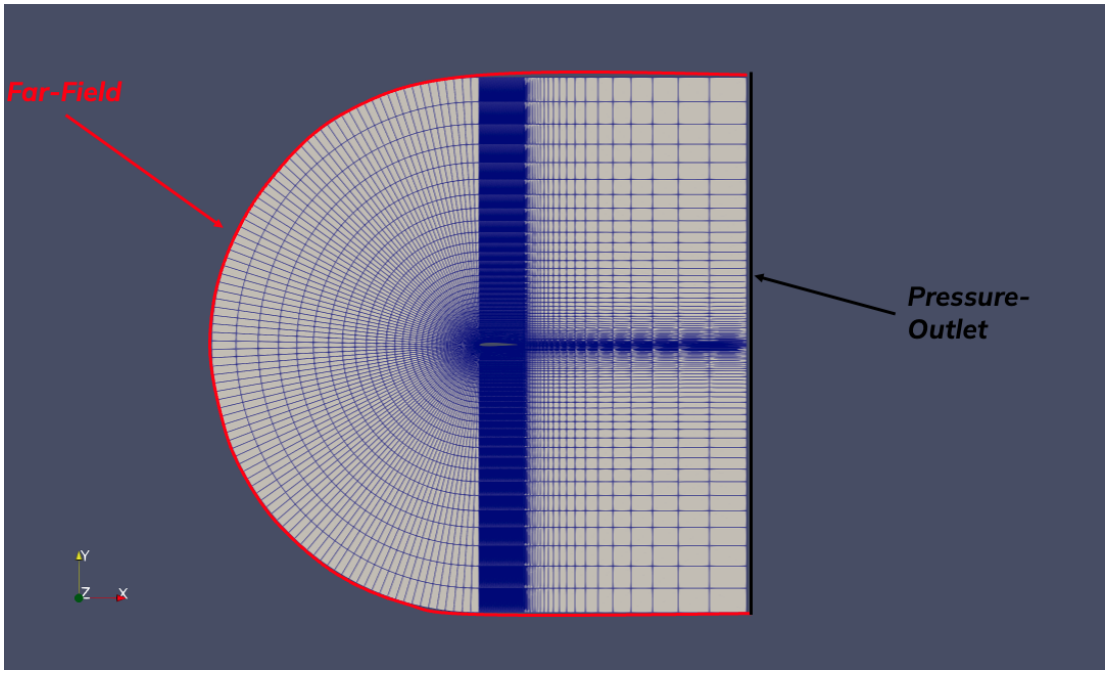


Figure 3.6: SD7003 computational mesh.

The mesh used for the SD7003 airfoil is a C-type configuration, as shown in 3.6, and consists of 149,120 elements. It is fully structured, with increased refinement near the wall to accurately resolve the boundary layer features.

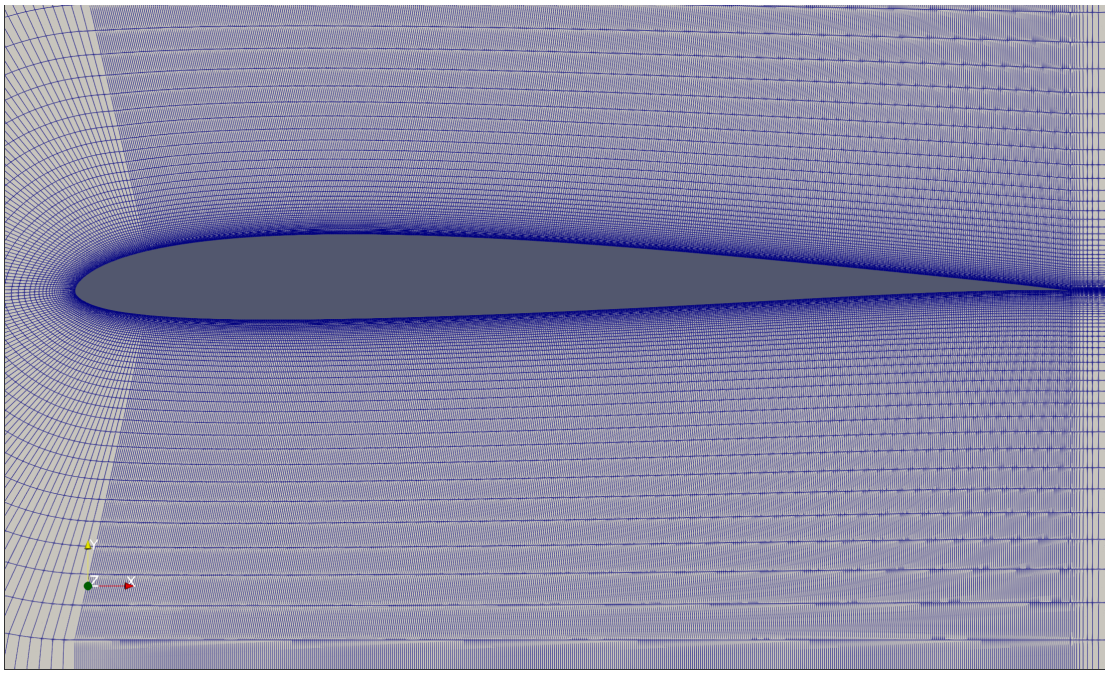


Figure 3.7: Detailed view of the boundary layer mesh around the SD7003 airfoil.

The size of the first cells adjacent to the wall was determined based on the target y^+ value, calculated using the Reynolds number specific to this simulation. A close-up view of the mesh near the airfoil surface is provided in Figure 3.7. Regarding the boundary conditions, a far-field condition was applied at the inlet, a pressure outlet was specified at the downstream boundary, and a no-slip condition was imposed on the airfoil surface.

The computational analysis is performed at a Reynolds number of 60,000 and angles of attack of $\alpha = 4°$ and $\alpha = 8°$. At an angle of attack of $\alpha = 4°$, moderate flow separation is expected, primarily near the trailing edge. At $\alpha = 8°$, a more pronounced separation is anticipated, along with the formation of laminar separation bubbles.

It is well established that the SA model has limitations in accurately predicting separation, particularly at low Reynolds numbers where complex laminar-to-turbulent transition processes occur. This makes the SD7003 airfoil a suitable test case for applying field inversion and machine learning techniques to assess their potential in improving the baseline turbulence model. For this test case, the comparison is performed using the lift and drag coefficients (C_L and C_D) obtained from high-fidelity LES simulations. In the field inversion results (Section 4.5.3), the skin friction and pressure coefficients (C_f and C_p) will also be considered. Further details about the LES simulations can be found in [29].

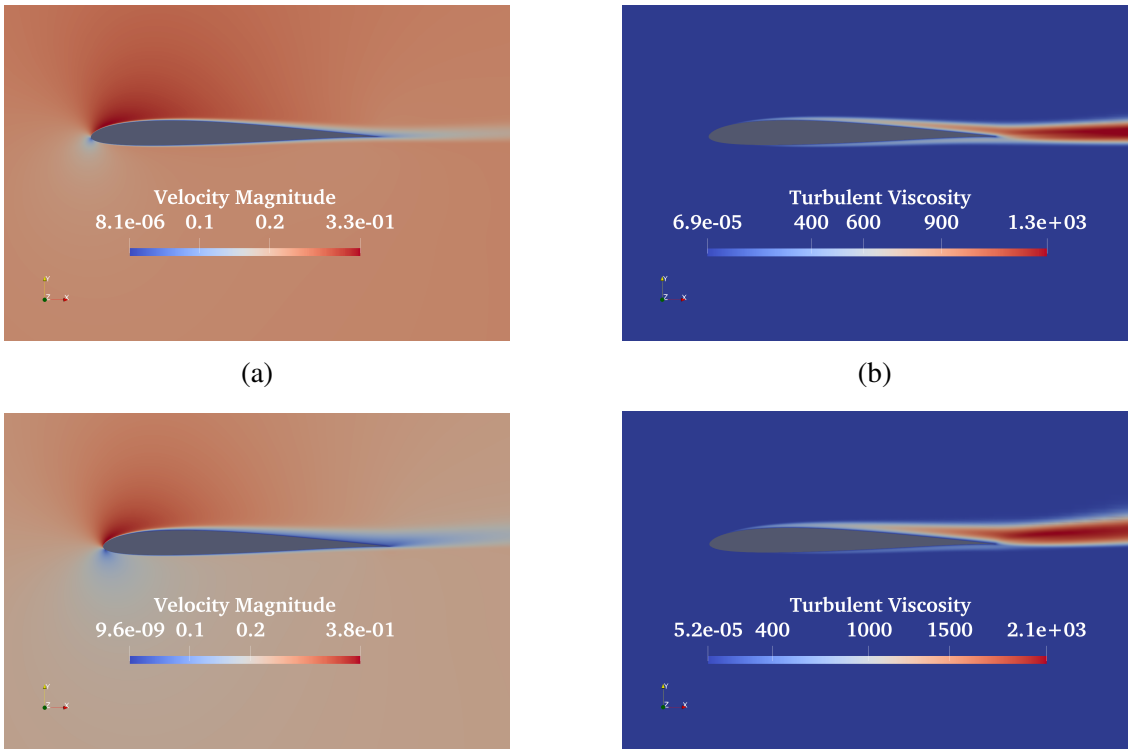


Figure 3.8: Results of the Spalart–Allmaras model for the SD7003 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 4°$. (**c**) Velocity magnitude and (**d**) turbulent viscosity at $\alpha = 8°$.

The results obtained from the original SA model are presented in Figure 3.8. The turbulent viscosity and velocity fields exhibit similar behavior at both angles of attack. However, a distinct difference is expected, as separation is anticipated to initiate near the trailing edge at $\alpha = 4°$ and extend toward the leading edge at $\alpha = 8°$.

		CL	**CD**
$\alpha = 4°$	LES	0.59	0.021
	SA	0.551	0.0099
$\alpha = 8°$	LES	0.92	0.043
	SA	0.895	0.0435

Table 3.1: Comparison of lift and drag coefficients for SD7003 airfoil.

More detailed results are presented in Table 3.1, where C_L and C_D are compared to the high-fidelity results from a LES simulation [29]. There is no significant difference in

C_L for both angles of attack. For C_D at $\alpha = 8°$, good agreement with the experimental results is observed, whereas a relative error of 52% is found for $\alpha = 4°$.

4

3.3.4 NACA0021 Airfoil

The NACA0021 airfoil test case is typically used to evaluate the performance of transitional RANS models. Choudhry et al. [15] examined the flow around a two-dimensional NACA0021 airfoil at low Reynolds numbers and turbulence intensities, focusing on the effects of long separation bubbles on aerodynamic performance.

The study evaluated two transition models: the correlation-based $\gamma - Re_\theta$ intermittency model and the laminar–kinetic-energy-based $k - k_L - \omega$ model. Experimental and computational lift and drag coefficients were compared at a Reynolds number of 120,000. The $\gamma - Re_\theta$ model under-predicts both coefficients, with increasing discrepancies at higher angles of attack. In contrast, the $k - k_L - \omega$ model demonstrates better agreement with experimental lift coefficients in the attached-flow regime and accurately captures drag coefficients and stall behavior. This model also successfully predicted the abrupt stall caused by the bursting of the separation bubble at low Reynolds numbers, although deviations from the experimental lift coefficient remained at higher angles of attack.

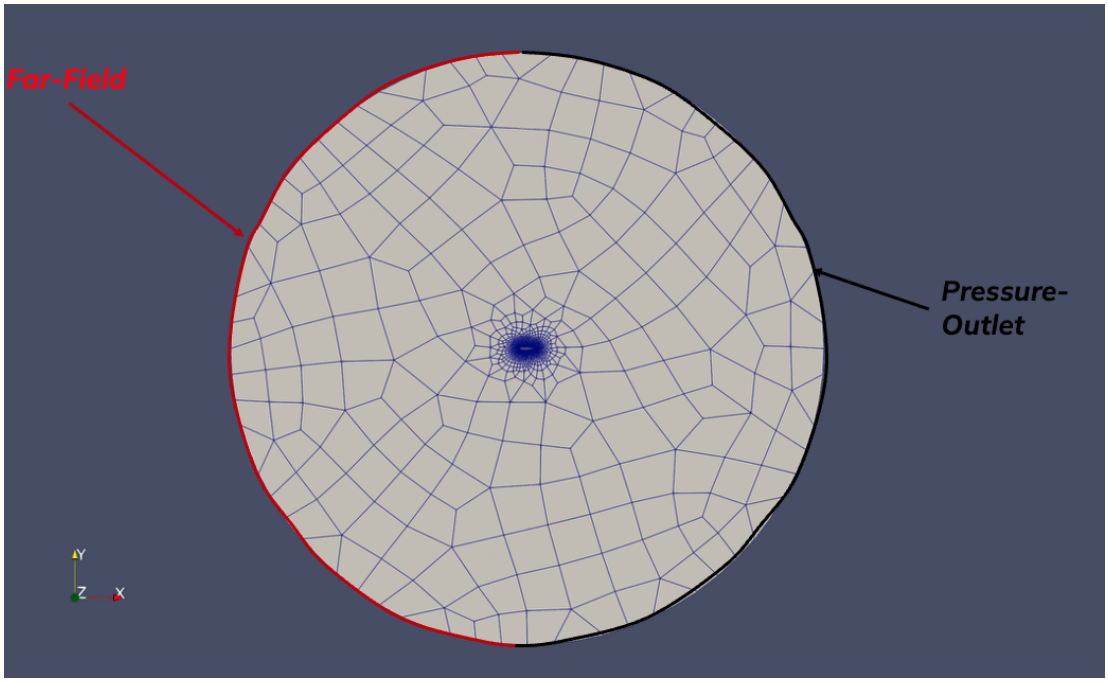


Figure 3.9: NACA0021 computational mesh.

For this test case, an O-type mesh was employed, as illustrated in Figure 3.9. The

mesh around the NACA0021 airfoil was generated using a hybrid approach that combines structured and unstructured elements, ensuring an accurate representation of the flow features. The final mesh consists of 14,824 elements. Following the approach of [15], the computational domain boundaries were placed at a distance of 20 chord lengths from the airfoil to minimize their influence on the flow field.

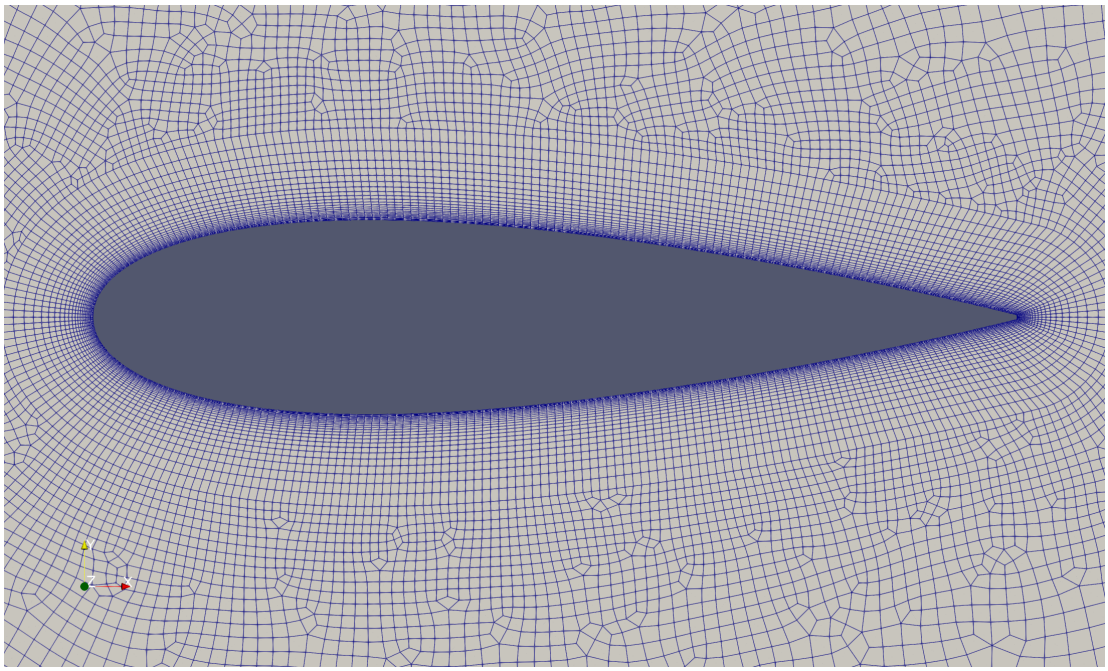
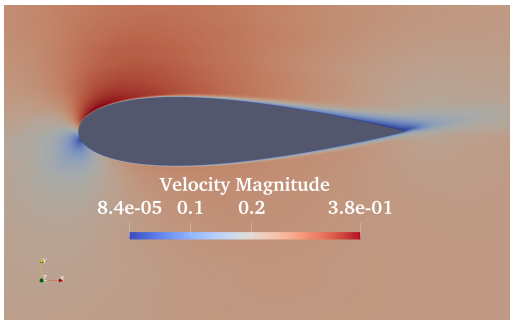
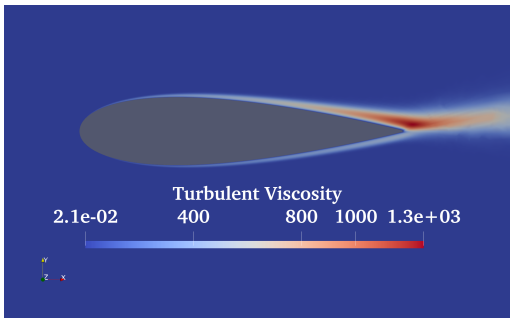


Figure 3.10: Detailed view of the boundary layer mesh around the NACA0021 airfoil.

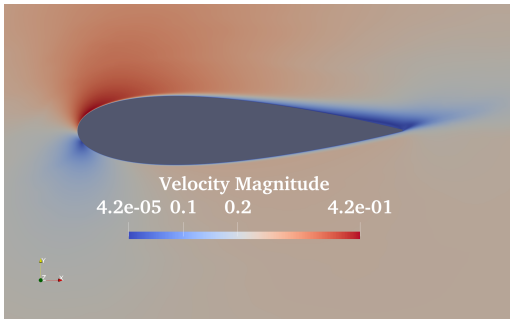
The size of the first cell was selected to satisfy the minimum y^+ requirements of the Reynolds number adopted in this study. To accurately resolve the boundary layer, 27 structured layers with linear growth were employed, as illustrated in Figure 3.10. Furthermore, transfinite interpolation was used to control the point distribution in critical regions, such as the leading and trailing edges, allowing for refined mesh resolution where it is most needed. As for the boundary conditions, a far-field condition was applied at the inlet, a pressure-outlet condition was set at the downstream boundary, and a no-slip condition was enforced on the airfoil surface. For this test case, experimental C_L and C_D data were available for $\alpha \in [0°, 20°]$. In this work six angles of attack were considered: $\alpha = 8°, 10°, 12°, 14°, 16°$, and $20°$.



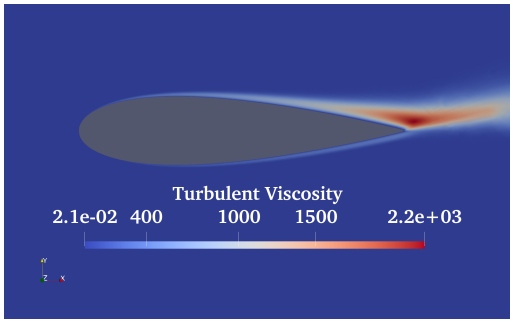
(a)



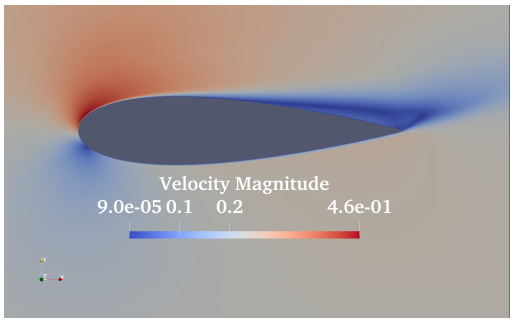
(b)



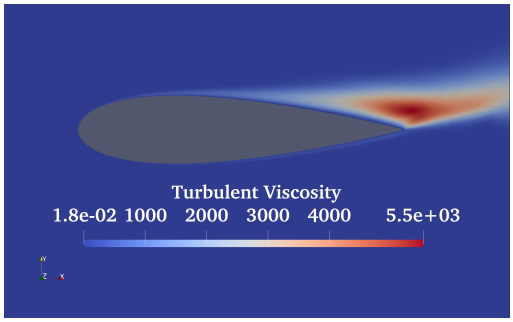
(c)



(d)



(e)



(f)

Figure 3.11: Results of the Spalart–Allmaras model for the NACA0021 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 8°$. (**c**) Velocity magnitude and (**d**) turbulent viscosity at $\alpha = 12°$. (**e**) Velocity magnitude and (**f**) turbulent viscosity at $\alpha = 16°$.

Figure 3.11 presents the velocity and turbulent viscosity fields predicted by the original SA model for $\alpha = 8°, 12°$, and $16°$. The velocity field reveals an expanding separation region originating from the trailing edge of the airfoil, resulting in increased turbulent viscosity in this area. However, this effect remains insufficient for accurately predicting C_L and C_D values.

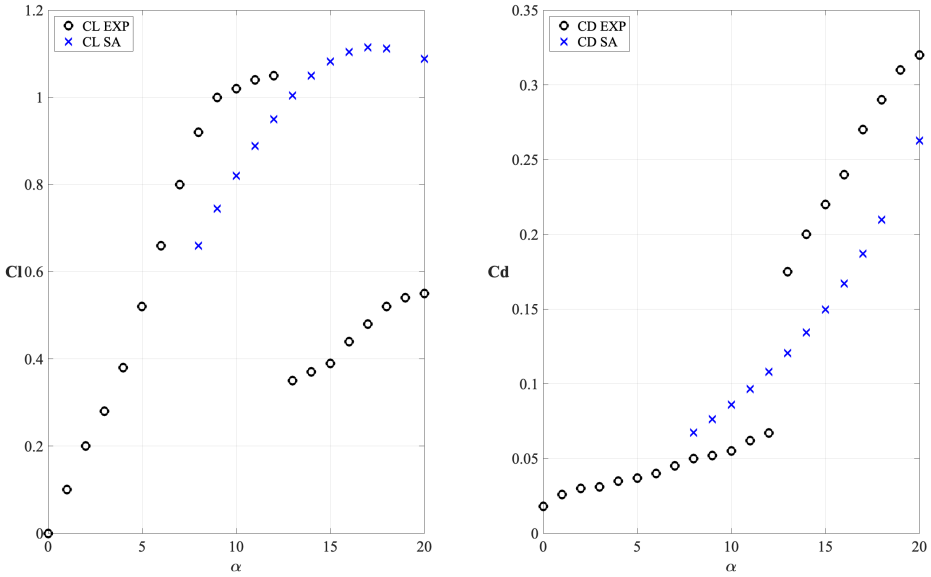


Figure 3.12: Lift and drag coefficients predicted for the NACA0021 test case by the Spalart–Allmaras model at various angles of attack.

Figure 3.12 presents the results obtained using the original SA model across various angles of attack, encompassing both pre-stall and post-stall conditions. The C_L trends indicate that as the angle of attack increases, the SA model fails to accurately capture the stall and subsequent flow separation.

3.4 Spalart-Allmaras Bas-Cakmakcioglu Mura Model

The Spalart-Allmaras model was originally developed for fully turbulent flow; however, in their initial paper, Spalart et al. introduced an additional term to account for separation phenomena. Their analysis begins with the stability behavior of the code under laminar flow conditions. They observed that the solution for $\tilde{\nu} = 0$ was not preserved in the Navier-Stokes code. To address this issue, they added a new term by multiplying the production term by $(1 - f_{t2})$. The equation for the function f_{t2} is presented as follows:

$$f_{t2} = c_{t3} \exp\left(-c_{t4}\chi^2\right) \tag{3.23}$$

The function in equation 3.23 serves to bring $\tilde{\nu}$ down from values around $\nu/2$. Based on this reasoning, the authors selected the constants $c_{t3} = 1.2$ and $c_{t4} = 0.5$. This new function has no impact for high Reynolds numbers preserving the original property of the model. Since this new term was added to the production term, a similar adjustment was made to the destruction term to maintain the balance near the wall.

The updated forms of the production and destruction terms in this version are given as follows:

$$P = c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu}, \quad D = \left(c_{w1}f_w - \frac{c_{b1}}{k^2 f_{t2}}\right)\left[\frac{\tilde{\nu}}{d}\right]^2 \tag{3.24}$$

To initiate the transition process, the authors introduced an additional source term, referred to as the trip term. This term is designed to be nonzero only within a localized domain of influence, ensuring that its effect is limited. Importantly, this domain must remain entirely within the boundary layer without extending beyond it.

$$T = f_{t1}\Delta U^2 \tag{3.25}$$

In equation 3.25 ΔU represents the magnitude of the difference between the velocity at the trip and the velocity at the field point under consideration. The function f_{t1} is expressed as follows:

$$f_{t1} = c_{t1}g_t \exp\left(-c_{t2}\frac{\omega^2}{\Delta U^2}[d^2 + g_t^2 d_t^2]\right) \tag{3.26}$$

In equation 3.26, g_t represents a grid-dependent parameter that ensures the trip term remains nonzero over a few streamwise stations. The corresponding expression is given as follows:

$$g_t \equiv min(0.1, \Delta U/\omega_t \Delta x_t) \tag{3.27}$$

This parameter depends on Δx_t, which is the grid spacing along the wall at the trip point. Additionally, in equation 3.26, d_t denotes the distance from the field point to the nearest trip point or line, while ω_t represents the wall vorticity at the trip point. The constants c_{t1} and c_{t2} are set to 1 and 2, respectively. It is evident that the transitional version of the Spalart-Allmaras model, as proposed in the original paper, is highly dependent on the location of transition, which must be known a priori. This dependence on prior knowledge is a significant limitation, as the exact transition point is often unknown. Ideally, a model should be capable of autonomously determining the correct transition point without requiring prior specification. However, the authors acknowledge that the responsibility for selecting transition points lies with the user, whether through an educated guess or by using a separate prediction method.

In 2020, Mura et al. [61] introduced a new transitional version of the Spalart-Allmaras model called SA BCM, which stands for Spalart-Allmaras Bas-Cakmakcioglu-Mura. The key enhancement in this model is the incorporation of a function, γ_{BCM}, which

acts as an intermittency factor, similar to the approach used in the $\gamma - Re_{\theta_t}$ model. Unlike the previous model, however, this intermittency function is not governed by a transport equation; instead, the authors developed an algebraic formulation for it.

$$\gamma_{BCM} = 1 - e^{-\sqrt{Term_1} - \sqrt{Term_2}} \tag{3.28}$$

In equation 3.28, $Term_1$ primarily estimates the intermittency and is expressed as follows:

$$Term_1 = \frac{\max(Re_\theta - Re_{\theta_c}, 0)}{\chi_1 Re_{\theta_c}} \tag{3.29}$$

The constant χ_1 is equal to 0.002. The Re_{θ_c} represents the critical value of the momentum thickness Reynolds number, which is a function of the freestream turbulence intensity, Tu_∞. An empirical relationship for this dependency is given by the following formulation:

$$Re_{\theta_c} = 803.73(Tu_\infty + 0.6067)^{-1.027} \tag{3.30}$$

The momentum thickness Reynolds number depends on the vorticity Reynolds number, and its formulation was previously presented during the introduction of the $\gamma - Re_{\theta_t}$ model in equation 2.19. The physical interpretation of $Term_1$ is that it determines the onset location of transition by comparing the locally Re_θ to the value of Re_{θ_c}. Once the vorticity Reynolds number, Re_v, exceeds a critical threshold, $Term_1$ becomes greater than zero, thereby activating the intermittency function γ_{BCM}. However, Re_v depends on the distance to the nearest wall, which means that Re_v takes very low values within the boundary layer close to the wall. As a result, $Term_1$ alone cannot generate intermittency within the boundary layer. $Term_2$ is introduced to address this issue. $Term_2$ is defined as follows:

$$Term_2 = \max\left(\frac{\mu_t}{\chi_2 \mu}, 0\right) \tag{3.31}$$

χ_2 is a calibration parameter set to 0.02. The bounding of this term to positive values is unnecessary because eddy viscosity cannot be negative. Nevertheless, the authors retained this bound for the sake of generality. For this model, the freestream value of $\tilde{\nu}$ is set to $\tilde{\nu}_{farfield} = 0.015\nu_\infty : to : 0.025\nu_\infty$. Additionally, the f_{t2} term must be set to zero; otherwise, the production term would consistently yield a nearly zero value under the specified freestream $\tilde{\nu}$.

This version of the SA model was incorporated into the CFD code to compare its performance against the original SA formulation, particularly in addressing transition phenomena. As previously mentioned, this model aims to replicate the intermittent behavior of flow during the transition from laminar to turbulent states. I selected the flat plate test for three different scenarios, which exemplify natural transition phenomena, distinguished by varying inlet velocities and turbulence intensities: 1) $V_\infty = 5.4; Tu = 3\%$

(T3A case shown in Figure 3.13), 2) $V_\infty = 9.4; Tu = 6\%$, (T3B case shown in Figure 3.14), 3) $V_\infty = 19.8; Tu = 0.9\%$ (T3A- case shown in Figure 3.15). The mesh used from all these test cases is the same as the one shown in 3.2.

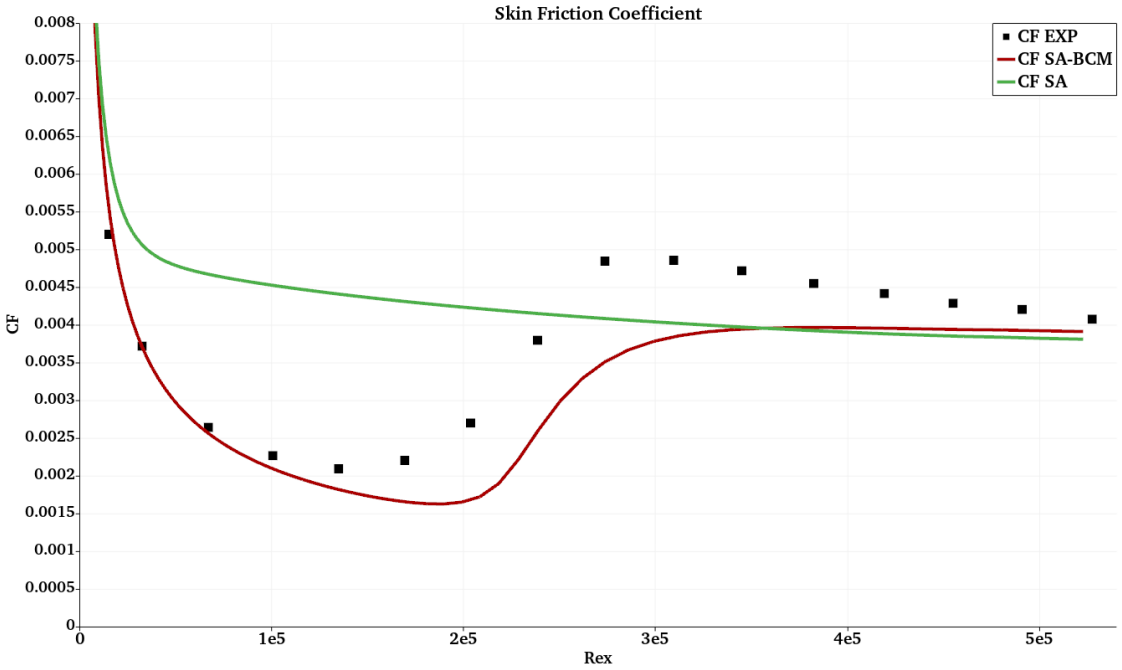


Figure 3.13: Comparison between experimental, original SA and SA-BCM results for the skin friction coefficient for the T3A test case.

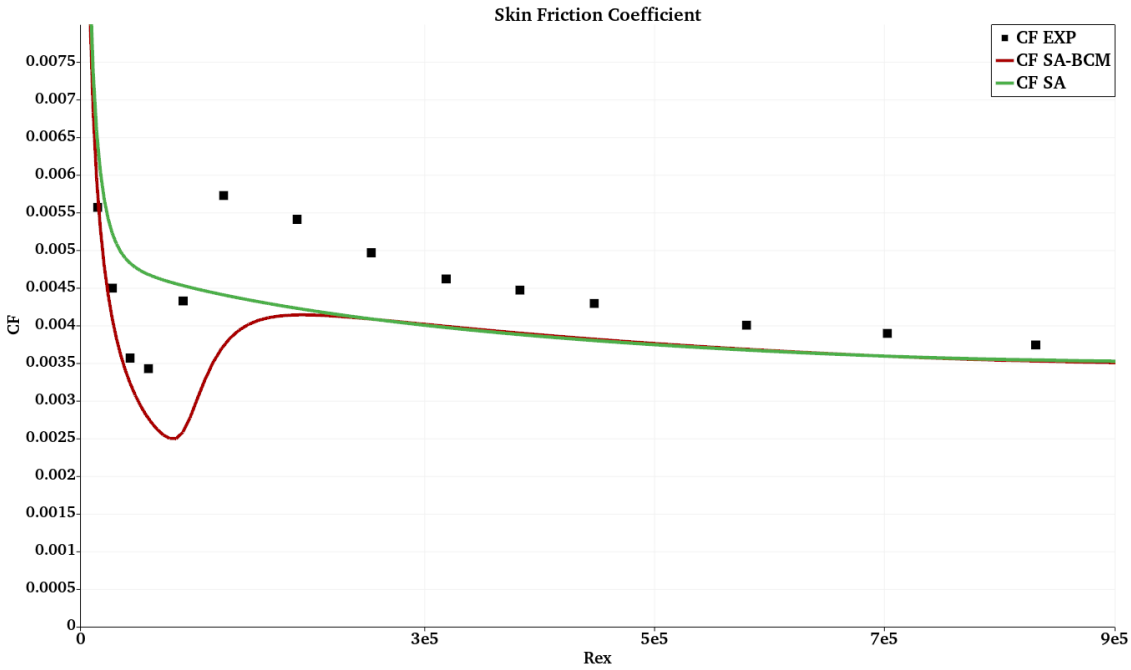


Figure 3.14: Comparison between experimental, original SA and SA-BCM results for the skin friction coefficient for the T3B test case.

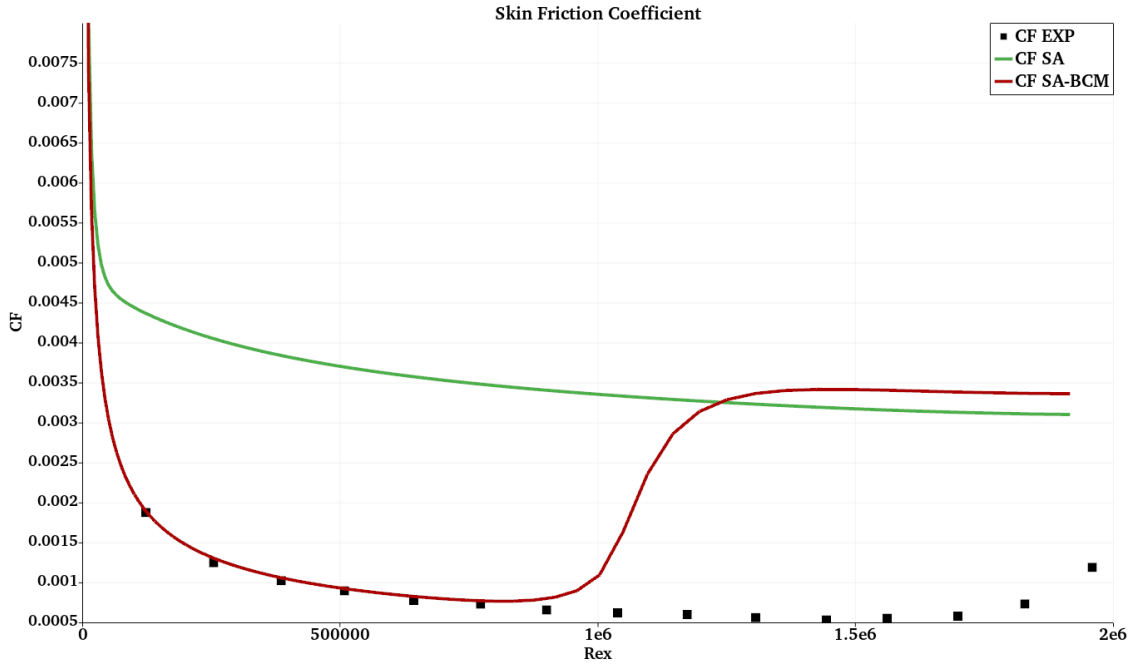


Figure 3.15: Comparison between experimental, original SA and SA-BCM results for the skin friction coefficient for the T3A- test case.

The results highlight the differences between the experimental data, the original SA model, and the SA-BCM model. The SA model is implemented with the $ft2$ term set to zero; since this simulation begins with a low value of turbulent viscosity, disabling the $ft2$ term is crucial to prevent the flow from remaining laminar, as the production term would be near zero. The figures clearly show the differences between the two models. However, the SA-BCM model does not exactly replicate the experimental results. Figures 3.13 and 3.14 indicate a transition behavior, but the transition point is incorrect. In Figure 3.15, the model shows a significant error by predicting the transition too early.

Chapter 4

Field Inversion

As detailed in Section 2.5, the primary objective of this thesis is the implementation and investigation of the Field Inversion and Machine Learning approach. This method, developed by Duraisamy et al. in [39], aims to enhance turbulence RANS models. The strength of this approach lies in its flexibility; it can utilize any type of data to augment the model, whether high-fidelity simulations, experimental data, or even global coefficients such as lift or drag. The method operates directly on the selected RANS model for the simulation, focusing on the RANS results to identify the missing components necessary for the model to accurately represent experimental or high-fidelity outcomes.

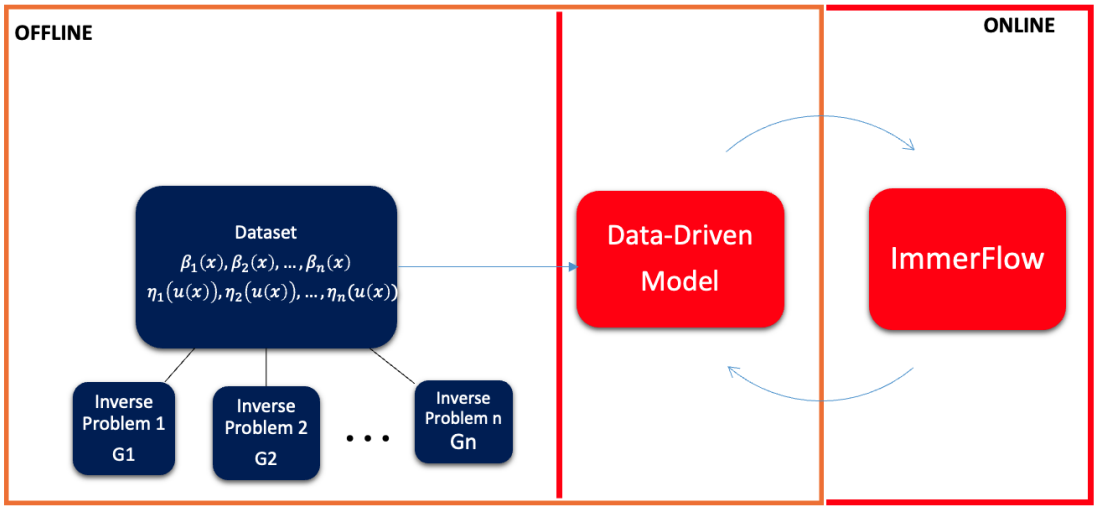


Figure 4.1: Schematic of Field Inversion and Machine Learning framework.

This chapter provides a comprehensive description of the model and presents the results obtained during the doctoral research. It is important to note that this method consists of two main phases. Figure 4.1 highlights these two phases, the offline and the

online phase. The offline phase consists of an optimization process called field inversion, which is designed to improve the original RANS model and generate the appropriate dataset for training the data-driven model. The field inversion framework is described in this chapter. From figure 4.1 it can be seen that the data-driven model belongs to both the offline and online phases. In fact, while the model is trained in the offline phase, it is used directly in the online phase to infer the behavior of the augmented model that interacts directly with the CFD solver.

4.1 Augmented Spalart-Allmaras model

This work aims to enhance the SA model, specifically the negative SA model introduced in 3.2. The goal is to equip this model with the capability to produce accurate results in the presence of transitional flow behavior. As described in 2.4, various phenomena can trigger flow transition, and the overarching objective is to predict the transition point. This prediction is critical for achieving reliable results in industrial design, as classical RANS models often struggle to capture these phenomena. Evidence presented in 3.4 demonstrates that the SA model fails to provide accurate predictions for the skin friction coefficient in transitional flat plate scenarios. Moreover, even the variant of the SA model, the SA-BCM, does not achieve a perfect match with experimental results. The aim here is to develop a better version of the SA model using the Field Inversion technique. By focusing on transitional transport equations like the $\gamma - Re_\theta$ model explained in 2.4.3, the idea is to modify the production term of the RANS equations to mimic the intermittent behavior of the flow during transition. The intermittency variable γ is determined via a transport equation in the case of the $\gamma - Re_\theta$ model or through an algebraic function in the case of the SA-BCM model, with both formulations grounded in physical and numerical reasoning. However, even the widely used $\gamma - Re_\theta$ model does not yield optimal results. This is evident in the work of Choudhry et al. [15], where the authors employed this model to predict the lift and drag coefficients of the NACA0021 at various angles of attack, ranging from pre- to post-stall flow. Accurately determining the global coefficients is crucial in this context, as it is essential to identify the angle of attack at which stall occurs, given its significant implications for performance and safety.

Ideally, the Field Inversion approach allows for the modification of any terms in the equations. For example, Brenner et al. in [9] utilized Field Inversion to directly alter the turbulent viscosity field by correcting the $\tilde{\nu}$ variable in the SA equation. In this case, the model equations remained unchanged; instead, the authors focused on correcting the f_{v1} equation in 3.6. However, from a numerical implementation perspective within a Finite Volume scheme, this approach presents challenges, as it requires retrieving variables from the grid cell interface to its center. Alternatively, similar results can be achieved by modifying the source terms of the SA equation, which involves a less complex implementation. In a Finite Volume scheme, source terms are treated as volume integrals, meaning that the variables reside at the grid cell centers. After implementing the Field

Inversion algorithm, I attempted to modify each source term individually; however, only the production term proved effective. I also explored the possibility of removing the source terms from the SA equation to create a global term or adding a new term, but both approaches failed without a foundational physical or mathematical term. The Field Inversion algorithm operates during the flow simulation, and without a solid physical basis, any corrections made can lead to significant numerical instability, hindering the algorithm's convergence. Consequently, after these trials, I opted to retain the original approach and to modify only the production term of the equation. Specifically, the production term P is adjusted using a correction field, referred to as β, which is inferred through the optimization process. This process involves solving an inverse problem to minimize the discrepancy between model predictions and experimental data. The augmented SA model is expressed as:

$$\frac{\partial \rho\tilde{\nu}}{\partial t} + \nabla\cdot(\rho\boldsymbol{u}\tilde{\nu}) = \rho\left[h(\beta)P - D\right] + \frac{1}{\sigma}\nabla\cdot(\rho(\nu+\tilde{\nu})\nabla\tilde{\nu}) + \frac{cb_2}{\sigma}\rho(\nabla\tilde{\nu})^2 - \frac{1}{\sigma}(\nu+\tilde{\nu})\nabla\rho\cdot\nabla\tilde{\nu} \quad (4.1)$$

Since I am using the negative SA model, the correction is also applied to the negative production term when $\tilde{\nu}$ becomes negative. In such cases, the equation becomes:

$$\frac{\partial \rho\tilde{\nu}}{\partial t} + \nabla\cdot(\rho\boldsymbol{u}\tilde{\nu}) = \rho(h(\beta)P_n - D_n) + \frac{1}{\sigma}\nabla\cdot(\rho(\nu+\tilde{\nu}f_n)\nabla\tilde{\nu}) + \frac{cb_2}{\sigma}\rho(\nabla\tilde{\nu})^2 - \frac{1}{\sigma}(\nu+\tilde{\nu}f_n)\nabla\rho\cdot\nabla\tilde{\nu} \quad (4.2)$$

In both equations 4.1 and 4.2, the production term is modified by $h(\beta)$, a function of β, often referred to as the mapping function (detailed definition of the mapping function can be found in section 4.3). The outcome of the optimization process is the correction field β, which is determined by defining the optimization problem as follows:

$$\beta^* = \arg\min_{\beta} G(u_{num}, u_{exp}) \text{ s.t.} \quad (4.3)$$

$$R(u_{num}(x,\beta(x)),\beta(x)) = 0 \quad (4.4)$$

where β^* is the optimal correction field, while u_{num} and u_{exp} are the numerical and the experimental values, respectively. Equation 4.3 defines the problem statement, which essentially seeks to find the value of the correction field that minimizes the goal function G. The goal function G measures the error between the numerical solution and the experimental or high-fidelity data. The problem statement is accompanied by a condition expressed in 4.4. This condition is fundamental, as it states that the residuals R of the equations must equal zero, indicating that the simulation must achieve a steady-state regime.

4.2 Adjoint Method

The correction field is obtained after solving the problem described in 4.3 and 4.4. Since this is an optimization problem, knowledge of the gradient is essential. Optimization algorithms can generally be divided into two categories: gradient-based and gradient-free methods. In the context of CFD, gradient-free methods tend to be prohibitively expensive because they require repeated evaluations of the objective function, each involving a full CFD simulation. This leads to significant computational cost. Gradient-based methods, while more computationally efficient, present their own challenges. Specifically, they require the gradient of the objective function G, which is dependent on the conservative flow variables. This makes the gradient a high-dimensional entity, with its size proportional to the number of cells in the computational mesh. Therefore, calculating the gradient numerically can be computationally prohibitive for large-scale CFD problems.

One of the most effective methods for calculating gradients in CFD is the adjoint method. This approach, originally developed within control theory, was first introduced by Lions in 1971 [48] and later applied to fluid dynamics by Pironneau in 1984 [67]. The adjoint method has gained widespread popularity in CFD due to its remarkable efficiency in high-dimensional optimization problems. By solving the adjoint equations, one can compute the gradient of the objective function with respect to all design variables at a cost that is nearly independent of the number of variables, making it especially suitable for large-scale CFD applications. The adjoint method has seen successful applications in a wide variety of fields. For example, in aerodynamics, it is commonly used to optimize aircraft wing shapes to reduce drag while maintaining sufficient lift [51, 42]. Another intriguing application is in climate modeling, where the adjoint method is employed to optimize model parameters and improve the accuracy of long-term forecasts [50]. A particularly interesting application of the adjoint method was demonstrated by Kostantinos Samouchos in his Ph.D. thesis [78]. He integrated the adjoint method into an immersed boundary CFD solver using the cut-cell method, which eliminates the complications associated with body-fitted meshes that arise from geometry changes during optimization. This method avoids issues of mesh perturbation by differentiating the corresponding mesh-handling tools. Traditional mesh deformation techniques can struggle to handle significant geometric changes, potentially causing the optimization to terminate prematurely. In such cases, manual intervention is often required, which compromises the autonomy of the optimization process. By contrast, Samouchos' use of Immersed Boundary Methods (IBMs) provides a more robust and reliable framework, allowing for seamless optimization without the limitations imposed by mesh deformation tools.

The adjoint method exists in two main forms: the continuous adjoint and the discrete adjoint. Both approaches are designed to efficiently compute the gradient of an objective function, but they differ significantly in how they are formulated and implemented. In the

continuous adjoint method, the adjoint equations are derived directly from the continuous governing equations of the physical system, such as the Navier-Stokes equations in fluid dynamics. This approach formulates the adjoint equations based on the partial differential equations (PDEs)that describe the system's physics. The key advantage of this method is its theoretical clarity and generality: it provides a system-wide view of the sensitivity of the objective function with respect to changes in design variables. Since the continuous adjoint equations are independent of any specific numerical discretization, they offer flexibility in handling changes to the underlying mesh or numerical methods. The gradient can thus be computed once and applied consistently, regardless of variations in the solver's configuration. By contrast, the discrete adjoint method takes a different approach. In this method, the governing equations are first discretized, and then the adjoint equations are derived from this discretized system. Essentially, the adjoint equations reflect the numerical approximation of the original PDEs, ensuring that the adjoint formulation is consistent with the solver's numerical implementation. Additionally, many modern solvers support automatic differentiation, which can be used to automatically generate the adjoint equations from the discretized system, making the implementation process easier. Among these two approaches, the discrete adjoint method is the most commonly used, particularly in CFD. The discrete adjoint method's primary advantage lies in its ability to ensure exact consistency between the forward (primal) and adjoint problems. Since the adjoint system is derived directly from the same discretized equations used to solve the forward problem, the gradients it computes are highly accurate and precisely aligned with the numerical solution. This level of accuracy is especially important in complex simulations, where even small discrepancies in the gradient can significantly degrade optimization performance. The discrete adjoint solver mirrors the discretization of the forward solver (mesh, numerical methods, etc.), ensuring that the gradients reflect the actual solution behavior at the discrete level. Furthermore, the discrete adjoint method excels in handling complex, nonlinear systems. Many real-world CFD applications, such as turbulence modeling, shock-wave interactions, and multiphase flows, involve highly nonlinear behaviors. The discrete adjoint method addresses these nonlinearities directly by deriving the adjoint equations from the exact, nonlinear discretized system used in the forward model, ensuring accurate gradient information even in challenging, nonlinear regimes.

4.2.1 Discrete Adjoint Method

During my Ph.D., I chose to implement the discrete adjoint method for the reasons previously discussed. The development of the adjoint equations begins with equations 4.3 and 4.4, where the goal is to derive a formula for the gradient of the objective function with respect to the design variable—in the context of field inversion, this design variable is the correction field, β. The condition in equation 4.4 is fundamental for the derivation of the adjoint equation, indeed given that equation 4.4 enforces the residuals to be zero,

we can define a Lagrangian function, L, by multiplying the residual equation by a multiplier, ψ, which represents the adjoint field. Expanding these equations, we arrive at the following formulation:

$$L = G + \psi^T \cdot R \tag{4.5}$$

Since the Lagrangian function in equation 4.5 is identical to the objective function, because the residuals R of the governing equations are zero, its derivatives will also be the same. Therefore, the following equation holds:

$$\frac{dL}{d\beta} = \frac{dG}{d\beta} + \psi^T \cdot \frac{dR}{d\beta} \tag{4.6}$$

We know that both the objective function and the residuals depend on the conservative flow variables, which in turn are functions of the correction field, as well as the correction field itself. As a result, we arrive at the following comprehensive equation:

$$\frac{dL}{d\beta} = \left(\frac{\partial G}{\partial \tilde{u}} + \psi^T \cdot \frac{\partial R}{\partial \tilde{u}}\right)\frac{\partial \tilde{u}}{\partial \beta} + \frac{\partial G}{\partial \beta} + \psi^T \cdot \frac{\partial R}{\partial \beta} \tag{4.7}$$

The term $\partial\tilde{u}/\partial\beta$ is the most computationally expensive term in equation 4.7. However, since the residuals R are zero, the adjoint variable ψ can be freely chosen. This flexibility allows avoiding the computational cost associated with $\partial\tilde{u}/\partial\beta$ by selecting ψ such that the expression within the brackets vanishes, as shown below:

$$\left[\frac{\partial R}{\partial \tilde{u}}\right]^T \cdot \psi + \left[\frac{\partial G}{\partial \tilde{u}}\right]^T = 0 \tag{4.8}$$

The set of equations in 4.8 represents the discrete adjoint equation, where $\partial R/\partial\tilde{u}$ is the Jacobian matrix associated with the RANS equations. Solving this linear system yields the adjoint field, ψ. The cost of constructing this equation primarily depends on calculating the derivative of the objective function with respect to the conservative variables, which can be done either manually or through automatic differentiation techniques. Additionally, the Jacobian matrix is readily available 'for free' when an implicit time-stepping scheme is used for the simulation. Once the linear system is solved, and given that the Lagrangian function is equivalent to the objective function, the gradient of the objective function with respect to the correction field can be easily evaluated using the following equation:

$$\frac{dG}{d\beta} = \frac{\partial R}{\partial \beta} \cdot \psi^T + \frac{\partial G}{\partial \beta} \tag{4.9}$$

From equation 4.8, it can be understood that the adjoint field is associated with each of the RANS equations and provides insights into how changes in the design variables impact the objective function. This allows us to assess the sensitivity of the objective function to modifications in the governing equations, helping to identify where and to

what extent adjustments in the equations will influence the overall performance of the system. Finally, equation 4.9 provides the gradient of the objective function, which is used to update the correction field, β. In a simple gradient descent method, the update is performed as follows:

$$\beta^{k+1} = \beta^k - \eta \frac{dG}{d\beta} \tag{4.10}$$

In equation 4.10, k represents the iteration number of the optimization process, which also corresponds to the simulation run since the process must be performed at a stationary state. The parameter η is the step size, a characteristic parameter of the optimization algorithm.

4.2.2 Addressing Convergence Challenges in the Discrete Adjoint Method

Traditionally, the discrete adjoint method involves directly solving a linear system. While effective, this approach can encounter convergence challenges in larger or more complex problems, particularly those involving viscous or three-dimensional flows. Additionally, the process often demands high memory usage and can become inefficient as the number of Krylov subspaces grows with each optimization iteration. To address these challenges, a promising alternative is the use of pseudo-time stepping for solving adjoint equations, as developed by Nielsen et al. in [63]. Their work builds on contributions by Giles [31], where adjoint solutions for the Euler and Navier-Stokes equations are computed using an explicit Runge-Kutta scheme. This approach leverages the duality between flow and adjoint systems, allowing the adjoint field to evolve in pseudo-time. The exact dual algorithm mirrors the algorithm used for solving the nonlinear primal equations. By applying the same time-marching techniques from the primal equations, this method offers flexibility in timestep selection. Unlike physical time, the pseudo-time timestep can be arbitrarily chosen, enabling both fixed and adaptive timestep strategies. This method not only addresses convergence issues but also effectively handles numerical transients, making it a robust solution for complex adjoint problems. As a result, Equation 4.8 can be rewritten as follows:

$$\Delta V \frac{\partial \psi}{\partial \tau} + \left[\frac{\partial R}{\partial \tilde{u}}\right]^T \psi + \left[\frac{\partial G}{\partial \tilde{u}}\right]^T = 0 \tag{4.11}$$

In Equation 4.11, τ represents the pseudo-time, and ΔV denotes the discrete cell volumes of the mesh. By applying the backward Euler method to solve this equation, the following result can be obtained:

$$\left(\frac{\Delta V}{\Delta \tau} I + \frac{\partial R}{\partial \tilde{u}}\right) \Delta \psi^n + \left[\frac{\partial R}{\partial \tilde{u}}\right]^T \psi^n + \left[\frac{\partial G}{\partial \tilde{u}}\right]^T = 0 \tag{4.12}$$

The construction of Equation 4.12 incurs no additional computational cost compared to 4.8, as the Jacobian matrix is already available from the solution of the primal equations, assuming the same time-stepping scheme is used. Consequently, the update at time step n simplifies to an explicit matrix-vector product. This approach was also employed in the Ph.D. thesis of Pini [66], where he applied the adjoint method for the design optimization of turbomachinery.

Once the gradient is computed, the next step is to update the correction field as part of the optimization process. Various gradient-based optimization algorithms can be employed for this task. Initially, I experimented with a simple gradient descent method due to its straightforward implementation. However, I found that this approach, while effective, was not as efficient in terms of speed during the optimization process. To improve performance, I opted for a more advanced method, specifically a modified version of the Gauss-Newton method, as detailed in the work of Brenner et al. [9]. This method provided a better convergence rate and overall efficiency, making it more suitable for this particular problem. The process of updating the correction field is illustrated by the following equations:

$$\beta^{k+1} = \beta^k - \omega \frac{G^k - G_{tol}}{\left.\frac{dG}{d\beta}\right|^k \cdot \left.\frac{dG}{d\beta}\right|^k} \left.\frac{dG}{d\beta}\right|^k \tag{4.13}$$

where ω represents the under-relaxation factor, $\omega \in [0,1]$, and G_{tol} is a tolerance threshold for the goal function.

4.3 Mapping Function

As demonstrated in 4.1 and 4.2, the production term is multiplied by $h(\beta)$ rather than being adjusted directly by β. The mapping function is sometimes employed to constrain the values of the correction field. In Duraisamy's original work, the correction field is applied directly without any mapping; however, in other studies, such as those by Ferrero et al. [24] and Brenner et al. [9], a mapping function is utilized. The purpose of the mapping function is to prevent excessively high values of the correction field or to avoid negative values. In this work, the mapping functions used in both cited studies have been implemented.

Before proceeding with the details of the implemented mapping functions, it is important to note that in Equation 4.9, the first term on the right-hand side includes the partial derivative of the residual with respect to the correction field. Since a general mapping function $h(\beta)$ is often used, this derivative is computed as follows:

$$\frac{\partial R}{\partial \beta} = \frac{\partial R}{\partial h}\frac{dh}{d\beta} \tag{4.14}$$

As previously explained in Section 4.1, the correction field influences only the production term of the Spalart-Allmaras equation, leading to $\partial R/\partial h = P$. Additionally, the choice of mapping function $h(\beta)$ directly affects the overall gradient of the objective function, which in turn influences how the correction field is updated during the optimization process.

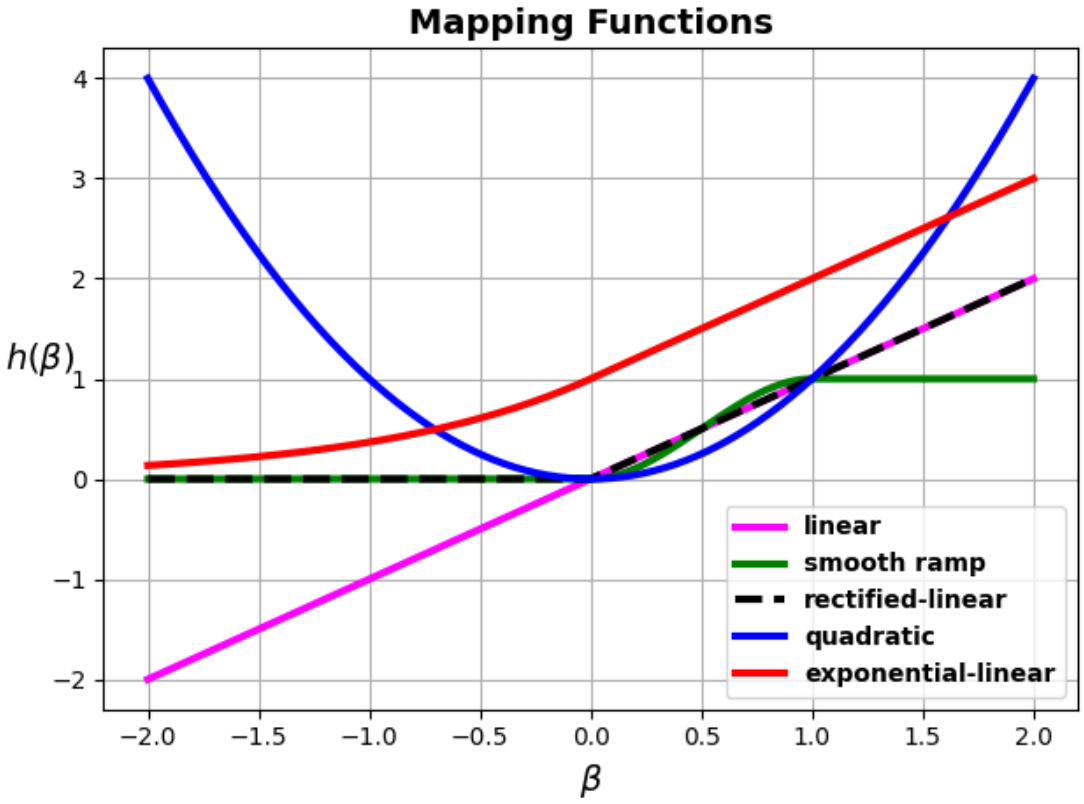


Figure 4.2: Different mapping functions implemented.

The five different mapping functions implemented are illustrated in Figure 4.2. It is important to note that the linear mapping function is not a true mapping function since it is equivalent to the correction field itself; however, for the sake of a general implementation, it is considered a mapping function here. The other four mapping functions exhibit similar behavior concerning the negative values of β, effectively constraining the correction field to be positive. The equation for the rectified linear mapping function is expressed as follows:

$$h(\beta) = \begin{cases} \beta & : \beta \geq 0 \\ 0 & : \beta < 0 \end{cases} \tag{4.15}$$

This function essentially disables the production term when the optimization process

predicts a negative value for the correction field. In this scenario, the intention is for the optimization process to signal the destruction of turbulence by inverting the sign of the production term. Consequently, this function allows the original destruction term to take over this responsibility while keeping the production term equal to zero. For this particular mapping function, regions of the domain where the correction field has negative values will remain unchanged, staying at zero. This can be understood by considering the derivative of the mapping function, which appears in Equation 4.14. If we assume that the second term on the right-hand side of Equation 4.9 is equal to zero, details of which will be provided in the next section, this leads to no changes in the correction field in these regions. In contrast, a different behavior is observed when using a quadratic mapping function.

$$h(\beta) = \beta^2 \tag{4.16}$$

In this case, negative values of the correction field would increase turbulence production. This function lacks a solid physical rationale and may also introduce numerical instability when the correction field exceeds the range $[-1,1]$. The smooth ramp function behaves like the rectified linear functions for negative values of β, while also limiting positive values. It effectively constrains the model to transition from a laminar state to a fully turbulent state, where the laminar condition corresponds to zero production and the fully turbulent state is represented by the original production term of the SA equation. The equations are shown as follows:

$$h(\beta) = \begin{cases} 0 & : \beta < 0 \\ 3\beta^2 - 2\beta^3 & : 0 \leq \beta \leq 1 \\ 1 & : \beta > 0 \end{cases} \tag{4.17}$$

Similar to the rectified linear equation 4.15, this function does not permit changes in regions of the domain with negative values of the correction field, nor in regions where the correction field exceeds one.

Finally, the exponential-linear mapping function is the one used in [15].

$$h(\beta) = \begin{cases} \beta + 1 & : \beta \geq 0 \\ \exp(\beta) & : \beta < 0 \end{cases} \tag{4.18}$$

This function causes the production term to asymptotically approach zero for negative values of the correction field while maintaining the same behavior as a linear function for positive values of β. For positive values of β, the function is defined as $\beta + 1$ to ensure continuity at $\beta = 0$, where $h(\beta) = \exp(\beta)$. This function demonstrates good performance because it allows the correction field to vary even in regions where it is negative while maintaining small values.

4.4 Objective Function for Field Inversion Optimization

The optimization problem outlined in Equation 4.3 can be effectively addressed once the goal or objective function G is established. This objective function serves as the cornerstone of the optimization process in field inversion, playing a critical role in identifying physically consistent corrections. Therefore, it is essential that this function satisfies some key criteria:

- Ease of Measurement: The objective function should be straightforward to measure with adequate precision. This ensures that the optimization process can rely on accurate data, which is vital for achieving reliable results. Inaccurate measurements can lead to erroneous conclusions, undermining the entire optimization effort.
- Relevance to System Performance: The objective function must have a strong correlation with the overall performance of the system. A well-defined function will directly reflect the critical parameters and behaviors that determine the system's efficiency and effectiveness.
- Sensitivity to Corrections: The objective function should exhibit a sensitivity to the corrections introduced by the field inversion process. This means that even small adjustments in the correction field should result in noticeable changes in the objective function. Such sensitivity allows for a more responsive optimization process, facilitating the identification of the most effective corrections to enhance system performance.

The fundamental concept is that the objective function should effectively represent the RANS error associated with variables that can be readily measured or evaluated through scale-resolving simulations. Such variables might include force coefficients, pressure distributions, or skin friction coefficients. By focusing on these measurable quantities, the optimization process can leverage reliable data that reflects the system's performance. This measured information serves a crucial role in determining corrections for other variables that are typically more challenging to assess directly in experimental settings. For instance, while quantities like turbulent eddy viscosity are pivotal for accurately modeling fluid behavior, they cannot be directly measured in an experiment. Instead, the corrections derived from the objective function can provide insights into how to modify these less accessible variables, thus enhancing the fidelity of the computational model. An example of a goal function based on the pressure coefficient distribution is shown below:

$$G = \int_{\partial\Omega} (Cp - Cp_{exp})^2 dl \tag{4.19}$$

The term on the right-hand side of Equation 4.19 represents a line integral conducted along the wall, which may correspond to the surface of an airfoil. This integral evaluates the norm-2 error in the wall pressure distribution. By calculating this error metric, we can quantify how well the computational model predicts the pressure distribution along the airfoil's surface compared to reference data or experimental measurements. A lower norm-2 error indicates a more accurate representation of the physical phenomenon, leading to enhanced reliability in predictions of aerodynamic forces and moments. It is important to recognize that experimental results may be subject to uncertainties that can significantly affect the optimization process. In a previous study conducted by Ferrero et al. [26], a sensitivity analysis was performed by perturbing the experimental data to identify the regions within the computational domain where these uncertainties have the most significant impact on the corrections derived from field inversion. This method enables researchers to identify critical areas where inaccuracies in experimental data can lead to significant deviations in the optimization outcomes. By systematically analyzing how variations in experimental measurements affect the optimization results, this approach provides valuable insights into the robustness of the field inversion procedure. Moreover, it facilitates the establishment of accuracy requirements for the reference experimental data, ensuring that such data meet the necessary standards for reliable application in field inversion.

Generally, the goal function is augmented with an additional component known as the penalization term, a technique commonly referred to as Tikhonov regularization. This term plays a crucial role in mitigating the occurrence of extreme values in the correction field during the optimization process. By imposing a penalty on large corrections, Tikhonov regularization ensures that the correction field remains active only in regions where it can make a meaningful contribution to reducing the overall error. Furthermore, this regularization technique encourages the highest values of the correction field β, to remain close to the initial value β_0, which corresponds to the original model. This constraint effectively prevents overfitting to noise or inaccuracies in the data, promoting a more physically realistic adjustment of the model. The objective function can be expressed mathematically by incorporating the penalization term, leading to the following formulation:

$$G = \int_{\partial\Omega} (Cp - Cp_{exp})^2 dl + \lambda \int_{\Omega} (\beta - \beta_0)^2 d\Omega \tag{4.20}$$

In the Equation 4.20, the second term on the right-hand side is a surface integral over the computational domain Ω where λ is the penalization term. This second term penalizes the goal function when the correction field deviates significantly from its initial value β_0.

One of the main advantages of the field inversion approach is its inherent flexibility in optimizing the correction field by defining a target function that can be tailored to

different objectives. This adaptability allows the optimization of different aspects of the flow, whether it is a global performance metric or specific local characteristics. In this thesis, I experimented with several different objective functions to assess their impact on the optimization process. Specifically, I focused on skin friction distribution, pressure distribution, lift coefficients and drag coefficients as different objectives. Global objective functions based on integral quantities are easier to measure, but pressure and skin friction distributions can provide more physical insight.

4.5 Field Inversion Optimization Results

The results of the optimization process using field inversion are presented in the following subsections. The test cases analyzed are the same as those introduced in Section 3.3. Here, the original SA model is compared with the augmented version incorporating the learned correction field. Different objective functions were used depending on the available experimental or high-fidelity data. As previously mentioned, two types of goal functions were considered: one based on integral variables and the other on local variables. Finally, a multi-objective optimization approach is demonstrated for the NACA0012 test case.

4.5.1 Transitional Flat Plate

The results obtained for the transitional flat plate by using the original SA model and the SA BCM model were presented, showing that neither model provided a fully satisfactory match with the experimental data. However, the SA BCM model showed a closer agreement with experimental results compared to the original SA model. In this section, the results obtained after applying the field inversion optimization procedure are presented. Section 4.3 introduced various mapping functions, highlighting their differences. For this test case, a simple rectified linear mapping function was employed. This choice was made to enforce a laminar region, represented by a null correction field value, effectively suppressing the production term.

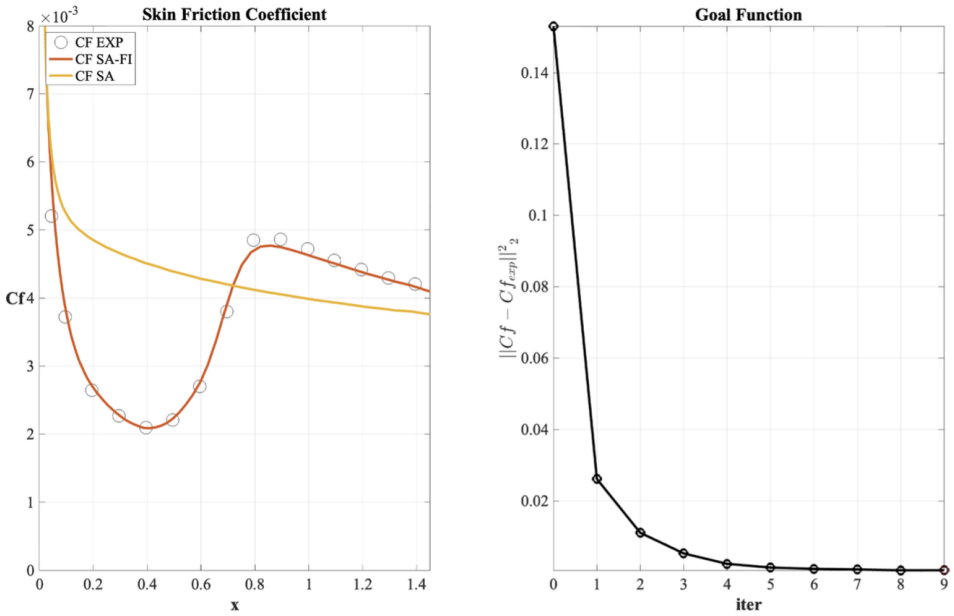


Figure 4.3: Comparison of experimental results, the original SA model, and the augmented SA model. The left panel shows the skin friction coefficient, while the right panel illustrates the loss function evolution over optimization iterations.

Figure 4.3 compares the skin friction coefficient predictions for the T3A case from the augmented SA model, the original SA model, and experimental data. The right panel of the figure shows the progressive reduction of the objective function throughout the optimization iterations. Notably, the augmented SA model exhibits strong agreement with experimental results, surpassing even the SA BCM model in accuracy. These results confirm that the correction field effectively modulates the production term to accurately capture the transition phenomenon.

4.5.2 Backward-Facing Step

As discussed in Section 3.3.2, for the backward-facing step test case, the skin friction coefficient and the pressure coefficient are considered as key objectives. The analysis focuses on the region near the step and the area immediately downstream. Separation is induced at the step due to the sudden expansion, with the flow reattaching further downstream. This phenomenon is evident in the skin friction coefficient, which exhibits negative values in a small region after the step, indicating separation. However, as shown in Figures 4.4 and 4.6, the SA model fails to accurately predict the separation region, underestimating both Cf and Cp. This discrepancy not only affects the separation zone but also influences the accuracy of predictions in the reattachment region.

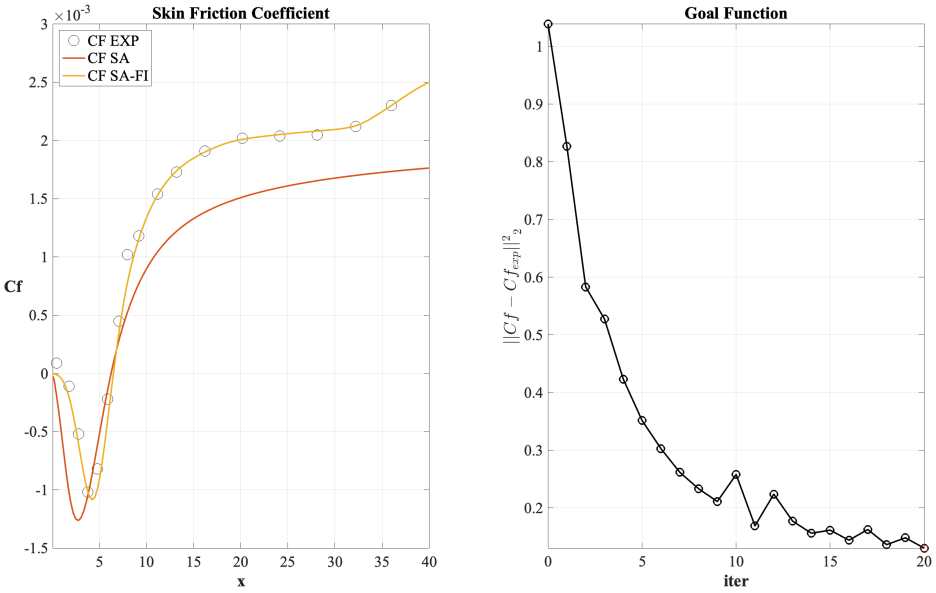


Figure 4.4: Comparison of experimental results, the original SA model, and the augmented SA model. The left panel shows the skin friction coefficient, while the right panel illustrates the loss function evolution over optimization iterations.

The optimization of the SA model through field inversion enables the model to better predict and calibrate turbulence production in specific regions of the domain by targeting a defined objective. Figure 4.4 illustrates the improvement (shown by the yellow line) achieved by incorporating the optimized correction field. The final correction field allows the model to closely match the experimental results. On the right side of Figure 4.4, the loss function values throughout the optimization iterations are presented. A reduction of almost one order of magnitude in the loss is achieved after 20 iterations. Each iteration corresponds to a CFD simulation; however, after the first simulation, subsequent iterations require less time as they start from the previously converged solution.

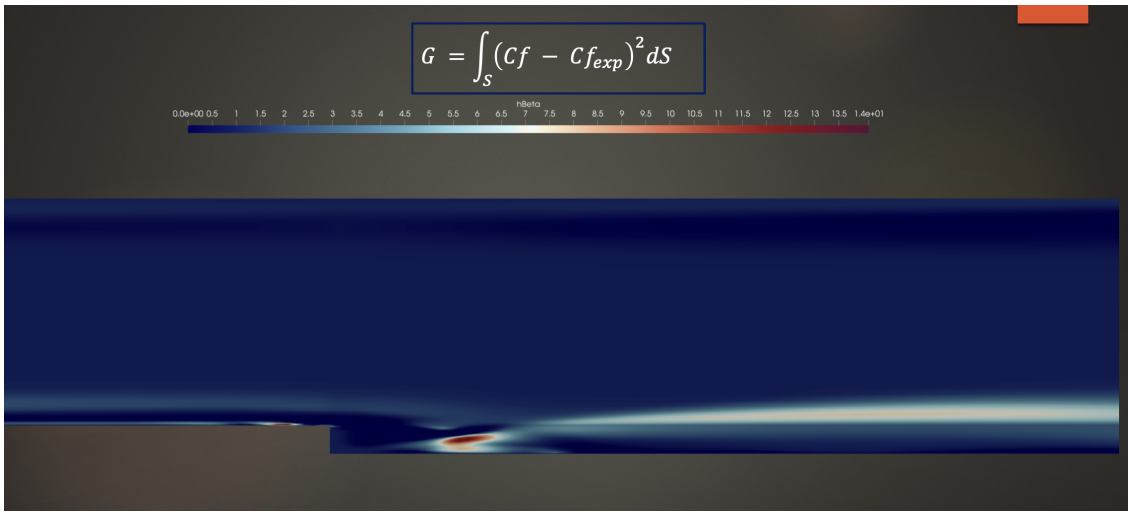


Figure 4.5: Correction field derived from the optimization of the skin friction coefficient.

In Figure 4.5, the correction field is displayed across the domain. Since the objective function is influenced by the lower wall, the correction field exhibits values that differ from the initial ones near the wall. The value next to the step is zero, indicating no turbulence production, while further downstream, the correction field increases the turbulence production beyond the original model. This is clearly reflected in the skin friction coefficient, where the value rises in the reattachment region to better align with the experimental data.

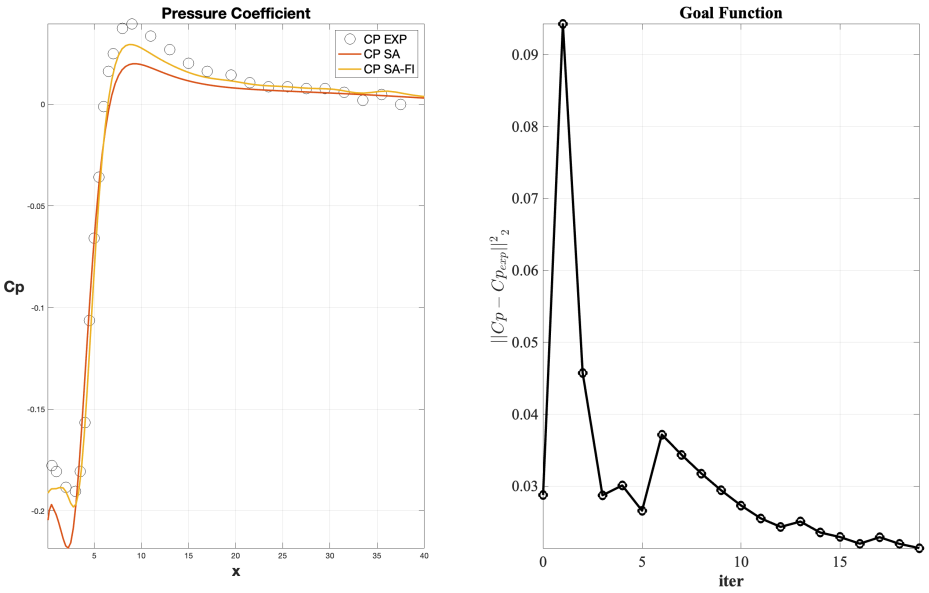


Figure 4.6: Comparison of experimental results, the original SA model, and the augmented SA model. The left panel shows the pressure coefficient, while the right panel illustrates the loss function evolution over optimization iterations.

A similar discussion can be made for the pressure coefficient. However, in this case, the optimization did not perform as well as it did for the skin friction coefficient. As seen in Figure 4.6, the pressure coefficient shows some improvement, but it does not match the experimental results as closely as the optimization of the skin friction coefficient. Additionally, the loss function, shown on the right of Figure 4.6, exhibits oscillations during the iterations. The L2 error also starts with a lower value compared to the L2 error in the skin friction coefficient, indicating that the optimization for the pressure coefficient was less effective.

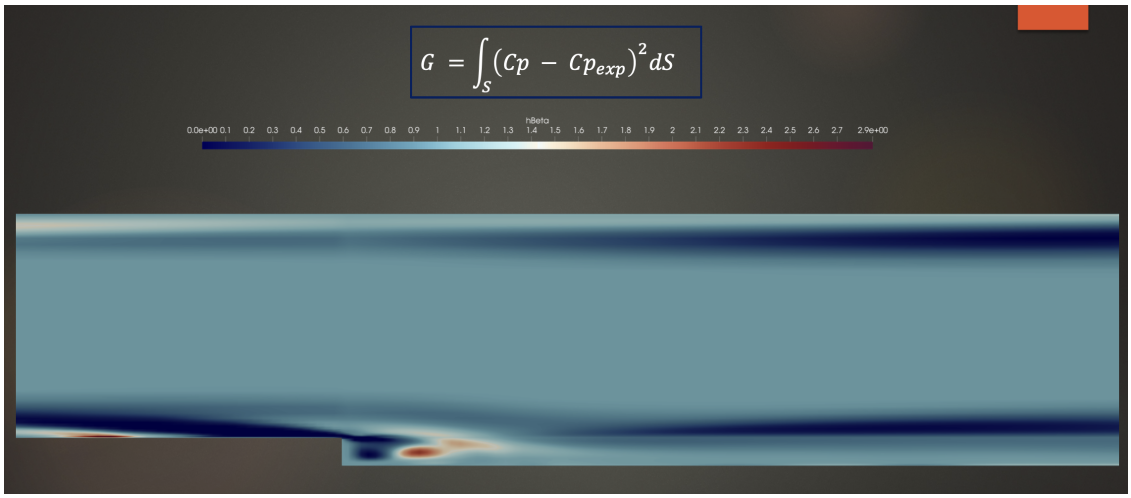


Figure 4.7: Correction field derived from the optimization of the pressure coefficient.

Figure 4.7 shows the correction field predicted in this case. Similar to the optimization of the skin friction coefficient, there is a significant variation in the lower wall where the objective is defined. However, both the shape and resulting values of the correction field differ, indicating that the two objectives are independent of each other.

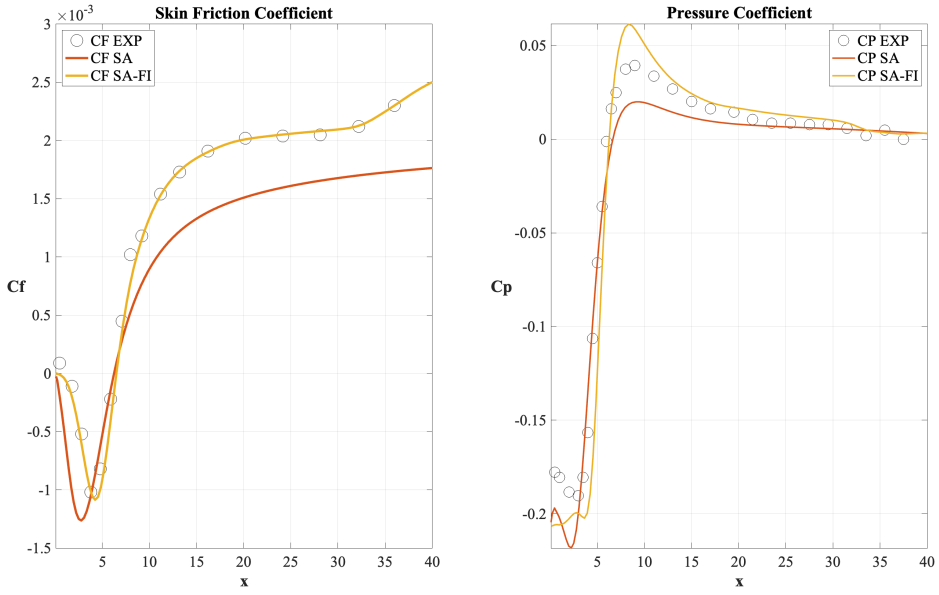


Figure 4.8: Comparison of experimental data, the original SA model, and the augmented SA model for Cf and Cp, optimized based on the Cf.

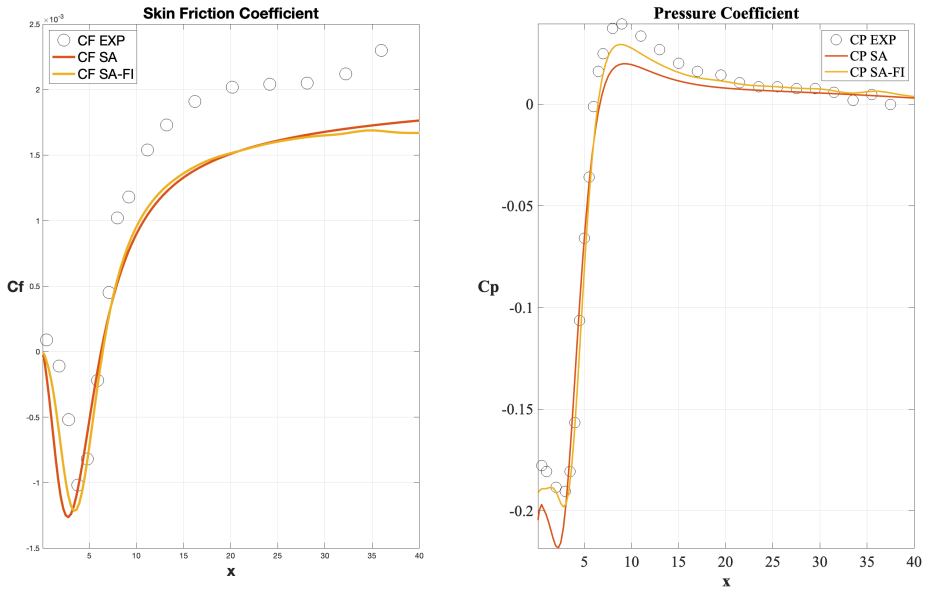


Figure 4.9: Comparison of experimental data, the original SA model, and the augmented SA model for Cf and Cp, optimized based on the Cp.

Figure 4.9 demonstrate the independence of the objective functions. Specifically, optimizing for Cp does not lead to any noticeable improvement in the prediction of Cf. This is evident in the left panel of Figure 4.9, where Cf remains nearly unchanged compared to the original model. However, when optimizing for Cf, the right panel of Figure 4.8 shows a slight improvement in Cp downstream of the step relative to the original SA model. Overall, for this test case, using Cf as the objective function appears to be more effective than using Cp.

4.5.3 SD7003 Airfoil

As previously mentioned, various high-fidelity data are available for the SD7003 test case, including the wall pressure distribution (Cp), skin friction distribution (Cf), lift coefficient (C_L), and drag coefficient (C_D). These provide the basis for defining four distinct objective functions. Each of these functions was tested, and the results were compared across different optimizations. Since each optimization modifies the turbulent viscosity field in a unique manner, all other fields are consequently affected. The differences are visible in Figure 4.10 below.

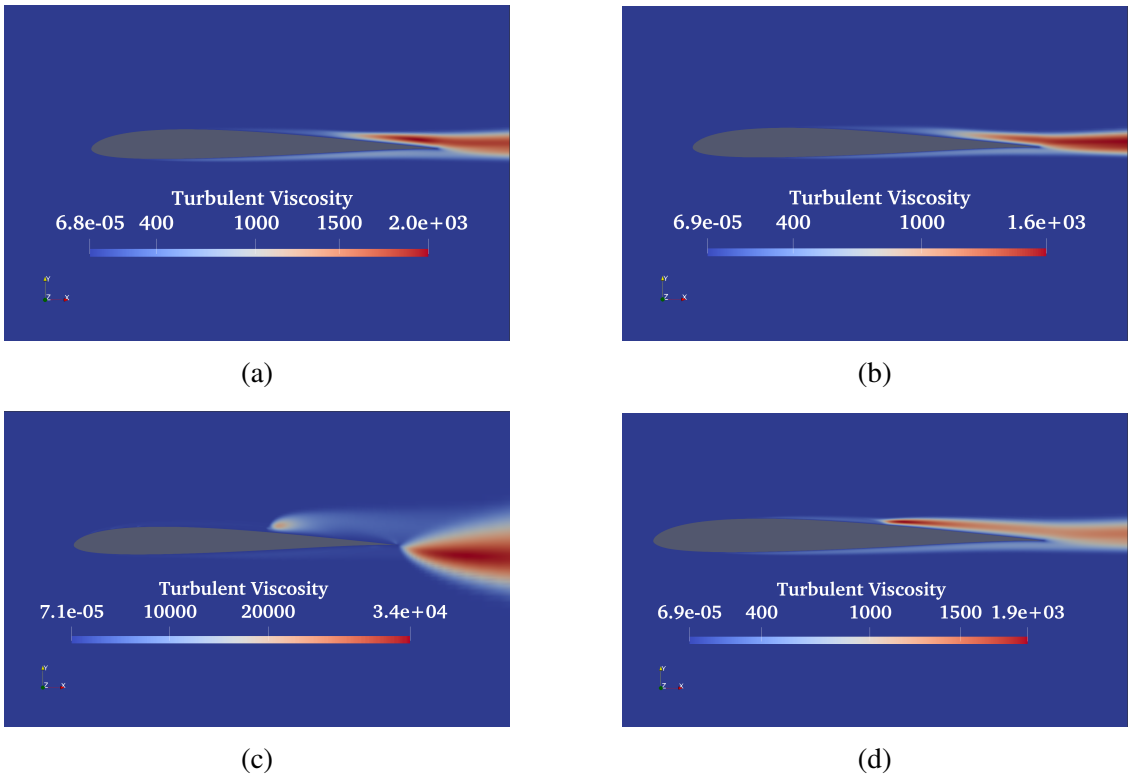


Figure 4.10: Turbulent viscosity results after field inversion optimization for the SD7003 airfoil at $\alpha = 4°$. (**a**) C_L optimization, (**b**) C_D optimization, (**c**) Cp optimization, and (**d**) Cf optimization.

In Figure 4.10 (a) and (b) , optimizing with C_L and C_D appears to produce a similar effect on the resulting turbulent viscosity field. In contrast, optimizing with Cf and Cp (Figure 4.10 (c) and (d)) generates a completely different turbulent viscosity distribution, highlighting the distinct influence of local and integral objective functions on the optimization process.

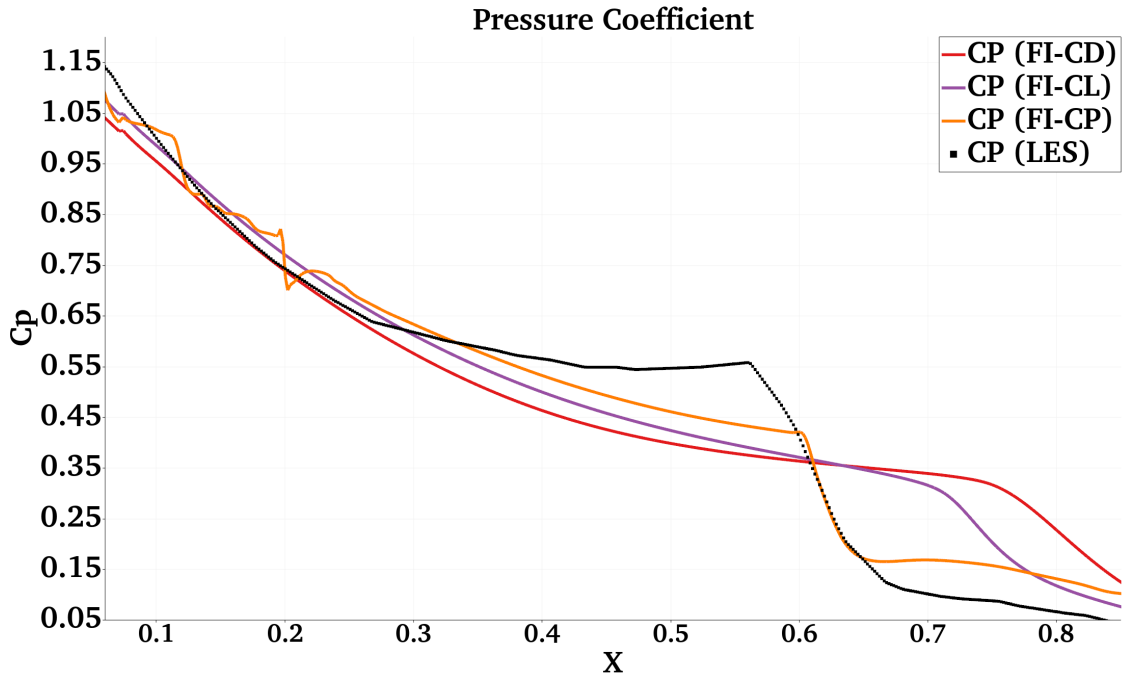


Figure 4.11: Comparison of pressure coefficients for different goal function definitions for SD7003 at $\alpha = 4°$.

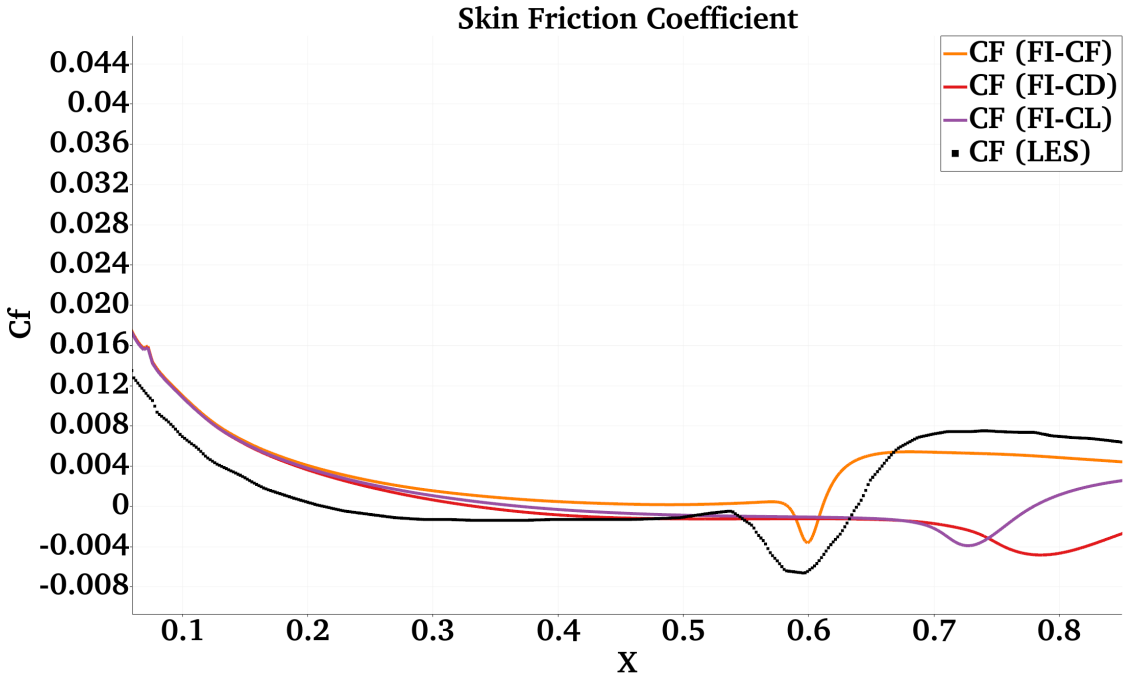


Figure 4.12: Comparison of skin friction coefficients for different goal function definitions for SD7003 at $\alpha = 4°$.

This test case serves as an ideal benchmark for evaluating the influence of different types of objective functions, whether integral or local, as discussed in Section 4.4. High-fidelity results from LES are represented by black dots, and as seen in Figures 4.11 and 4.12, different objective functions lead to distinct Cf and Cp distributions. Specifically, the red and purple curves, which correspond to optimizations based on C_D and C_L respectively, fail to capture the correct shape of the high-fidelity results. In contrast, while the optimizations based on Cf and Cp do not perfectly align with the high-fidelity values, they successfully approximate the shape and the location of the separation region.

	CL	**CD**
LES [29]	0.59	0.021
SA	0.551	0.0099
SA (FI−CL)	0.586	0.0147
SA (FI−CD)	0.571	0.0191
SA (FI−CP)	0.571	0.0052
SA (FI−CF)	0.602	0.011

Table 4.1: Comparison of lift and drag coefficients for SD7003 airfoil at $\alpha = 4°$.

Table 4.1 presents the results for C_L and C_D obtained from each optimization. While almost all optimizations show an improvement compared to the original SA model, the best results in terms of C_L and C_D are naturally achieved when the optimization is guided by an integral objective. However, Figures 4.11 and 4.12 reveal that, despite capturing the integral values, the location of the separation phenomenon is not accurately predicted. This is expected, as C_L and C_D are defined as the area under the curves of Cf and Cp, meaning that different distributions of these coefficients can still yield the correct integral values. As a result, the optimization process is susceptible to multiple local minima.

The same analysis is performed for $\alpha = 8°$. Figures 4.13, 4.14, and 4.15, along with Table 4.2, highlight the differences in turbulent viscosity, as well as the variations in local and global coefficients.

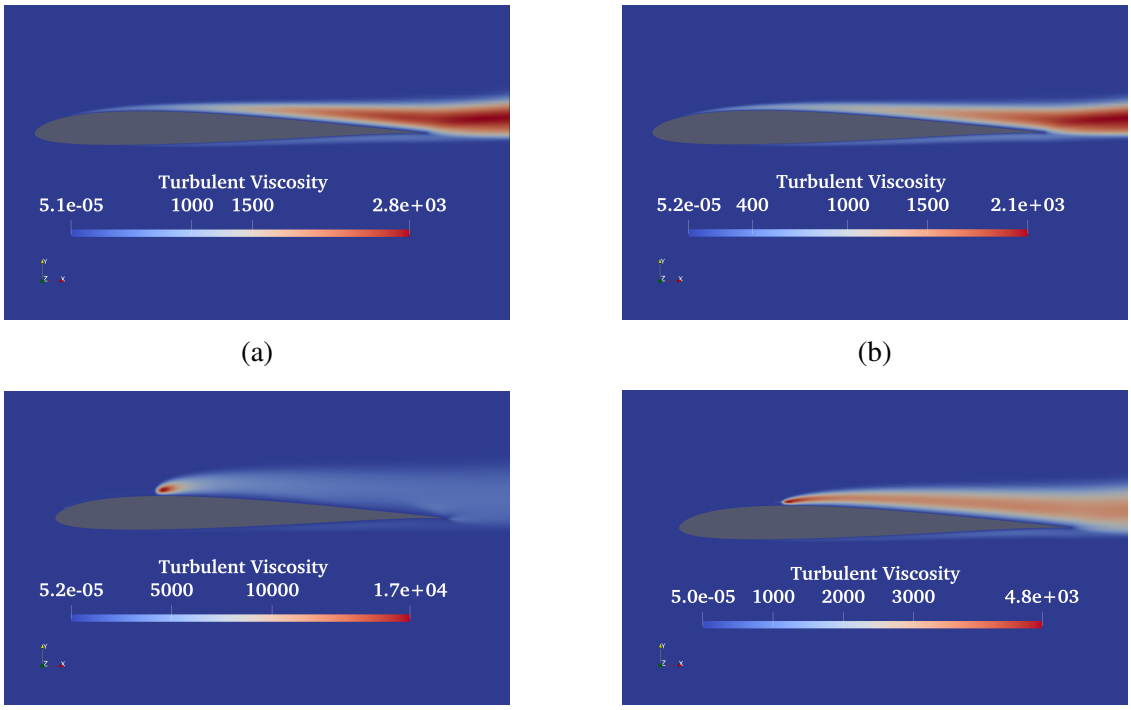


Figure 4.13: Turbulent viscosity results after field inversion optimization for the SD7003 airfoil at $\alpha = 8°$. (**a**) CL optimization, (**b**) CD optimization, (**c**) CP optimization, and (**d**) CF optimization.

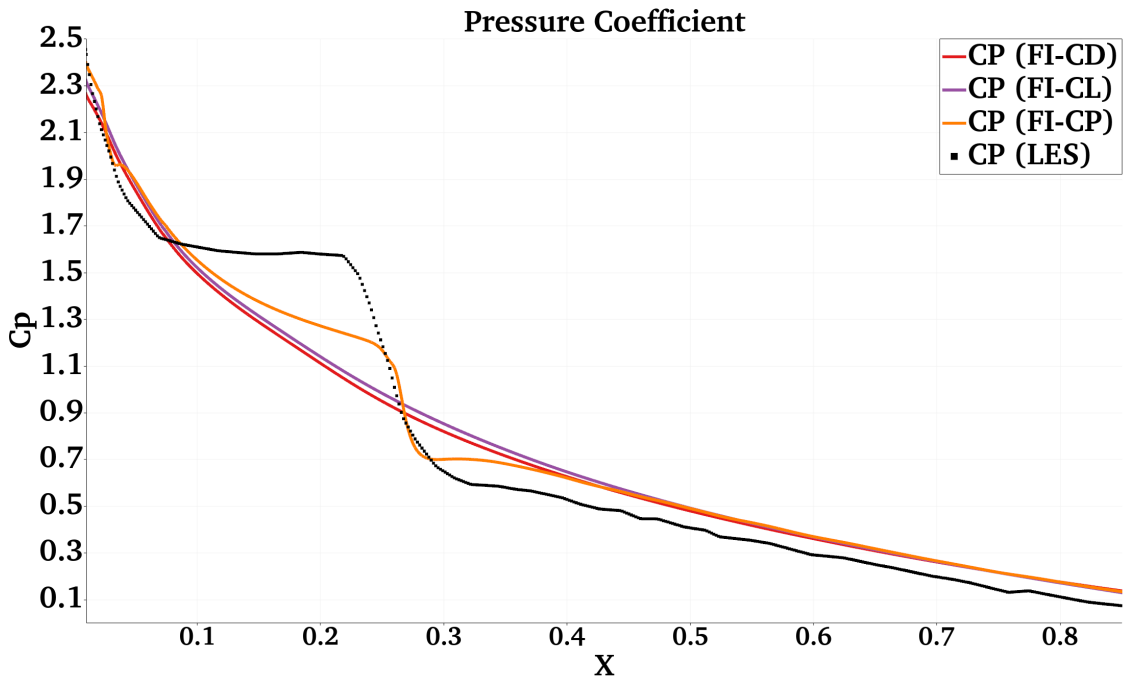


Figure 4.14: Comparison of Pressure coefficient for different goal function definitions for SD7003 at $\alpha = 8°$.

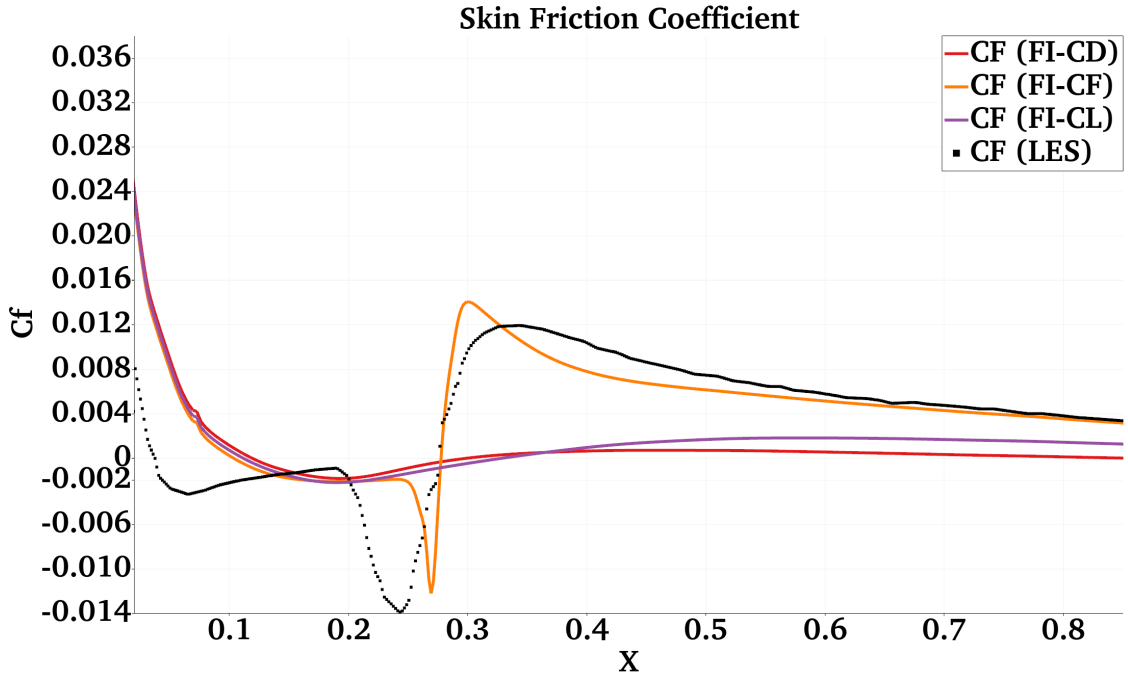


Figure 4.15: Comparison of Skin friction coefficient for different goal function definitions for SD7003 at $\alpha = 8°$.

	CL	CD
LES [29]	0.92	0.043
SA	0.895	0.0435
SA (FI−CL)	0.917	0.0436
SA (FI−CD)	0.894	0.044
SA (FI−CP)	0.939	0.0372
SA (FI−CF)	0.935	0.0411

Table 4.2: Comparison of lift and drag coefficients for SD7003 airfoil at $\alpha = 8°$.

From both angles of attack, the same effects were observed. The use of objective functions based on the error in global coefficients yields good results for global coefficient prediction but fails to accurately capture the correct shape distributions, missing important flow features such as separation. In contrast, the use of local distributions in the objective function provides more physical insights and results in corrected fields that are closer to the reference ones.

4.5.4 NACA0021 Airfoil

Continuing the discussion on the low-Reynolds airfoil, the results of the field inversion optimization for the NACA0021 airfoil are presented and analyzed here. As previously discussed in Section 3.3.4, this test case benefits from various experimental datasets at different angles of attack, particularly for C_L and C_D. The significance of this test case has already been outlined in Section 3.3.4. Both C_L and C_D were used to evaluate the performance of the Field Inversion framework, and the corresponding results are presented in the current section.

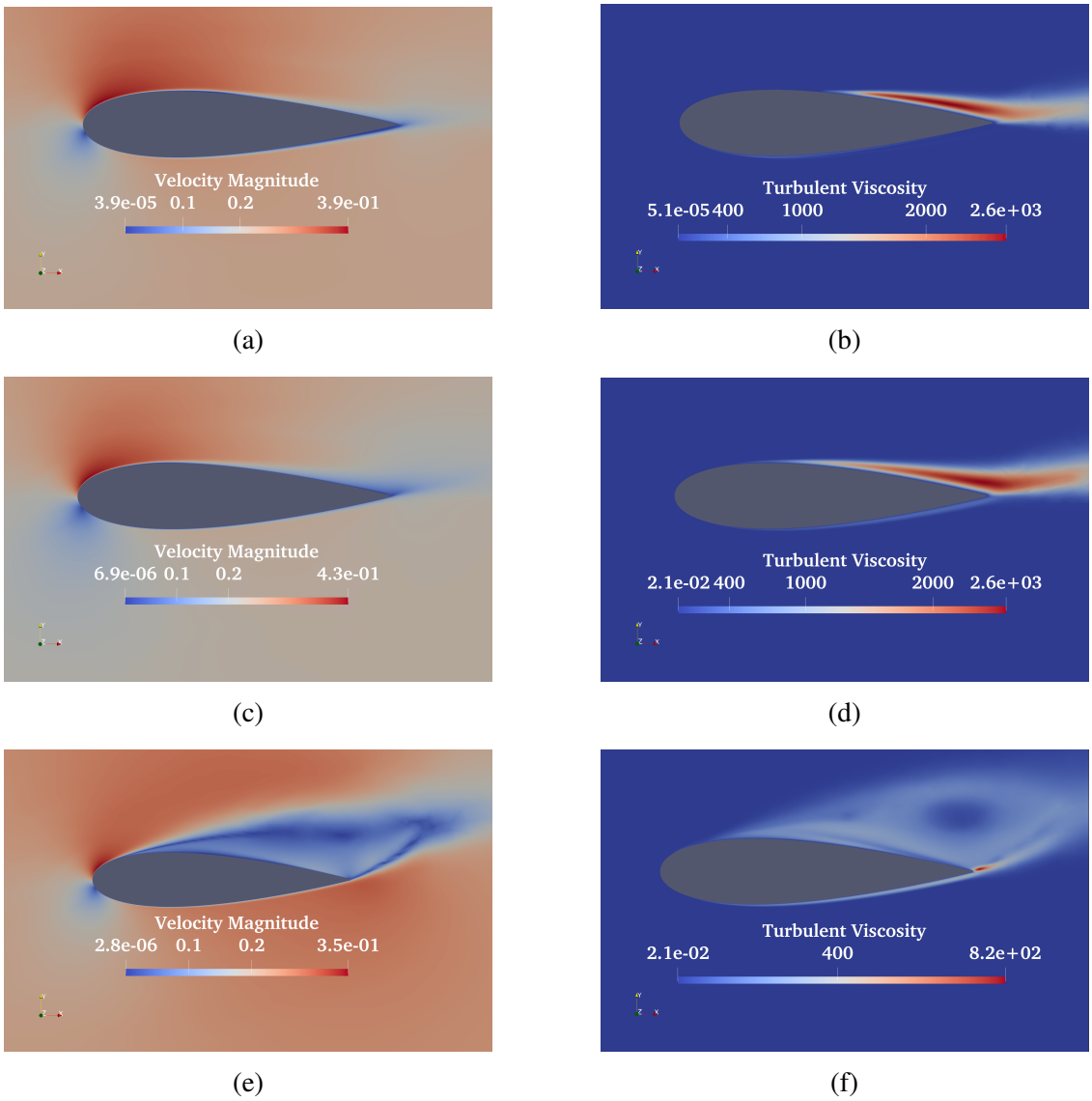


Figure 4.16: Results of the C_L Augmented Spalart–Allmaras model for the NACA0021 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 8°$. (**c**) Velocity magnitude and (**d**) turbulent viscosity at $\alpha = 12°$. (**e**) Velocity magnitude and (**f**) turbulent viscosity at $\alpha = 16°$.

Figure 4.16 presents the optimization results in terms of velocity and turbulent viscosity fields at different angles of attack, specifically at $\alpha = 8°, 12°$ and $16°$, corresponding to pre-stall, incipient stall, and post-stall conditions. A significant difference is observed in both the velocity and turbulent viscosity fields at $\alpha = 16°$ when compared to the results of the original SA model shown in Section 3.3.4. After applying the optimal correction field, the modified turbulent viscosity field leads to a more pronounced flow separation

in the velocity field, which was almost absent in the original SA model. These effects are even more evident in Figure 4.17.

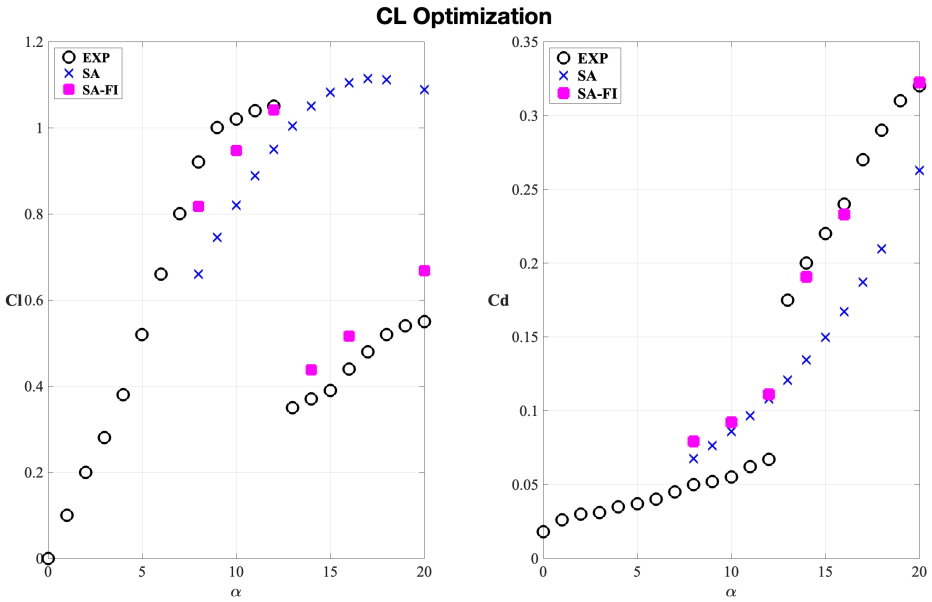
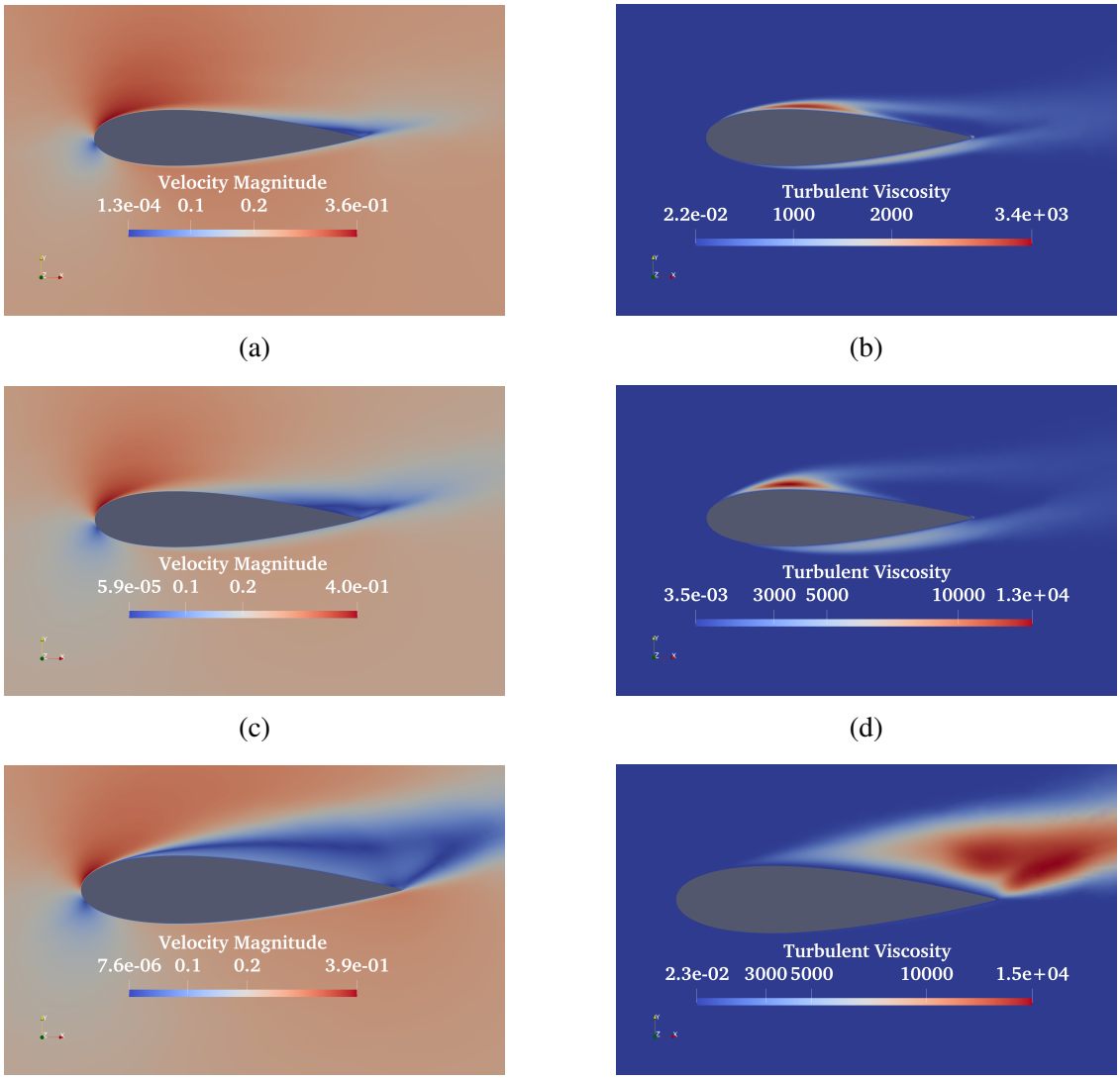


Figure 4.17: Comparison of C_L and C_D across different angles of attack for experimental data, the original SA model, and the augmented SA model optimized for C_L.

Figure 4.17 shows a significant improvement in terms of C_L, both in the pre-stall and post-stall regions, compared to the original SA model. The C_D also improves, but mainly in the post-stall region, while it remains very close to the original SA model in the pre-stall region. The same analysis was performed by optimizing C_D, and the results are presented below.



(a) (b) (c) (d) (e) (f)

Figure 4.18: Results of the C_D Augmented Spalart–Allmaras model for the NACA0021 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 8°$. (**c**) Velocity magnitude and (**d**) turbulent viscosity at $\alpha = 12°$. (**e**) Velocity magnitude and (**f**) turbulent viscosity at $\alpha = 16°$.

Figure 4.18 illustrates the effect of the correction field in terms of the turbulent and velocity fields across different angles of attack. In contrast to the C_L optimization, the turbulent viscosity field adopts a different shape compared to the original SA model, even in the pre-stall region.

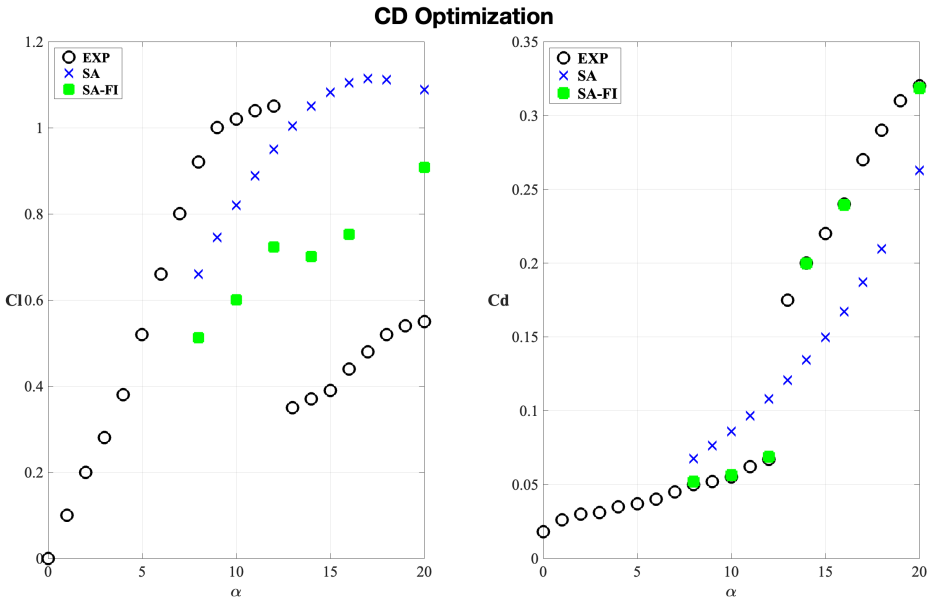


Figure 4.19: Comparison of C_L and C_D across different angles of attack for experimental data, the original SA model, and the augmented SA model optimized for C_D.

Figure 4.19 provides a clearer understanding of the results in terms of the integral coefficients. It is evident that, in this case, a perfect match in terms of C_D is achieved. However, the predicted C_L values are completely inaccurate. As highlighted in the analysis of the backward-facing step test case, optimizing for one objective does not necessarily lead to better results for another. Indeed, in this case, the C_D optimization results in erroneous C_L values, which are actually worse than those produced by the original SA model, particularly in the pre-stall region. From Figures 4.17 and 4.19, it is evident that the two objectives are in contrast in the pre-stall region. Optimizing for C_L improves C_L in the pre-stall region but does not lead to any improvement in C_D. Conversely, optimizing for C_D improves C_D, but it deteriorates the C_L prediction in the pre-stall region. With this in mind, the next section will discuss a trial aimed at obtaining better results in the pre-stall region by using a multi-objective optimization algorithm.

4.5.5 Multi-Objective Optimization

The NACA0021 test case revealed contrastive behavior of the objective functions in the pre-stall region. For this reason, a multi-objective optimization approach was employed to explore whether the results in this region could be improved. Among the various techniques available, an approach known as Adaptive Weights using the concept of the Hyperplane technique was chosen. This framework was introduced by Ryu et al. in [77]. The idea behind their work comes from the fact that the weighted multi-objective

function, despite being one of the most widely used methods for solving multi-objective problems, presents the challenge of determining the optimal set of weights for combining the different objectives. The process of selecting the best set of weights is crucial, as it directly influences the effectiveness of the optimization in balancing the competing objectives. The adaptive weights method offers a way to overcome this challenge by dynamically adjusting the weights during the optimization process, thus improving the trade-off between the objectives. The algorithm described in [77] operates in several steps. The first step involves Lexicographic optimization, which is a multi-objective method used to identify solutions on the Pareto front by prioritizing one objective function while treating the others as constraints. In single-objective optimization, focusing on a single function might lead to a weakly Pareto optimal solution. However, lexicographic optimization aims to overcome this by incorporating multiple objectives in a hierarchical order, allowing it to find true Pareto optimal solutions. This approach ensures that the solution is robust and aligned with the desired outcomes across multiple objectives.

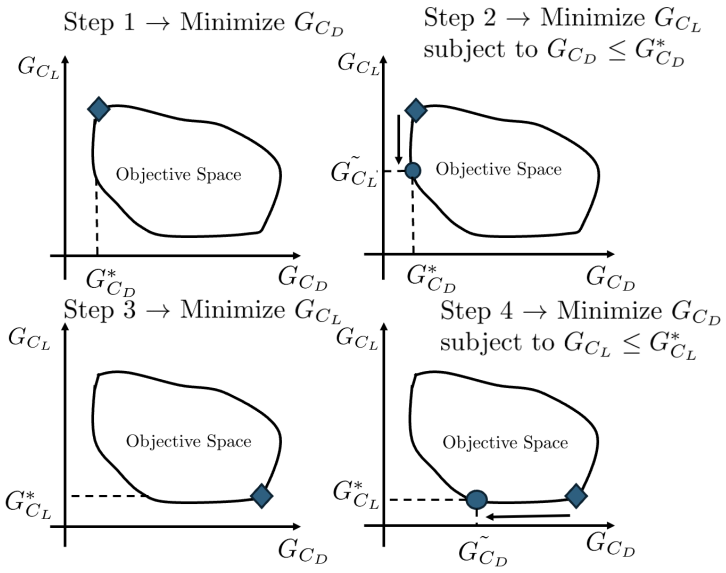


Figure 4.20: Lexicographic optimization

Figure 4.20 illustrates the four steps involved in finding the two best points in the objective space. The process is explained with two objective functions, which corresponds to the case of the NACA0021 optimization. In the first step, one of the two objectives is minimized. Then, in the second step, the second objective is optimized, while imposing the constraint that the first objective must remain less than or equal to the minimum value obtained in the first step. In Figure 4.20 the asterisk ($*$) denotes the optimal value of the objective, while the tilde (~) indicates the minimum achievable value for that objective

when the other objective is constrained. This restriction ensures that the second objective is minimized under the condition that the first objective does not worsen, thereby identifying the first optimal point on the Pareto front. The same two steps, shown as step 3 and step 4 in Figure 4.20, are then performed starting from the second objective. In this test case, the first objective is the error in C_L (G_{C_L}), while the second objective is the error in C_D (G_{C_D}). This method ensures a balanced optimization process that considers both objectives, leading to the identification of optimal solutions on the Pareto front.

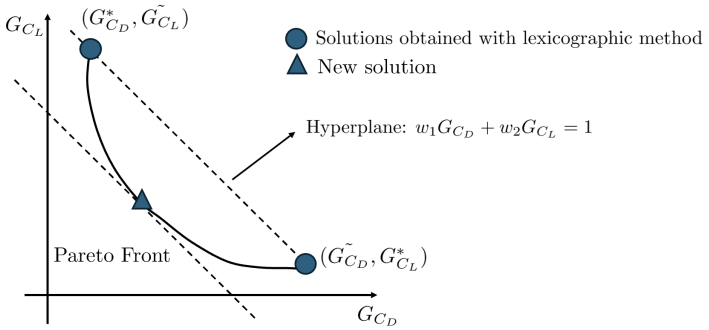


Figure 4.21: Hyperplane in the 2D objective space.

Figure 4.21 depicts the adaptive weights process. A hyperplane is a subspace with one dimension less than its ambient space. In two-dimensional space, it is represented as a line. The hyperplane is constructed in the objective space using previously obtained solutions, and its coefficients are then used as new weights for the different objective functions. The weights for combining the individual objectives are determined by solving the following linear system.

$$\begin{bmatrix} \tilde{G_{C_D}} & G^*_{C_L} \\ G^*_{C_D} & \tilde{G_{C_L}} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.21}$$

By using the weights obtained from Equation 4.21, new Pareto optimal solutions are found along the Pareto front. The hyperplane, which is a line in this case, is constructed using these new weights. It moves within the objective space to minimize the objective function, expressed as a weighted sum of the different objectives.

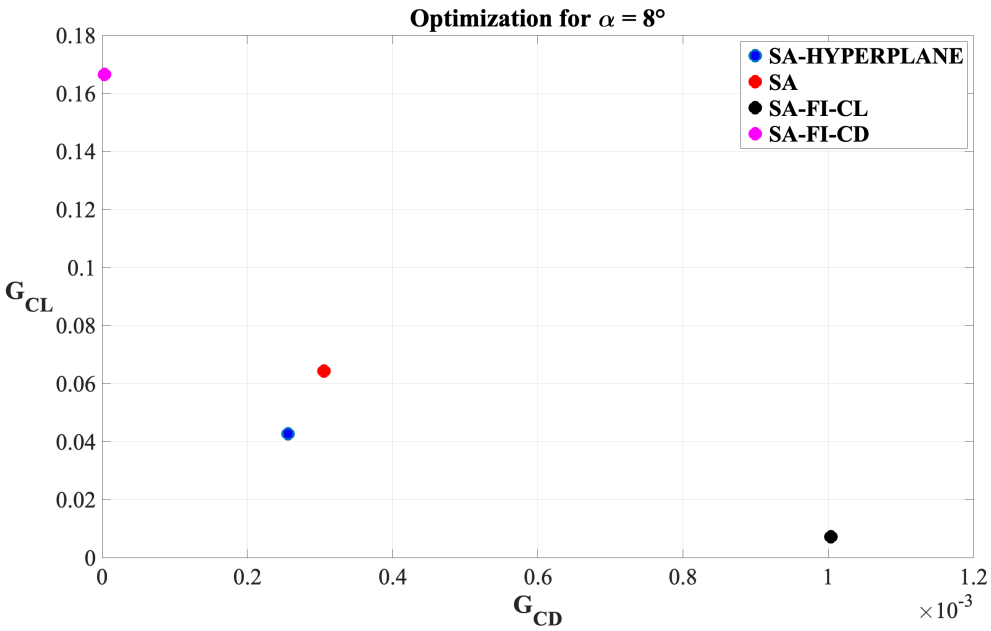


Figure 4.22: Pareto front for $\alpha = 8°$.

Figure 4.22 shows the results obtained at $\alpha = 8°$ using the technique described earlier. The axes G_{CL} and G_{CD} represent the objective functions defined as the errors in C_L and C_D, respectively. As shown, the blue dot corresponds to the results from the adaptive weight method, which successfully minimizes the errors in both C_D and C_L compared to the original SA model (represented with the red dot in the figure). The same results are presented in Figure 4.23 for $\alpha = 10°$.

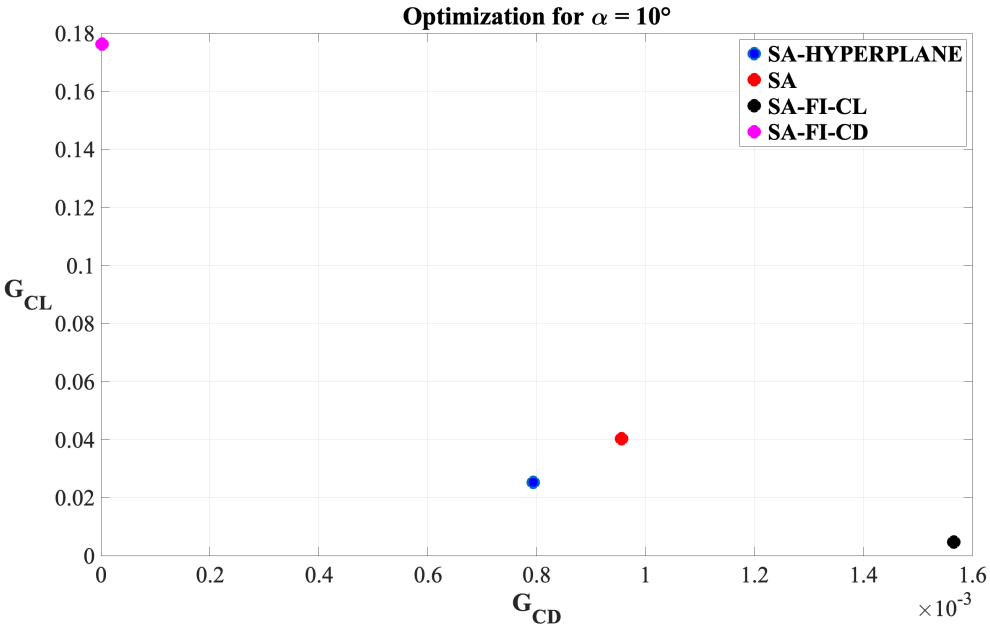


Figure 4.23: Pareto front for $\alpha = 10°$

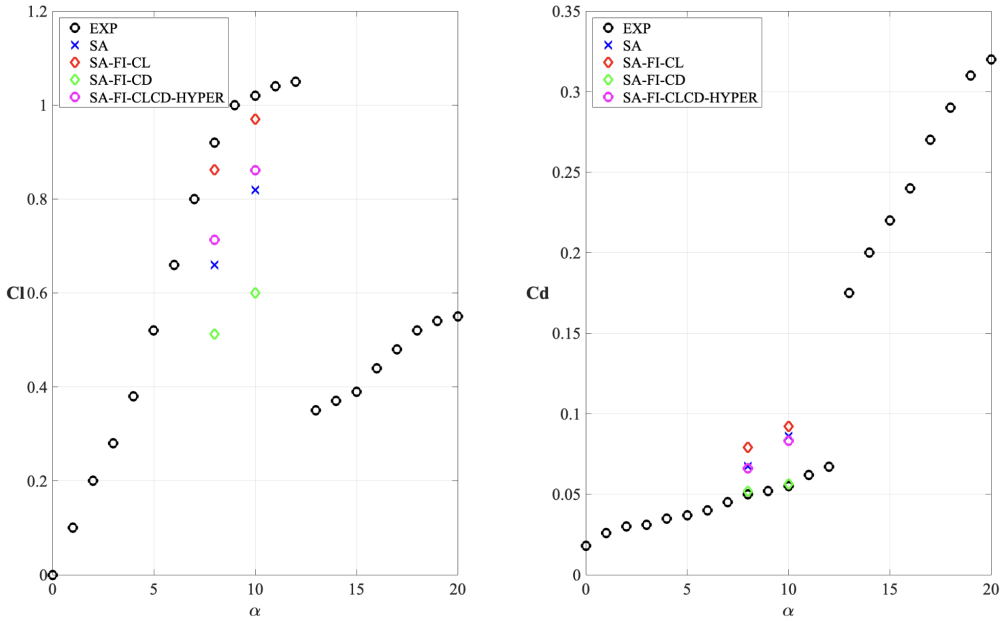


Figure 4.24: Single and Multi-Optimization comparisons.

Finally, in Figure 4.24, the results in terms of C_L and C_D are shown. The differences between the experimental, original SA, single-objective (SA-FI-CL and SA-FI-CD in

the legend) and multi-objective optimizations are highlighted. The multi-objective technique provides a compromise between the two objectives. However, there was limited improvement in the prediction of C_D in the pre-stall region compared to the original SA model. The computational cost of performing lexicographic optimization is nearly twice that of a single-objective optimization. However, in this case, the error between the SA model and the experimental value of C_D is already small. Notably, the results from the single-objective optimization targeting C_L yield better accuracy in terms of C_L error, while the error in C_D remains relatively low. Therefore, for the next step of the offline phase, the training of the data-driven model, the results from the C_L optimization were preferred, even though the predicted C_D values are less accurate than those of the original SA model in the pre-stall region.

Chapter 5

Feed Forward Neural Network

Field inversion optimization, as outlined by Duraisamy et al. [39], traditionally requires coupling with a data-driven model. This pairing facilitates the development of surrogate models to predict correction fields under varying input parameters. However, these models have limited generalizability, as they cannot predict correction fields outside the scope of the data set. For instance, a model trained on high-fidelity simulations of airfoils at different angles of attack cannot extrapolate to entirely different geometries, like cars or blunt bodies. However, machine learning algorithms can generalize to unseen conditions within the range of the data set, such as untrained angles of attack. This thesis explores the use of a Feedforward Neural Network (FNN), a fundamental neural network architecture. Neural networks trace their origins to 1943, when McCulloch and Pitts introduced a computational model of neurons [53]. Subsequent advances included Rosenblatt's 1958 perception [75], the first neural network capable of classifying images using error correction learning. Later innovations, such as Widrow and Hoff's ADALINE and MADALINE models in 1959 [98], expanded neural network applications to real-world problems such as noise filtering in phone lines.

Today, Artificial Neural Networks have gained prominence for their ability to identify patterns, handle large datasets, and solve prediction and classification tasks in various fields. Their nonlinear structure enables them to capture complex relationships, making them a powerful tool in computational research and industry. The exponential growth of Artificial Neural Networks research reflects their growing importance as a versatile solution for modern challenges.

5.1 Neural Network Principles

A neural network is composed of interconnected layers of neurons that work together to learn patterns and make predictions. Its structure typically includes three main components: the input layer, one or more hidden layers, and the output layer.

The input layer receives data and passes them to subsequent layers for processing.

Each neuron in the input layer corresponds to a feature or variable in the dataset. For example, in an image processing task, the input layer might have one neuron for each pixel in the image. Hidden layers, located between the input and output layers, perform the primary computations that enable the network to learn complex patterns. Each neuron in these layers computes a weighted sum of its inputs, expressed as:

$$z = \sum_i (w_i \cdot x_i) + b \tag{5.1}$$

where w_i are the weights, x_i are the inputs and b is the bias term. This result is then passed through an activation function, a critical element that introduces non-linearity. Without activation functions, the network would behave like a linear model and fail to capture complex patterns. Common activation functions include the rectified linear unit (ReLU) function for efficiency and sparsity, the sigmoid and hyperbolic tangent functions for smoother gradients, and the softmax function for multi-class classification tasks. The output layer provides the final predictions. Regression tasks, typically have one neuron outputting a continuous value, while for classification tasks, the number of output neurons corresponds to the number of classes, often using softmax or sigmoid activation functions.

The training process relies on the backpropagation algorithm, which minimizes the error between the model's predictions and actual values by adjusting weights and biases. Backpropagation works in two phases: a forward pass and a backward pass. During the forward pass, data propagates through the network, and predictions are compared to target values using a loss function. The backward pass calculates gradients of the loss with respect to the network's parameters, applying the chain rule of differentiation layer by layer. These gradients are then used to update the weights and biases using optimization algorithms like Stochastic Gradient Descent (SGD) or adaptive moment estimation known as Adam [40]. This iterative process enables the network to learn complex mappings from inputs to outputs, making neural networks highly effective for tasks requiring sophisticated pattern recognition.

5.2 Correction Field Prediction

Despite their simple architecture, neural networks theoretically can approximate any measurable function [35]. However, achieving this is not straightforward. Not all architectures can easily reach such objectives, and even though modern Python libraries like TensorFlow and PyTorch simplify implementation, selecting the right architecture, activation functions, optimization algorithms, and hyperparameters requires significant effort. This process, coupled with the need for clean, well-prepared datasets, often constitutes the most time-consuming aspect of developing neural network models.

In the CFD field, neural networks are commonly used to generate surrogate models for relatively simple problems. For example, they can predict global coefficients like

drag or lift for an airfoil based on its geometry, but usually under laminar flow conditions without turbulence. More complex applications include generating surrogate models for non-linear terms in the Reynolds stress tensor, where the neural network uncovers relationships between predefined inputs and outputs. However, in this work, the focus is on cases where only the output, the correction field β obtained through the field inversion optimization process, is known. This correction field modifies turbulence equations to meet specific objectives and is computed within a CFD framework using finite volume methods. Since the correction field depends on the specific geometry, mesh, and case used in the field inversion process, it cannot directly generalize to other cases. The goal, therefore, is to extend the utility of this correction field by discovering a generalizable relationship using machine learning. For example, if the objective is to improve the model for transition phenomena in airfoils, we aim to predict the correction field for new airfoils with similar characteristics or for the same airfoil at different angles of attack where similar transition behavior is expected. The challenge lies in identifying the appropriate physical input variables, derived from the same CFD computation, to relate to the correction field. Once a model is trained to approximate this relationship, the correction field can be predicted for new cases based on these physical variables. This approach transforms the field inversion optimization procedure into a dataset generation process for training machine learning algorithms. Although this method may appear time-intensive, it is a computationally cheaper alternative to creating datasets through high-fidelity simulations like DNS or LES. High-fidelity simulations, while desirable for their accuracy, are often impractical due to computational constraints. Moreover, directly using DNS results to improve RANS models can be problematic, as the two rely on fundamentally different assumptions. Field inversion, by contrast, generates datasets specifically tailored to RANS models, bridging the gap between data-driven methods and traditional turbulence modeling.

5.3 Input choice

The primary objective of using the FFNN is to establish a relationship between local flow features and the local correction field. Ideally, this relationship could then be used in any CFD code to solve the modified Spalart–Allmaras equation over time. The selection of input variables is critical to the success of this approach, as it directly impacts the accuracy and reliability of the results. General guidelines recommend choosing local, Galilean-invariant variables derived from RANS simulations. However, the final selection is determined by the researcher and must align with the specific goals and requirements of the study. One major challenge with this methodology is that there might not be an exact relationship between the input variables and the correction field. Additionally, identifying the correct set of input features can be difficult. Worse, the FFNN might learn a relationship that lacks physical realism, potentially leading to issues such as numerical instability during transient simulations. Various researchers have selected

different sets of input variables, adapting them to the characteristics of the flow under study and the specific corrections required for the turbulence model. Ferrero et al. [27] employed five input variables to train the feedforward neural network (FNN) in detail: χ, $log(\tau/\tau_{ref} + \epsilon)$, $f'd$, $log(P/(D + \epsilon) + \epsilon)$ and $log(|\nabla(\tilde{\nu})|d/(\tilde{\nu} + \nu) + \epsilon)$.

The first input, χ, represents the turbulence intensity, defined as the ratio between turbulent viscosity and molecular viscosity. The second input is the normalized stress tensor, scaled by a reference stress tensor defined as $\tau_{ref} = \rho(\nu + \tilde{\nu})^2/d^2$, to ensure locality in the input. A logarithmic transformation was applied to mitigate overfitting in the neural network, with an additional small term ϵ added to prevent undefined values, particularly in the freestream region. The third input, $f'd$, is a modified version of the shield function f_d, originally introduced in [87]. It is defined as:

$$f'_d = 1 - \tanh((rd)^{0.5}) \tag{5.2}$$

where rd is a dimensionless quantity given by:

$$rd = \frac{\nu + \tilde{\nu}}{d^2 k^2 \sqrt{\frac{\partial u_i}{\partial x_j}\frac{\partial u_i}{\partial x_j}}}. \tag{5.3}$$

Unlike f_d, the modified function f'_d provides a better characterization of the features near the wall. The fourth input is the ratio of production (P) to destruction (D) scaled logarithmically for consistency with the second input. Finally, the gradient of turbulent viscosity is incorporated through proper normalization. This term is included because it appears in the cross-term of the SA equation and helps identify regions with strong variations in the turbulent viscosity field.

The field inversion framework was fully integrated into the Computational Fluid Dynamics solver used in this research (Immerflow). To validate the framework, the transitional flat plate test case was used as mentioned in 4.5.

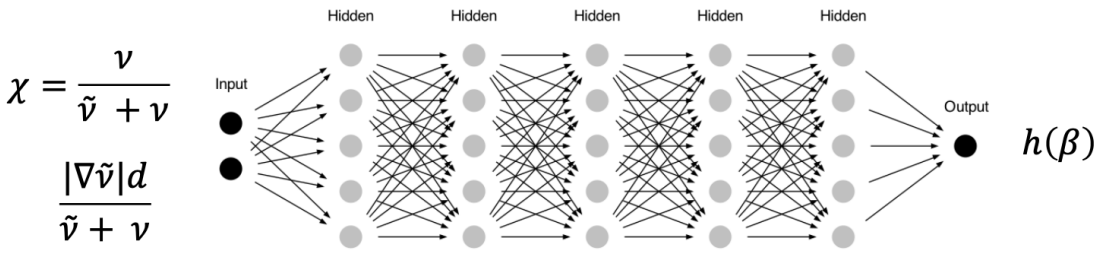


Figure 5.1: Transitional flat plate FFNN Architecture

Figure 5.1 shows the network architecture employed. The choice of architecture including the number of layers and the number of neurons results from a series of trials and errors. The same approach was adopted for the other neural network architectures presented in this thesis. The Adam algorithm was chosen for its adaptive learning rates, computational efficiency, robustness to hyperparameters, and bias correction, all of which contribute to faster convergence and improved stability. The Leaky-ReLU activation function was selected due to its simplicity, computational efficiency, ability to induce sparse activations, and mitigation of the vanishing gradient problem, making it well-suited for deeper networks. For this test case, after a series of trials and errorss, only two inputs were chosen, shown as follows:

- Modified turbulent viscosity ratio:

$$\chi = \nu/(\tilde{\nu} + \nu) \tag{5.4}$$

- Magnitude of the turbulent viscosity gradient:

$$\frac{|\nabla\tilde{\nu}|d}{(\nu + \tilde{\nu})} \tag{5.5}$$

The first input serves as an index of turbulent viscosity intensity, as also utilized in [89]. The second input was already discussed and used by Ferrero et al. [27]. The results obtained with this network will be discussed in the results section 5.5

A deeper analysis of the input choice was performed by using the backward-facing step. This case was particularly relevant due to its well-documented flow characteristics and the ability to optimize different objective functions, specifically the skin friction coefficient (C_f) and the pressure coefficient (C_p). The flexibility in selecting the objective function provided valuable insight into the behavior of turbulence models, particularly in flow separation regions. The outcomes of this study were presented at the ParCFD 2023 conference, where the methodology and improvements achieved through the field inversion approach were discussed. To enhance the accuracy of the turbulence model corrections obtained from the field inversion process, a feedforward neural network was implemented and trained using the Levenberg-Marquardt optimization algorithm, which is well suited for handling nonlinear optimization problems efficiently. The architecture consisted of three hidden layers, structured with 10, 10, and 5 neurons per layer, respectively, as shown in figure 5.2. With regard to overfitting, the training process is systematically monitored through the loss functions computed on both the training and validation datasets. In particular, certain early stopping strategies terminate the training as soon as the validation loss begins to increase, exceeding that of the training set, thereby preventing overfitting. This strategy is applied consistently in all test cases examined.

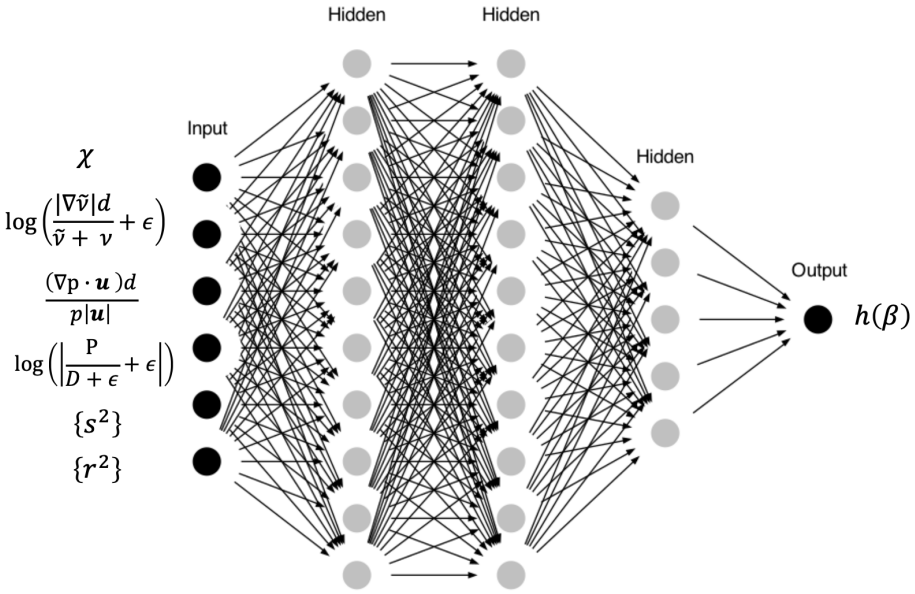


Figure 5.2: Backward-Facing step FFNN Architecture

For the activation functions, the ReLU was employed in the hidden layers because of its advantageous properties, such as preventing the vanishing gradient problem and accelerating convergence during training. The output layer utilized a Tan-Sigmoid activation function, which facilitated a smooth representation of the predicted corrections. Training was conducted using the backpropagation algorithm, with the dataset divided into 80% for training, 10% for testing, and 10% for validation.

The input variables used for this test case are shown below.

- Modified turbulent viscosity ratio:

$$\chi = \nu/(\tilde{\nu}+\nu) \tag{5.6}$$

- Magnitude of the turbulent viscosity gradient:

$$\frac{|\nabla\tilde{\nu}|d}{(\nu+\tilde{\nu})} \tag{5.7}$$

- Streamwise pressure gradient:

$$\frac{(\nabla p\mathbf{u})d}{|\mathbf{u}|p} \tag{5.8}$$

- Production Destruction Ratio:

$$log(P/(D+\epsilon)+\epsilon) \tag{5.9}$$

- First Invariant:

$$\{s^2\} \tag{5.10}$$

- Second Invariant:

$$\{r^2\} \tag{5.11}$$

The selection of input variables was inspired by the work of Ferrero et al. [27], who demonstrated the effectiveness of specific turbulence-related quantities in training neural networks for turbulence model corrections. However, in this study, modifications were made to the input set to improve the robustness and physical interpretability of the model. Notably, f_d' and the stress tensor ratio were omitted, and instead, the first two invariants from Pope's theory ($\{s^2\}$ and $\{r^2\}$) were incorporated [68]. Pope's theory provides a physically consistent framework for describing turbulence using invariant quantities, which ensures that the model respects fundamental turbulence properties. These invariants help capture the non-linear nature of turbulence interactions. A similar strategy was first employed by Ling et al. in [47]. Moreover, another input is introduced the streamwise pressure gradient, which is crucial as it directly affects boundary layer behavior. In regions with an adverse pressure gradient, the flow is more prone to separating from the surface. This separation plays a crucial role in the transition from laminar to turbulent flow. By including the streamwise pressure gradient, the model can better predict and correct for regions susceptible to separation. However, one could argue that incorporating the streamwise pressure gradient introduces a potential violation of Galilean invariance. This issue can be mitigated by replacing the absolute velocity in Equation (5.18) with the velocity relative to the nearest solid wall. In this study, since all bodies are at rest with respect to the chosen frame of reference, the absolute and relative velocities are identical.

A simple analysis was conducted to examine the correlation between the correction field and the input variables. Specifically, a leave-one-out analysis was performed to identify which input variable had the most significant impact on output variation. The Backward-facing step test case involved two objective functions for training, resulting in two distinct correction fields obtained through field inversion optimization. Consequently, the leave-one-out analysis was conducted separately for each dataset. The importance of each input variable was quantified using the R-squared (R^2) parameter. The R^2 coefficient measures the goodness of fit in neural network training by assessing the proportion of variance in the dependent variable that can be explained by the independent variables. It is calculated as follows:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{5.12}$$

where SS_{res} is the sum of squares of residuals, defined as:

$$SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{5.13}$$

and SS_{tot} is the total sum of squares, given by:

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \overline{y_i})^2 \tag{5.14}$$

In these equations, n represents the total number of data points, y_i is the ground truth value, $\hat{y}_i$ is the predicted value, and $\overline{y_i}$ is the mean of the ground truth values. The R^2 values range from 0 to 1, where 1 indicates a perfect prediction, while 0 implies that the model does not explain any of the variance. Higher R^2 values indicate a better fit and more accurate predictions.

Used features	R^2	% ΔR^2
χ, $\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{s^2\}$, $\{r^2\}$	0.9037	
$\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{s^2\}$, $\{r^2\}$	0.5776	-36.09
χ, $\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\{s^2\}$, $\{r^2\}$	0.6033	-33.24
χ, $\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{s^2\}$, $\{r^2\}$	0.7925	-12.31
χ, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{s^2\}$, $\{r^2\}$	0.8008	-11.39
χ, $\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{s^2\}$	0.9297	2.88
χ, $\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{r^2\}$	0.9340	3.53

Figure 5.3: Leave One-Out Cf dataset

Used features	R^2	% ΔR^2
χ, $\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{s^2\}$, $\{r^2\}$	0.5048	
χ, $\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\{s^2\}$, $\{r^2\}$	0.2096	-58.48
$\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{s^2\}$, $\{r^2\}$	0.3022	-40.13
χ, $\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{s^2\}$	0.3759	-25.53
χ, $\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{r^2\}$	0.4000	-20.76
χ, $\log\left(\frac{\lvert\nabla\tilde{\nu}\rvert d}{\tilde{\nu}+\nu}+\epsilon\right)$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{s^2\}$, $\{r^2\}$	0.4547	-9.39
χ, $\frac{(\nabla p\cdot u)d}{p\lvert u\rvert}$, $\log\left(\left\lvert\frac{P}{D+\epsilon}+\epsilon\right\rvert\right)$, $\{s^2\}$, $\{r^2\}$	0.5492	8.80

Figure 5.4: Leave One-Out Cp dataset

For the C_f dataset, the nominal R^2 value when using all input variables was $R^2 = 0.9037$, while for the C_p dataset, it was $R^2 = 0.5048$ as shown in Figure 5.3 and 5.4. The leave-one-out analysis revealed that removing the turbulent intensity ratio caused a 36.09% reduction in R^2 for the C_f dataset. In contrast, for the C_p dataset, the most significant reduction, 58.48% from its nominal value, occurred when the ratio between the production and destruction terms was removed. Excluding the first or second invariants slightly increased the R^2 value in the C_f dataset training, improving the regression fitting. Conversely, in the C_p dataset, these invariants had the opposite effect—removing one of them reduced the model's fitting capability. In this dataset, the streamwise pressure gradient was the least influential variable, while removing the turbulent viscosity gradient led to an increase in R^2.

In the last two test cases, the focus is on low-Reynolds number separation phenomena in airfoils, specifically in the SD7003 and NACA0021 test cases. For both cases, an identical neural network architecture was employed.

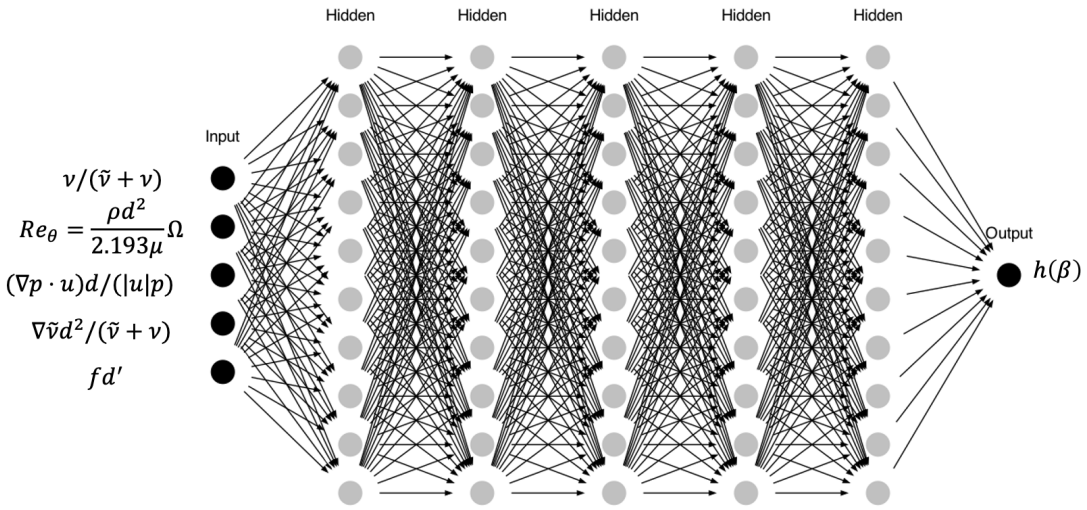


Figure 5.5: Airfoil FFNN Architecture

As depicted in Figure 5.5, the following five input variables were considered for this study:

- Modified turbulent viscosity ratio:

$$\nu/(\tilde{\nu}+\nu) \tag{5.15}$$

- Reynolds thickness number:

$$Re_\theta = Re_\nu/2.193 \tag{5.16}$$

where

$$Re_\nu = \frac{\rho d^2}{\mu} S \tag{5.17}$$

- Streamwise pressure gradient:

$$\frac{(\nabla p\mathbf{u})d}{|\mathbf{u}|p} \tag{5.18}$$

- Magnitude of the turbulent viscosity gradient:

$$\frac{|\nabla\tilde{\nu}|d}{(\nu+\tilde{\nu})} \tag{5.19}$$

- Modified shield function:

$$fd^{'} \tag{5.20}$$

The inputs effective for the BFS test case did not perform well in the present cases. The first and second invariants, as well as the production-destruction ratio, did not provide measurable improvement. Therefore, these inputs were discarded and replaced with a new one. The input in 2.23 was chosen because it appears in a variant of the SA model known as SA-BCM [12]. In this version, the authors introduce an algebraic function to compute an intermittency multiplier for the production term. Since the correction field acts similarly to an intermittency parameter, the Reynolds thickness number was included as an input to the FFNN. The importance of the other inputs has already been discussed. As shown in Figure 5.5, the FFNN consists of five hidden layers, each with 10 nodes. For this Neural Network the Adam backpropagation algorithm and the Leaky-ReLU activation function were employed. The dataset includes simulation results from various angles of attack. For the SD7003 airfoil, data from two angles were used, while for the NACA0021 airfoil, six angles were included. In each case, the dataset was divided into 70% for training and 30% for validation. To enhance training efficiency, a variable learning rate was used, allowing for faster initial learning while ensuring stable convergence. Additionally, early stopping, based on validation loss, was applied to prevent overfitting. Training was stopped once the validation loss ceased to improve, ensuring that the best model parameters were retained.

5.4 Dataset Region Selection

The strength of the field inversion approach lies in its ability to incorporate both global and local coefficients, which can be derived from experimental data or high-fidelity simulations. Global coefficients, such as lift and drag, are generally easier to measure, as they depend solely on fields calculated at the boundary layer. Consequently, the correction field primarily influences the equations within the boundary layer region. This effect is evident from the results presented in the field inversion chapter (Section 4.5), where, in all test cases, the goal functions depend on wall-related variables. As a result, the variation of the correction field is confined to a specific region of the domain. When constructing the training dataset for the neural network, it is more efficient to focus on these specific regions rather than using the entire domain. In areas where the original SA model does not require modification, the correction field remains unchanged. By concentrating on zones with significant correction field variation, the complexity of the problem is reduced, thereby improving the efficiency of the training process. This efficiency gain can be explained intuitively: if the dataset is dominated by regions where the correction field remains unchanged, the FFNN will tend to overfit these areas instead of learning the meaningful variations near the wall. While this might result in an apparent reduction of the loss function, this reduction would primarily reflect the network fitting unchanged regions rather than accurately capturing the necessary variations near the boundary layer, where the correction is most critical. To identify these relevant regions, a modified version of the shield function is employed, defined in Equation 5.2.

Originally introduced in the Detached Eddy Simulation (DES) model [87] to blend LES and RANS models, the shield function ranges from 0 to 1, where a value of 0 indicates regions inside the boundary layer and 1 indicates areas outside it. This enables the generation of a dataset focused on regions where the correction field is most likely to be active. Following this approach, the dataset is selected from areas where the modified shield function, fd', has values less than or equal to 0.9.

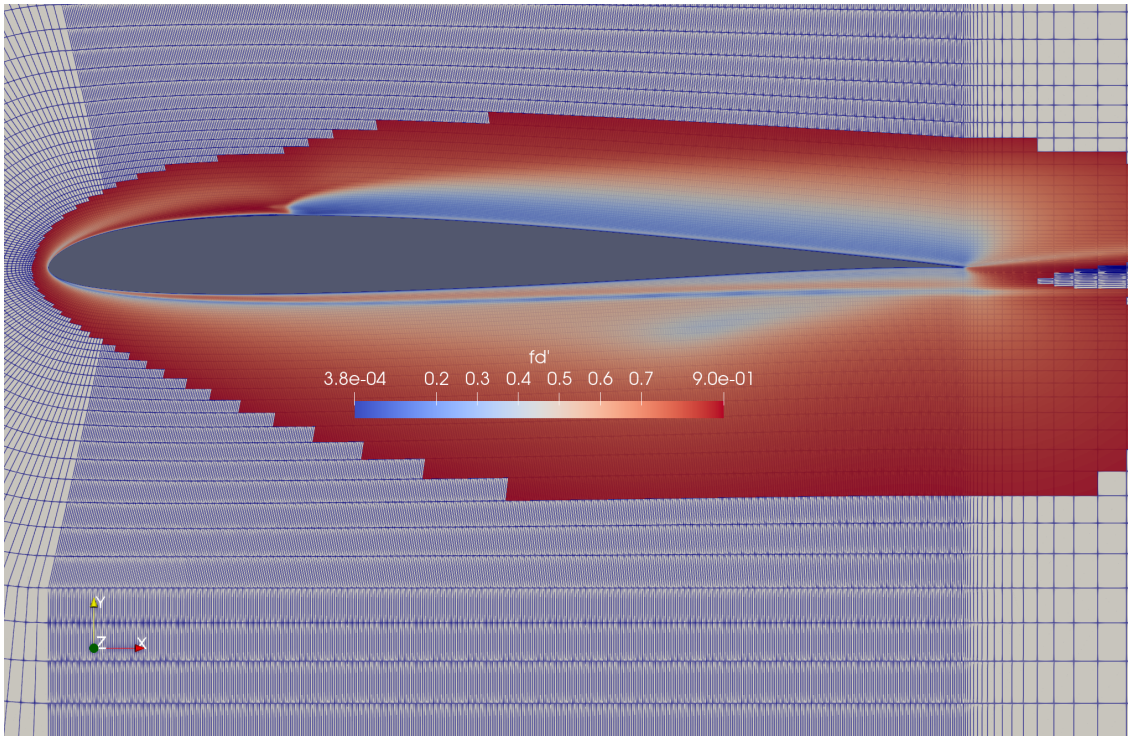


Figure 5.6: Shield function distribution near the SD7003 airfoil wall

Figure 5.6 highlights the regions near the wall of the SD7003 airfoil where the modified shield function remains below 0.9, effectively identifying the portion of the domain where the correction field is most active. This targeted selection ensures that the correction field is accurately represented in the training dataset, enhancing the effectiveness and precision of the neural network.

5.5 FFNN Results

After presenting the three main neural network architectures utilized during this PhD research, along with the studies conducted to assess input parameters and define the dataset extraction domain, the following subsections provide a case-by-case analysis of the results. The evaluation begins with the T3A test case, where the field inversion approach demonstrated excellent agreement with experimental results. Next, the

backward-facing step case is analyzed, comparing different local objective functions, including skin friction and pressure coefficients. Finally, results for low Reynolds number airfoils, specifically the SD7003 and NACA0021, are presented. As discussed in previous sections, for these airfoils, the lift coefficient was selected as the target variable for neural network inference. It is important to note that not all results achieved a satisfactory match with the optimized field inversion outcomes. In some cases, the neural network predictions failed to capture the correct trend and, in certain instances, even performed worse than the original SA model.

5.5.1 Transitional Flat Plate

The neural network illustrated in Figure 5.1 is employed here to infer the correction field previously obtained during the optimization process. The methodology remains consistent across cases. For this particular case, no alternative domain extraction was performed, as the computational domain and mesh were sufficiently small, allowing the entire domain to be used for dataset generation and training. Once trained, the model was integrated with the CFD solver and applied during the simulation.

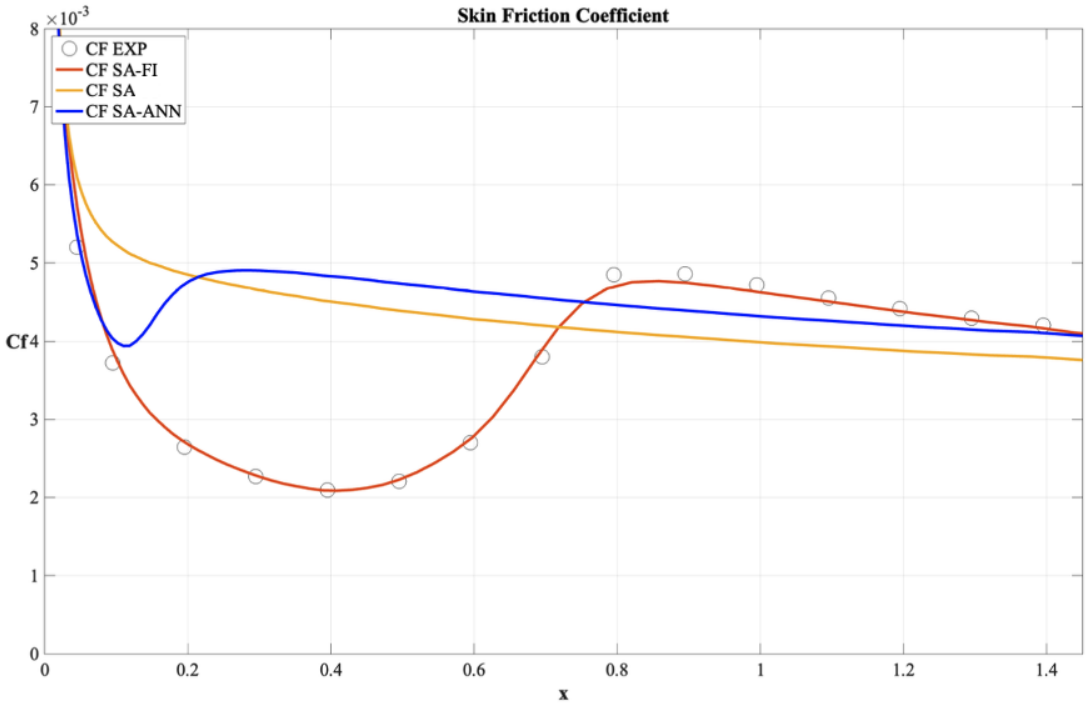


Figure 5.7: Skin friction coefficient comparison results: experimental, original SA model, and augmented SA with Field Inversion and Neural Network.

Figure 5.7 presents the skin friction coefficient results, where the blue line represents the outcome of coupling the CFD solver with the trained artificial neural network.

As observed, the FFNN-augmented model performs worse than the original SA model, predicting an early transition to the turbulent regime. This outcome highlights a key limitation of this approach when integrating the neural network model with the CFD solver. While input selection may have contributed to the poor performance in this case, the results suggest that, during the numerical transient, the neural network predicts correction field values that prematurely enhance the production term in the S–A equation. This, in turn, increases turbulent viscosity too early, reinforcing an overly turbulent regime.

5.5.2 Backward-Facing Step

Results from the Neural Network applied to the backward-facing step test case, incorporating all input sets described in Section 5.3, are presented and discussed. The objective of this test case is to evaluate the method's behavior using two distinct objective functions. Specifically, as mentioned in Section 4.5, both the skin friction coefficient and the pressure coefficient are considered. The full dataset generated through field inversion is utilized for this analysis. The nature of the objective functions is similar, as both produce a correction field that deviates from 1, particularly near the lower wall and especially around the step where flow separation occurs.

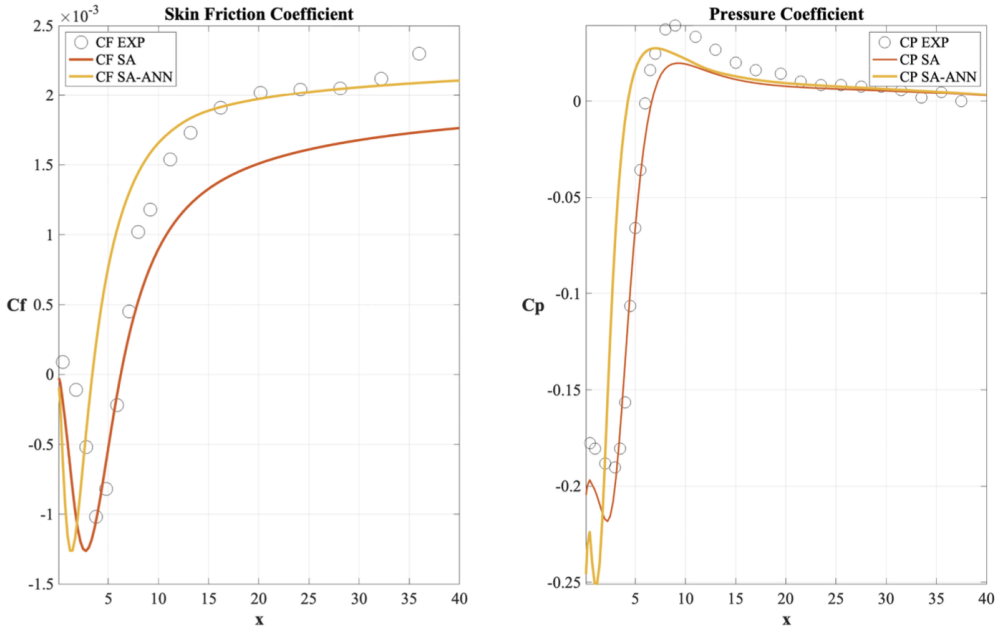


Figure 5.8: Comparison of experimental results, original SA model, and augmented SA model (Cf dataset): skin friction coefficient (left) and pressure coefficient (right).

Figure 5.8 presents the results obtained by training the neural network on the skin friction dataset. Compared to the original SA model, the augmented model demonstrates

improved predictions of the skin friction coefficient, particularly in the region downstream of the step. On the right side of Figure 5.8, the pressure coefficient is shown. It is important to emphasize that these results are derived from the same neural network trained solely on the skin friction dataset; therefore, significant improvements in the pressure coefficient are not expected. As anticipated, the results are largely similar to those of the original SA model, with minor discrepancies observed near the step.

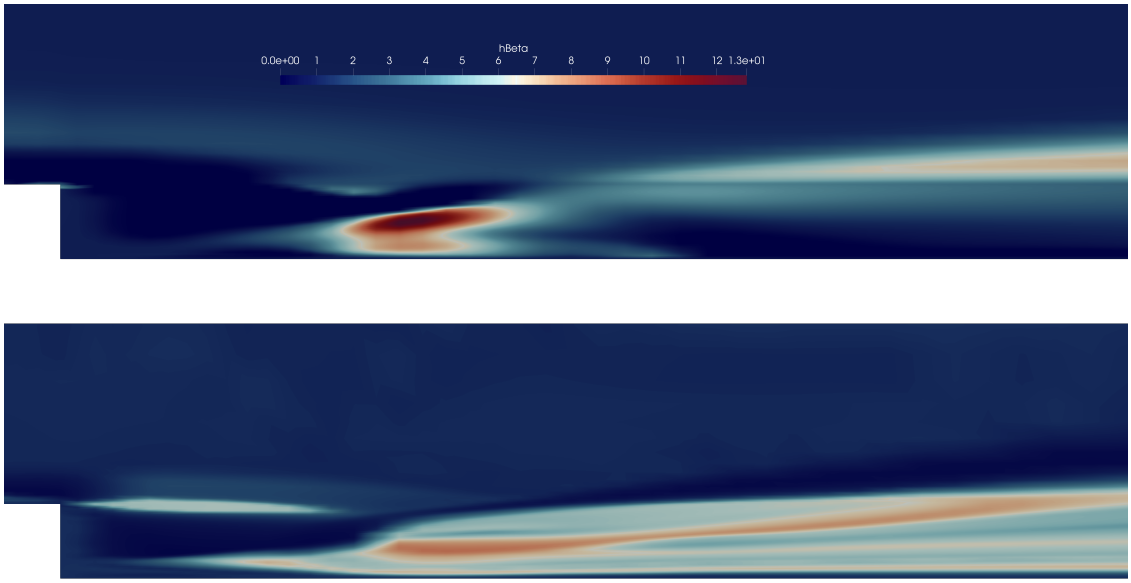


Figure 5.9: Correction field comparison: top, obtained through field inversion; bottom, inferred from the neural network (Cf dataset).

The results shown in section 4.5 shows an almost perfect match between the experimental results and the augmented SA with the correction field in the final optimization iteration. Here the figure 5.9 shows differences in the inferred correction field with the neural network in comparison with the target. That is the reason why the skin friction coefficient does not match the experimental results while improving with respect to the original model.

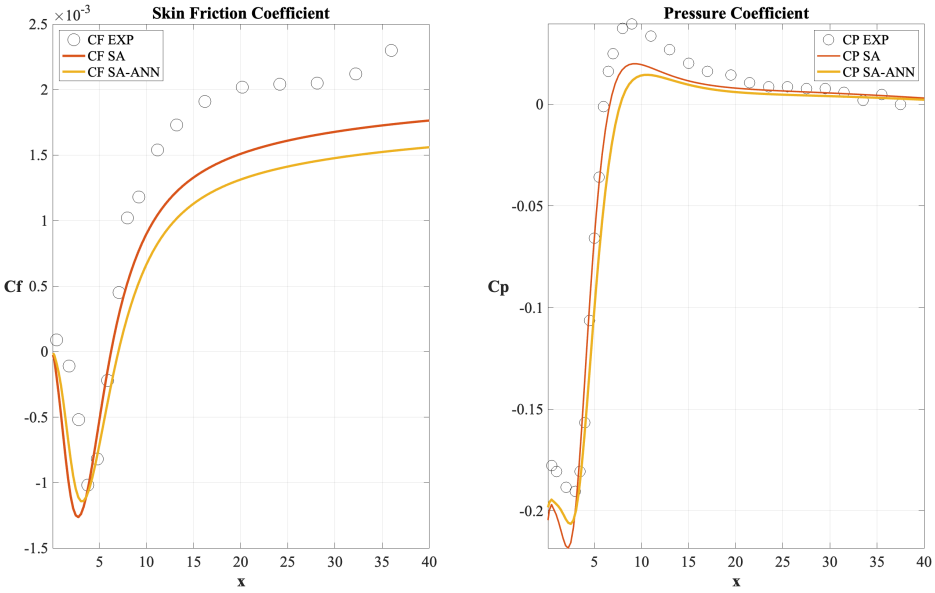


Figure 5.10: Comparison of experimental results, original SA model, and augmented SA model (Cp dataset): skin friction coefficient (left) and pressure coefficient (right).

The neural network trained on the pressure coefficient dataset, using the same set of inputs as the one used for the skin friction coefficient, exhibited a lower R-squared value. Consequently, the improvement in the pressure coefficient is limited compared to the original SA model, as shown in Figure 5.10. However, a slight improvement is observed in the region near the step ($0 < x < 5$) for both the pressure and skin friction coefficients. Nevertheless, since the prediction of the skin friction coefficient is worse than that of the original SA model, it is preferable in this case to use only the neural network trained on the skin friction coefficient dataset.

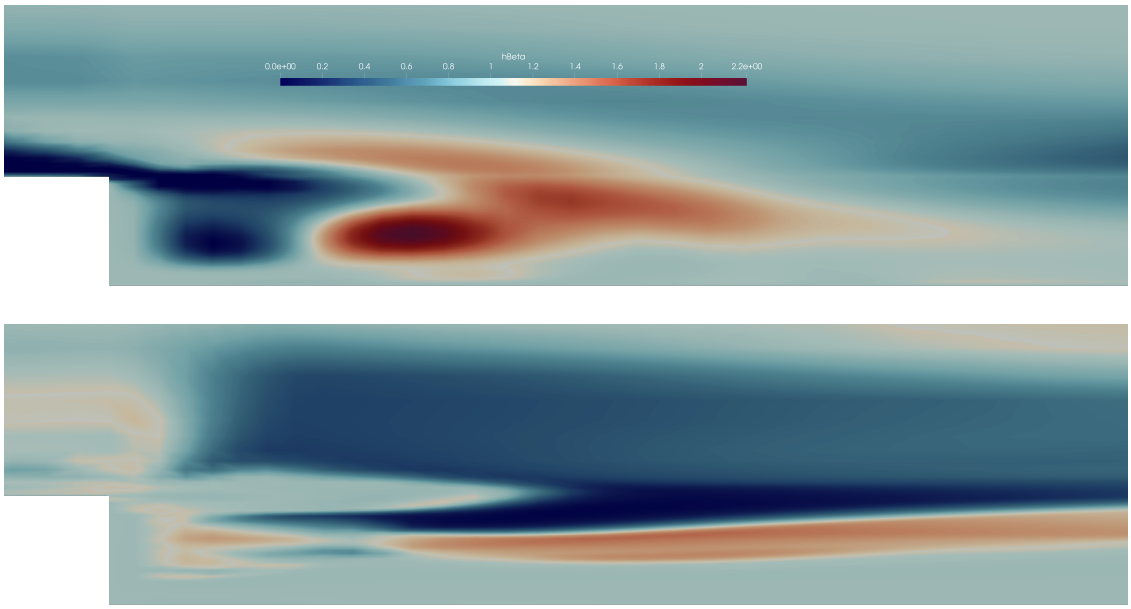


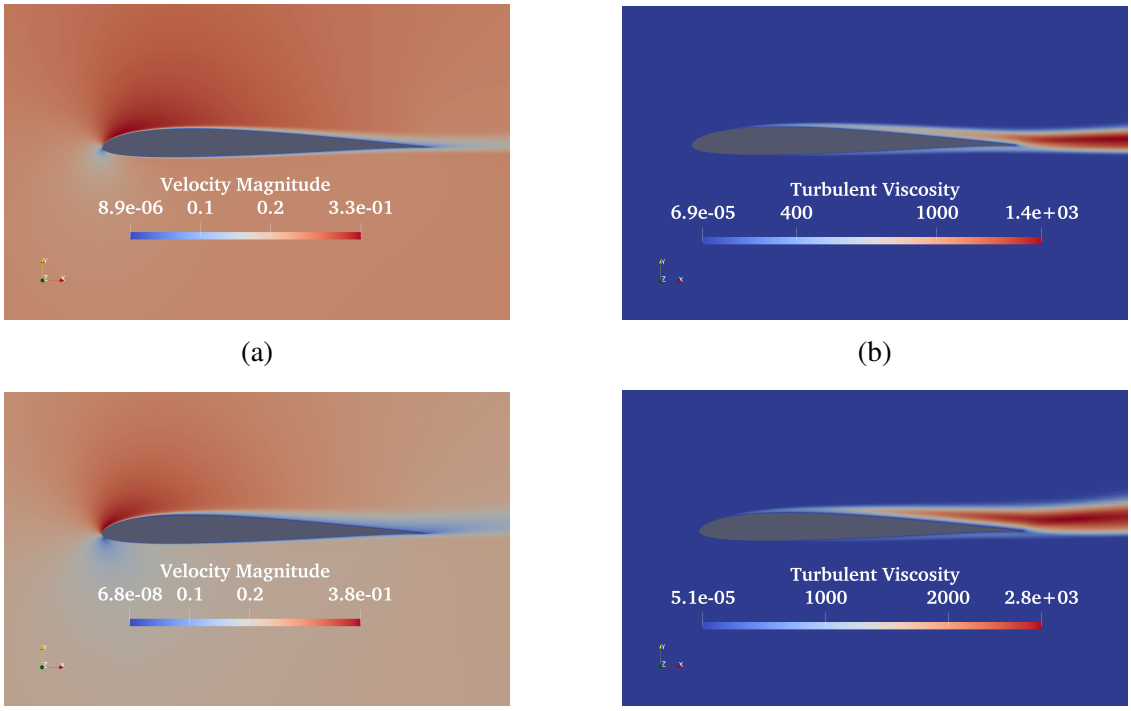
Figure 5.11: Correction field comparison: top, obtained through field inversion; bottom, inferred from the neural network (Cp dataset).

Figure 5.11 compares the correction field obtained through field inversion with the one inferred by the FFNN. It is evident that, in this case, the neural network was unable to capture the correct shape and variation of the target correction field. These results demonstrate the neural network's limitations for this test case. While the model performs relatively well with the skin friction dataset, it struggles to capture the target correction field, especially with the pressure coefficient dataset. Since the input-output pairs do not represent a smooth function, the neural network's performance is inherently limited. Even though the R-squared value is relatively high for the skin friction dataset, it does not ensure good results, as shown by the model's difficulties.

5.5.3 SD7003 Airfoil

The results obtained using the FFNN for the SD7003 test case are presented here. As previously mentioned, ground truth data are available for two angles of attack for this case. In Section 4.5, the effectiveness of using local coefficients over global ones was demonstrated by comparing field inversion results based on different goal functions. However, to maintain consistency with the NACA0021 test case, where only global coefficients are available, the neural network is trained using only global coefficients. Specifically, the results obtained using the lift coefficient error as the objective function are considered. The architecture and input variables for the FFNN were previously detailed in Section 5.3. The loss function is defined as the mean squared error between the simulated correction field and the field inversion results.

An early stopping method is applied to prevent overfitting, which stops the training automatically when necessary. The R-squared obtained during training was $R^2 = 0.862$.



(a) (b)

(c) (d)

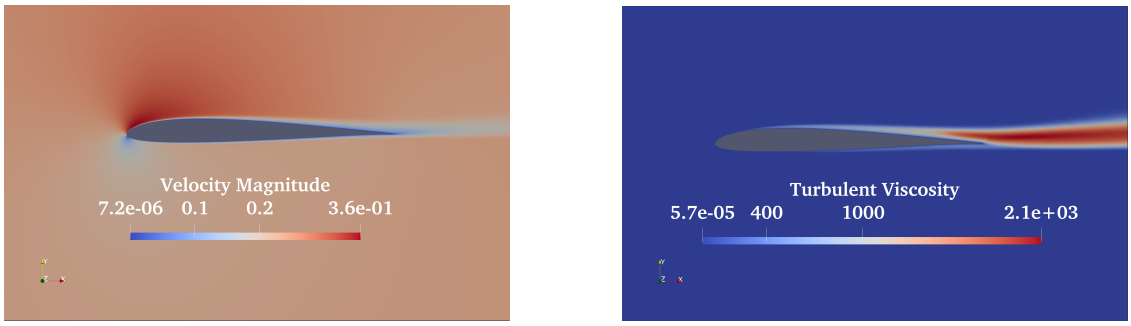
Figure 5.12: Results of the FFNN-augmented Spalart–Allmaras model for the SD7003 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 4°$. (**c**) Velocity magnitude and (**d**) turbulent viscosity at $\alpha = 8°$.

The results from the FFNN-augmented model, implemented within the CFD solver, are presented in Figure 5.12. The simulation begins from the converged solution of the original SA model.

		CL	CD
$\alpha = 4°$	LES	0.59	0.021
	SA	0.551	0.0099
	SA (FI–CL)	0.586	0.0147
	SA (FFNN–CL)	0.550	0.0103
$\alpha = 8°$	LES	0.92	0.043
	SA	0.895	0.0435
	SA (FI–CL)	0.917	0.0436
	SA (FFNN–CL)	0.917	0.0435

Table 5.1: Comparison of lift and drag coefficients for SD7003 airfoil.

To better interpret the results, Table 5.1 compares the lift and drag coefficients. A strong agreement in C_L with the field inversion result is observed for $\alpha = 8°$. However, this is not the case for $\alpha = 4°$, where the trained model fails to predict the correct C_L value, which remains consistent with the original SA model. These outcomes suggest that while the FFNN captures the desired behavior for specific angles of attack, it struggles to maintain accuracy across different conditions. This indicates the need for further tuning or alternative approaches to enhance the model's generalization and predictive performance. An angle of attack at $\alpha = 6°$ was deliberately excluded from the training set and field inversion to assess the model's extrapolation ability for unseen conditions.



(a) (b)

Figure 5.13: Results of the FFNN-augmented Spalart–Allmaras model for the SD7003 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 6°$.

	CL	CD
LES [29]	0.748	0.0319
SA	0.735	0.023
SA (FFNN−CL)	0.746	0.0237

Table 5.2: Comparison of lift and drag coefficients for SD7003 airfoil at $\alpha = 6°$.

Table 5.2 compares the global coefficients obtained from high-fidelity, original SA, and augmented SA models. The results reveal an improvement in C_L prediction but not in C_D, which aligns with expectations, given that the model was trained with data optimized for C_L.

Overall, these findings suggest that the classical FFNN approach lacks sufficient robustness to capture all necessary corrections for the angles of attack included in the training set. Nonetheless, the model performed unexpectedly well for the unseen angle of attack. It is important to note that the original S.A model already exhibited low error in C_L for the SD7003 airfoil.

5.5.4 NACA0021 Airfoil

This section presents the results obtained using the FFNN applied to the NACA0021 test case. The architecture and input variables remain consistent with those used for the SD7003 test case, with the data also optimized for C_L. Unlike the previous case, this dataset includes a wider range of angles of attack for both training and testing, allowing for an evaluation of the model's performance on unseen conditions.

Incorporating a broader range of angles of attack aims to enhance the model's robustness and generalization capabilities, allowing for better capture of complex aerodynamic behaviors across varying flow conditions. In this test case, no mapping function was applied to the correction field. The trained model still achieved a relatively high R^2 value of 0.758. Despite this, the predictive performance fell short of expectations, even at training points.

As discussed in Section 5.3, the model may establish relationships that do not accurately reflect the underlying physical behavior, potentially introducing instability during numerical transients. This issue became apparent in this case, even when starting from a converged solution of the original SA model. While reducing the CFL number can mitigate instabilities, the solver was unable to achieve a new converged solution capable of accurately predicting the corrected turbulent viscosity field.

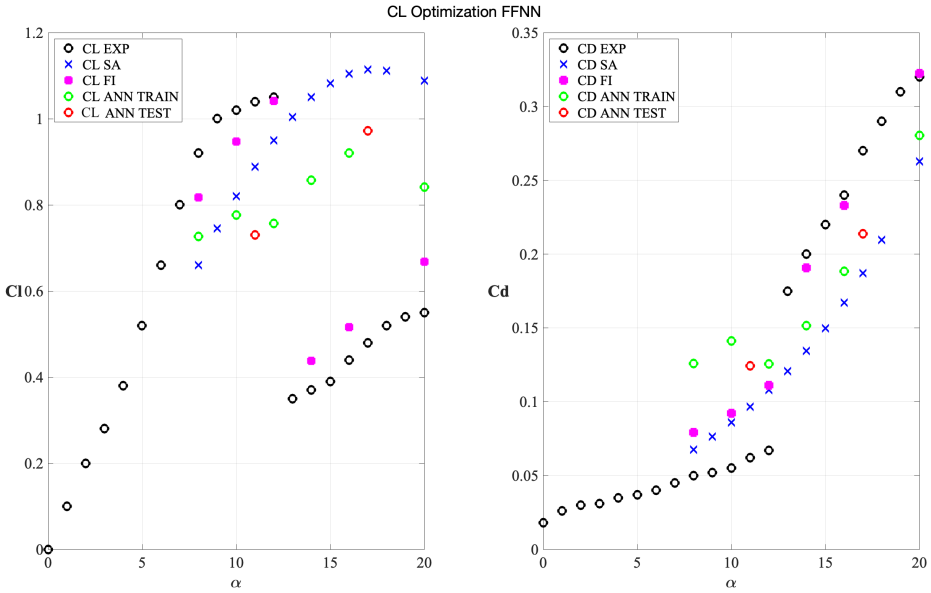


Figure 5.14: Lift and drag coefficients predicted with FFNN-augmented Spalart–Allmaras model for the NACA0021 test case at various angles of attack.

Figure 5.14 shows the predicted lift and drag coefficients at various angles of attack. An improvement in C_L is observed at higher angles of attack, particularly after the incipient stall ($\alpha = 12°$). However, a noticeable discrepancy persists compared to the experimental results. No significant improvement is detected before the stall, and at $\alpha = 12°$, the model performs worse than the original SA model. Further evaluation was conducted on two additional angles of attack, one before the incipient stall ($\alpha = 11°$) and another in the post-stall region ($\alpha = 17°$). These angles, excluded from the training set, exhibited similar performance trends, indicating consistent limitations in the model's predictive accuracy across different flow conditions.

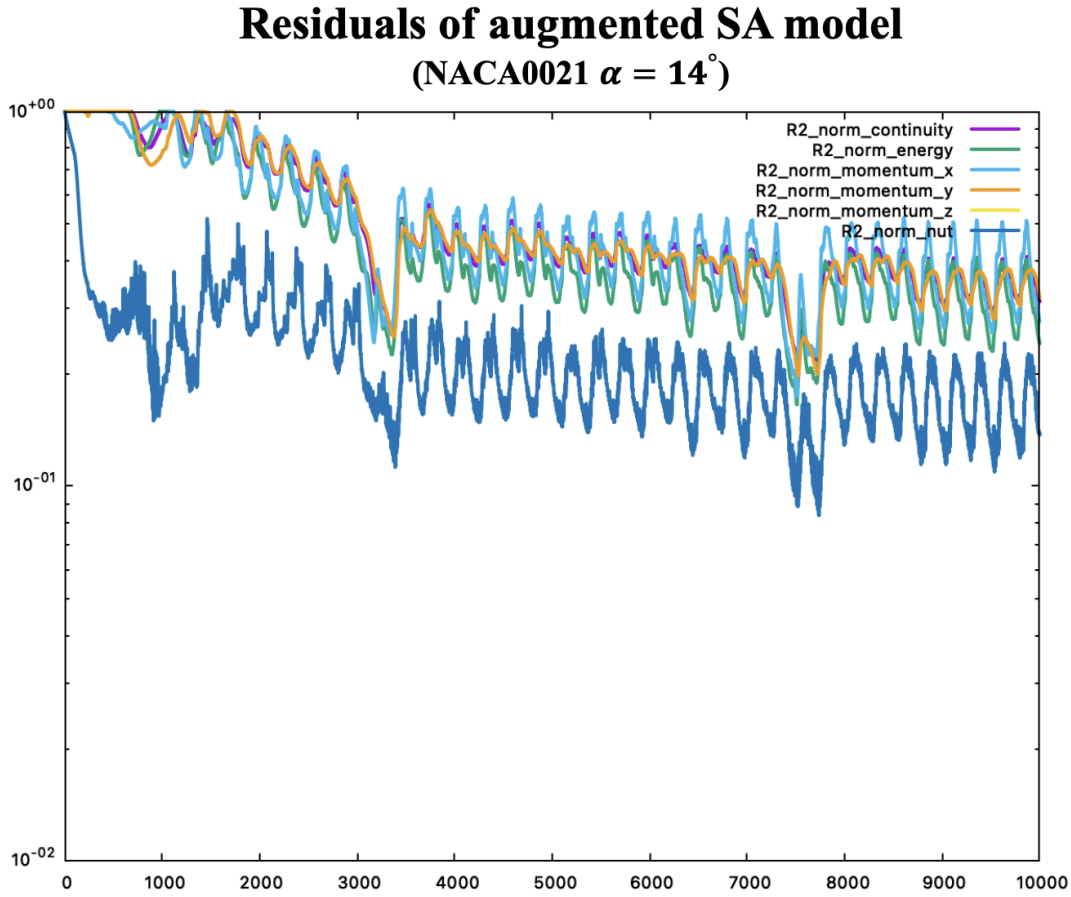


Figure 5.15: Residuals instability.

Figure 5.15 illustrates the residuals for the case with an angle of attack $\alpha = 14°$, providing insight into why the results for this test case were not satisfactory. The primary issue stems from the unstable behavior induced by the coupling between the CFD solver and the neural network model. Establishing a well-defined physical relationship between the correction field and the selected input variables is not always straightforward, which can lead to instability during numerical transients. In some cases, these instabilities persist even when the CFL number is reduced, making convergence difficult. For instance, while the SD7003 test case yielded promising results with this approach, applying the same neural network architecture and input selection to the NACA0021 airfoil did not produce satisfactory outcomes. This highlights the sensitivity of the method to different flow conditions and configurations. To address these challenges, the next section introduces an alternative data-driven model approach featuring a different architecture and application strategy. This revised method was specifically designed considering the

limitations observed in previous results, with a focus on mitigating instability issues arising from the coupling between the neural network and the CFD solver during numerical transients.

Chapter 6

Encoder-Decoder Model for Enhancing Transition Modeling

A completely different strategy was employed for predicting the augmented turbulent viscosity field. Unlike the classical FFNN approach, which presents significant challenges, the new strategy was chosen to address specific complexities. The FFNN method has two main drawbacks: firstly, selecting appropriate input variables is not straightforward, as it can vary from case to case, and there is no guarantee that the selected inputs allow for describing the correction through a function. This makes the model design inherently challenging. Secondly, coupling an FFNN with the CFD code can introduce instabilities during simulations. These instabilities may occur due to inadequate correlation modeling or because the input data during transient states might differ substantially from those seen during training. It is well known that FFNN struggles to generalize effectively outside the range of its training data, especially when training datasets are small, leading to unreliable predictions. Considering these issues, I opted for a different approach that utilizes image-like data. Specifically, I chose the U-Net model. This model, initially developed for biomedical image segmentation, is well-suited for extracting features from complex data representations. Its architecture allows the network to effectively capture both local and global context, making it a robust choice for the prediction of turbulent viscosity fields.

6.1 Convolutional Neural Network

The foundational process of the U-Net model is the convolutional operation, which is performed by a Convolutional Neural Network (CNN). Convolutional Neural Networks were pioneered by Yann LeCun in 1989 [46]. LeCun and his collaborators introduced an early version of CNNs called LeNet-5, which was developed for recognizing handwritten digits, particularly on the MNIST dataset. This innovation laid the groundwork for contemporary CNN architectures and deep learning applications in computer vision. The

architecture of LeNet-5 was revolutionary for its use of convolutional layers to automatically learn features from images, showcasing the power of CNN in pattern recognition. CNN is a type of deep learning model that is particularly well-suited to processing grid-like data structures, such as images. Their architecture is highly effective at identifying patterns in visual data, which makes them ideal for tasks like image classification, object detection, and segmentation.

One major advantage of CNN is its ability to automatically learn spatial hierarchies of features through convolutional layers. This allows CNN to effectively extract meaningful features without relying on manual feature engineering. Despite this, CNN often requires substantial amounts of data and considerable computational power for training, which can pose challenges in some applications. The convolutional process itself involves several key elements which are detailed below:

- **Filter (or Kernel):** Filters are fundamental components of CNN used for feature extraction. A filter is a small matrix (often 3x3 or 5x5) that slides over the input image to detect different features, such as edges, textures, or specific shapes. Each filter has a set of weights that are learned during training, and the convolution operation applies these filters across the image to generate feature maps, which represent different characteristics of the input.

- **Padding:** Padding involves adding extra pixels around the input image, usually set to zero, before applying a convolution operation. Padding helps maintain the spatial dimensions of the image after convolution. Without padding, the dimensions of the resulting feature map will be reduced after each convolutional layer, which can result in a loss of important edge information.

- **Stride:** Stride refers to the number of pixels by which the filter moves across the input image. A stride of 1 means the filter moves one pixel at a time, resulting in a dense feature map. A stride greater than 1 leads to fewer computations and smaller output sizes, but also reduces the level of detail captured. Stride is an important parameter in controlling the resolution and computational complexity of the model.

- **Pooling:** Pooling, or downsampling, is used to reduce the spatial dimensions of feature maps, which helps in decreasing the computational load and controlling overfitting. The most common types are max pooling, which selects the maximum value in a region, and average pooling, which takes the average of values in the region. Pooling layers help the model focus on the most prominent features, and by reducing dimensionality, they make the network more efficient.

These elements collectively form the foundation of convolutional operation. After the convolution operation, the output is passed through an activation function, which, like in a classic neural network, introduces non-linearity, enabling the model to learn complex, non-linear patterns within the data.

In CNN architecture, another important characteristic to take into account is the presence of channels. Channels are essentially different layers of information present in the input data. For example, a typical RGB image has three channels corresponding to the red, green, and blue color components. Each of these channels provides specific information that, when combined, fully represents the image. In a more general context, each channel represents a distinct feature or aspect of the data.

In the context of CFD, channels can represent different physical fields, such as streamwise velocity, pressure, temperature, or turbulent viscosity. This makes CNN highly versatile, as it can be used to analyze complex flow fields by treating each field as a separate channel of input data. Importantly, during convolutional operation, each filter is applied across all of these channels simultaneously, combining information from multiple fields to identify relevant features. This is crucial for capturing the intricate relationships between different physical quantities in CFD simulations.

6.2 U-Net Model

The convolutional operation is commonly employed in a sequential manner, where multiple layers of convolutions are applied in a chain to achieve classification tasks. The final layers consist of fully connected layers, similar to those found in traditional neural networks. The output of this setup is typically a single class label that enables the model to interpret the input image, essentially serving as a recognition mechanism. However, since the primary focus of my work is on a regression problem rather than classification, a different architectural approach is required. This approach still builds upon the convolutional operation but is tailored for continuous output rather than discrete classes. The architecture I am using is known as U-Net. Originally introduced by Olaf Ronneberger et al. in 2015 [74], the U-Net model is particularly well-suited for tasks that involve image segmentation and regression, making it an ideal choice to predict the corrected turbulent viscosity field. The U-Net's encoder-decoder structure allows it to capture both global and local features, effectively addressing the complexities involved in regression tasks.

This model was specifically designed for biomedical image processing, where the goal is to achieve precise localization; each pixel in the output is assigned a corresponding class label. The U-Net architecture features a contracting path that effectively captures contextual information, paired with a symmetric expanding path that facilitates accurate localization. Notably, the authors demonstrated that this network can be trained end-to-end using a limited number of images while outperforming previous state-of-the-art methods. The contracting path comprises a series of convolutional layers followed by 2x2 max pooling operations, which progressively reduce the spatial dimensions of the input image until reaching a bottleneck. Specifically, the convolutional operations utilize 3x3 filters with a stride of one and no padding. The number of channels is doubled before each max pooling operation, enhancing the model's capacity to learn complex

features. Following the bottleneck, the expanding path mirrors the contracting path. It includes additional convolutional layers, and at each layer, the spatial dimensions of the image are increased through upsampling operations. To ensure precise localization, high-resolution features from the contracting path are combined with the upsampled output, allowing subsequent convolutional layers to refine the final output based on this enriched information.

Skip connections are a crucial aspect of the U-Net model, playing a vital role in preserving spatial details throughout the network. The skip connection operation is referred to as "copy and crop" in the original paper. Cropping is necessary because the shape of the image is reduced during each convolutional operation, leading to a mismatch between the dimensions of the decoder output and those of the contracting path. These connections facilitate the direct transfer of features from the encoder to the decoder, helping to alleviate the vanishing gradient problem often encountered in deep networks. By providing a shortcut for gradient flow, skip connections improve training efficiency while retaining critical information during the backpropagation process. They effectively combine low-level spatial information from the encoder with high-level contextual information from the decoder, enabling accurate segmentation even with a limited number of labeled examples. An important modification in the U-Net architecture is the addition of a significant number of feature channels in the upsampling phase. This approach allows the network to convey contextual information to higher-resolution layers effectively. Consequently, the expansive path remains largely symmetric to the contracting path, resulting in a U-shaped architecture. Notably, the network design does not include any fully connected layers and solely relies on the valid regions of each convolution.

6.3 U-Net Applications in CFD and a Novel Post-Processing Method

The U-Net model's notable advantages, its effectiveness with small datasets, have led to its widespread adoption in the CFD field. This model is utilized to accelerate computations within CFD, showcasing significant potential across various applications. For instance, the authors in [106] leveraged the U-Net model for rapid and accurate prediction and optimization of wind environments around buildings. Additionally, a modified U-Net architecture has been effectively employed to predict flow field information in porous media, demonstrating the capability of deep learning to tackle complex fluid dynamics challenges efficiently [107]. Beyond flow field predictions, U-Net architectures have also been utilized in thermal-hydraulic analyses for reactor simulations. One notable example is the Unet-IMR, an intelligent mesh refinement method that enhances mesh quality by predicting and refining poorly defined mesh regions, ultimately improving simulation accuracy in complex fluid dynamics scenarios [57]. High-fidelity CFD simulations are

crucial for the analysis and optimization of components such as heat exchangers in thermal management systems, especially in electric aircraft propulsion. However, these simulations are often computationally intensive, particularly during sensitivity analyses and geometry optimizations. To address these challenges, a deep-learning-based surrogate model, specifically a geometry-adaptive CNN, was developed to predict RANS mean flow fields in heat exchangers with varying geometries. By incorporating a physics-informed approach, this model achieved high accuracy in predicting velocity, pressure, and temperature distributions while complying with governing equations and boundary conditions [8]. In another study, Wang et al. [96] employed a U-Net model as a surrogate, referring to it as a physics-informed CNN, due to enhancements made to the loss function with physical constraints. They applied this model to predict the turbulent mixing layer over time at different Reynolds numbers. Thuerey et al. [91], used a U-Net-based model to predict the velocity and pressure fields around airfoils. Their model took as input the airfoil mask and the initial velocity field, and it predicted the steady-state velocity and pressure fields for various airfoil shapes and flow conditions.

Most applications of the U-Net model in CFD serve as surrogate models, effectively replacing traditional CFD solvers to reduce computational costs while maintaining high accuracy. In this work, however, the application diverges from this typical use. Here, the U-Net model is employed as a post-processing tool. Unlike the approach taken with the FFNN model, the U-Net is used to predict the augmented turbulent viscosity field, starting from the field generated by the original SA model. A RANS simulation with the original SA model is performed. Once these initial results are obtained, the steady solution is provided to the U-net to compute the corrected turbulent viscosity field. This corrected field is imposed as a frozen field in a second simulation in which only the other conservative variables are integrated until a new steady state is reached. By accurately predicting the optimal turbulent viscosity, the solver will quickly converge to the correct steady-state solution with minimal iterations. In this way, the workflow combines the speed of the traditional SA model with the improved accuracy provided by the U-Net, ultimately leading to more reliable simulation results. Like all models, this one has its limitations. Because convolutional layers require structured grids, but CFD data often comes from unstructured meshes, a preprocessing of the data is necessary. This is done by extracting a region around the airfoil and interpolating it onto a lower-resolution structured grid. This step ensures compatibility with the U-Net model and helps manage computational costs. However, interpolation introduces errors that may reduce the accuracy of the model, as the transformation can distort the original data.

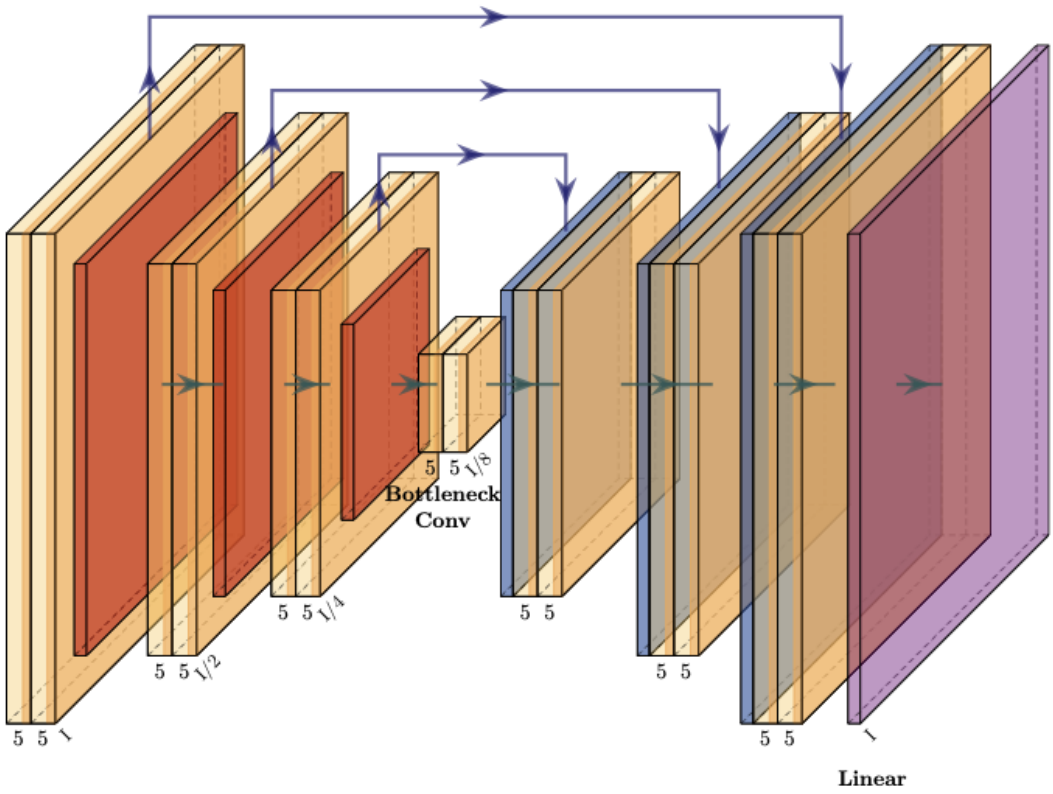


Figure 6.1: U-Net model architecture.

Figure 6.1 illustrates a schematic of the U-Net model developed for this application. Specifically, the architecture is composed of an encoder and a decoder, each consisting of three layers. Every layer includes two convolutional operations, utilizing four filters with dimensions of 3x3, a stride of one, and "same" padding. This combination of parameters was chosen to ensure that the original scene dimensions are preserved throughout the convolutional operations, in line with the formula presented in equation 6.1.

$$Nr = \frac{(N - N_f + 2P)}{S} + 1 \tag{6.1}$$

In equation 6.1, Nr represents the shape of the output image after the convolution, N is the original shape of the input image, N_f represents the size of the filter matrix, S is the stride length, and P indicates the padding applied. In this specific case, "same" padding means $P = (N_f - 1)/2$, ensuring that the dimensions of the output match those of the input.

For this application, the number of features remains constant throughout the convolutional operations. The approach taken here is to apply each filter to every channel of

the input data, with the resulting outputs being combined to compute the final value before applying the activation function. In this model, the Exponential Linear Unit (ELU) activation function is used. The ELU function helps mitigate issues with vanishing gradients and provides smoother convergence compared to the standard ReLU function by allowing small negative values.

Regarding the number of channels involved, the input layer consists of five channels, while the output layer has a single channel. The output represents the augmented turbulent viscosity, which is the primary prediction target. The input channels include the flow fields from the SA model, specifically: streamwise velocity, spanwise velocity, pressure, temperature, and the turbulent viscosity fields. By applying the same number of filters in each convolution, the goal is to effectively blend the information across these fields to accurately predict the augmented turbulent viscosity.

For the training process, the Mean Squared Error is employed as the loss function, coupled with the Adam optimizer, consistent with what was used in the FFNN model. This combination ensures effective optimization while minimizing the prediction error of the turbulent viscosity field.

6.4 U-Net Approach Results

After explaining the methodology behind the U-Net model and its application in this work, the following subsections present the final results obtained using this approach. This methodology was extensively studied during the final year of the PhD, and as such, it was applied to the last two test cases, the SD7003 and the NACA0021 airfoils. The application of the U-Net model to these cases reflects the culmination of the research conducted, demonstrating the model's effectiveness in improving predictions for complex aerodynamic flows.

6.4.1 U-Net Results on the SD7003 Airfoil

This section presents the results for the SD7003 test case using an alternative approach, as outlined in Section 6.3. The U-Net model is trained using data from the two angles of attack for which field inversion was performed. The inputs to the U-Net model consist of the converged results from the original SA model, including velocity, pressure, temperature, and turbulent viscosity fields. The model then predicts a turbulent viscosity field, which is subsequently compared with the corrected field obtained through field inversion optimization. Once trained, the corrected turbulent viscosity field predicted from the U-Net is imposed in the CFD solver. The idea is to turn off the evolution of the SA turbulent equation and leave the mass, momentum and energy equations free to evolve. In this way, the remaining flow variables are allowed to evolve until a new steady-state solution is reached.

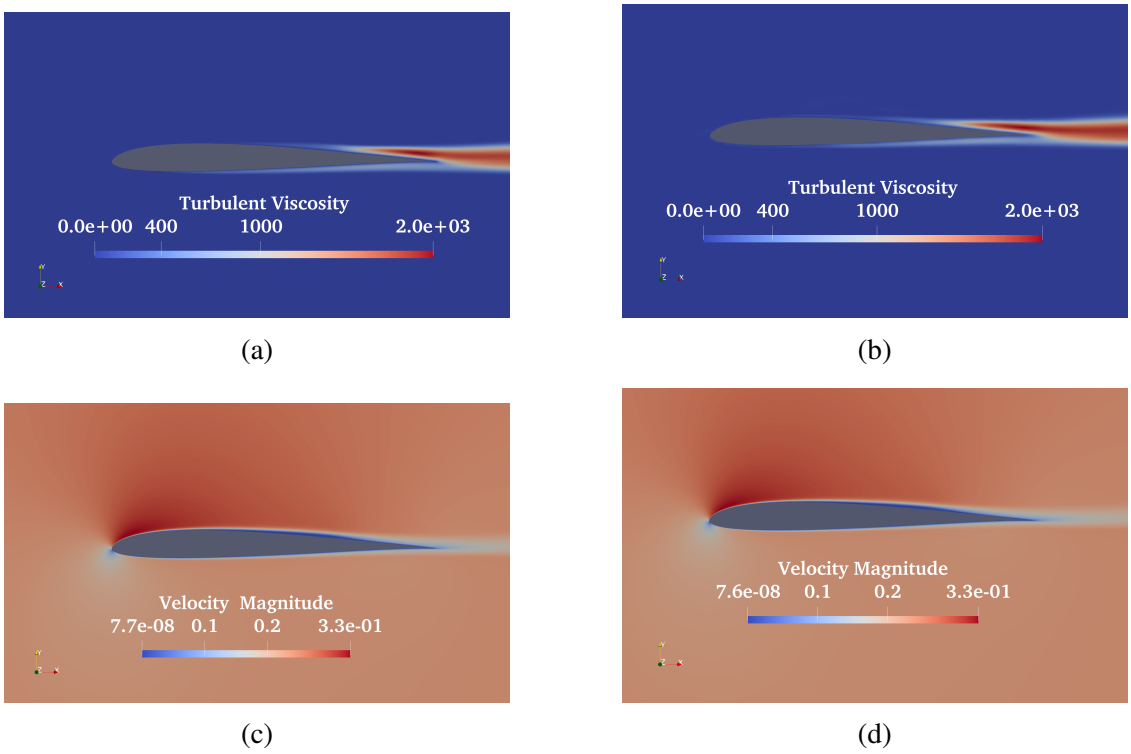


Figure 6.2: Results of the U-Net-augmented Spalart–Allmaras model for the SD7003 test case at $\alpha = 4°$. (**a**) Turbulent viscosity FI and (**b**) Turbulent viscosity U-Net. (**c**) Velocity magnitude FI and (**d**) Velocity magnitude U-Net.

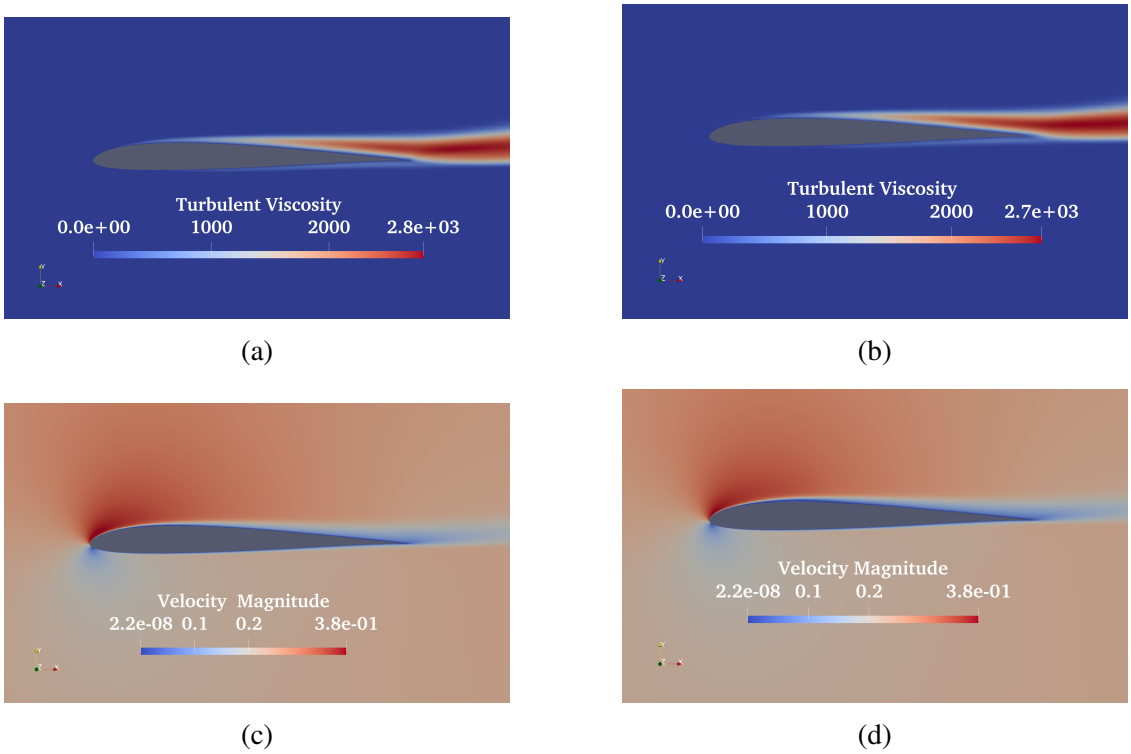


Figure 6.3: Results of the U-Net-augmented Spalart–Allmaras model for the SD7003 test case at $\alpha = 8°$. (**a**) Turbulent viscosity FI and (**b**) Turbulent viscosity U-Net. (**c**) Velocity magnitude FI and (**d**) Velocity magnitude U-Net.

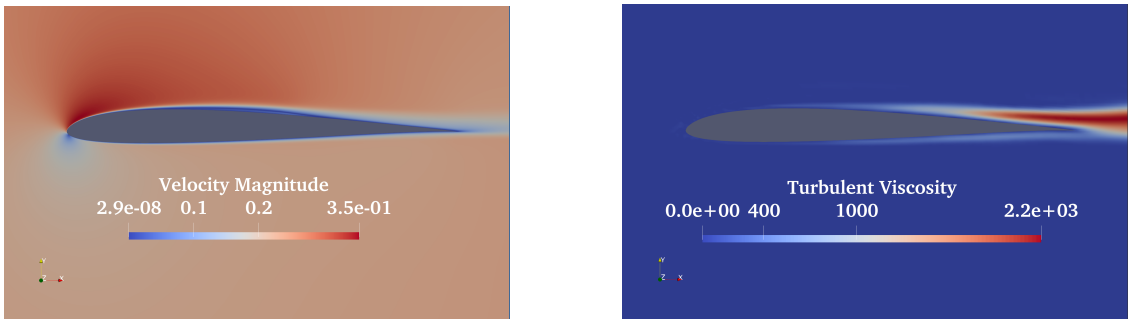
Figures 6.2 and 6.3 illustrate the outcomes of the proposed approach. In both figures, panels (a) and (b) depict the corrected turbulent viscosity obtained through field inversion and the corresponding prediction from the U-Net model. Conversely, panels (c) and (d) present the velocity magnitude derived from field inversion and the velocity magnitude resulting from the RANS equations after imposing the U-Net–corrected turbulent viscosity. The turbulent viscosity field predicted by the U-Net shows strong agreement with that obtained via field inversion, and the velocity magnitude fields also exhibit a close match. However, to rigorously assess the quality of the correction, it is necessary to compare the lift coefficient, which is the ultimate target of the field inversion. In this way, one can determine whether the turbulent viscosity predicted by the U-Net, and the corresponding velocity field, are sufficiently accurate to reproduce the correct lift coefficient.

		CL	CD
$\alpha = 4°$	LES [29]	0.59	0.021
	SA	0.551	0.0099
	SA (FI–CL)	0.586	0.0147
	SA (U-Net–CL)	0.584	0.0147
$\alpha = 8°$	LES [29]	0.92	0.043
	SA	0.895	0.0435
	SA (FI–CL)	0.917	0.0436
	SA (U-Net–CL)	0.917	0.0494

Table 6.1: Comparison of lift and drag coefficients for SD7003 airfoil.

Table 6.1 presents a detailed comparison of the results in terms of C_L and C_D. Since the optimization process was designed to match C_L, the values obtained in this simulation closely align with those from the optimization. For $\alpha = 4°$, the C_D value remains consistent with the optimized result but still deviates from the experimental data. In contrast, for $\alpha = 8°$, the C_D value is higher than the experimental measurement. This discrepancy is expected, as the turbulent viscosity field was specifically optimized to match C_L. Nonetheless, in the case of $\alpha = 4°$, the C_D value demonstrates an improvement over the original model.

The $\alpha = 6°$ case is employed to assess the capability of the U-Net model when applied to conditions outside its training dataset. This evaluation aims to determine how well the model can predict flow behavior at an unseen angle of attack and whether it can accurately capture key aerodynamic features. By analyzing the model's performance in this case, valuable insights can be gained into its robustness, potential limitations, and the extent to which it can be reliably used for flow predictions beyond the training range.



(a) (b)

Figure 6.4: Results of the U-Net-augmented Spalart–Allmaras model for the SD7003 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 6°$.

Figure 6.4 presents the velocity and turbulent viscosity fields for this case. The velocity field reveals a shift in the separation point toward the leading edge compared to the $\alpha = 4°$ case, indicating stronger flow detachment at this higher angle of attack.

	CL	**CD**
LES [29]	0.748	0.0319
SA	0.735	0.023
SA (U-Net−CL)	0.759	0.033

Table 6.2: Comparison of lift and drag coefficients for SD7003 airfoil at $\alpha = 6°$.

A more detailed quantitative comparison is provided in Table 6.2. The results demonstrate that the U-Net model exhibits a discrepancy in the predicted C_L, similar to the original SA model. However, in this case, C_L is overestimated, suggesting that the model may not fully capture the complex transitional effects at this intermediate angle of attack. On the other hand, the predicted C_D closely aligns with the values obtained from LES, indicating that the model reasonably captures the overall flow resistance despite the lift overestimation. These findings highlight both the strengths and limitations of the U-Net approach when applied to unseen data, suggesting that while the model can provide reasonable drag predictions, additional refinement may be necessary to improve lift accuracy across varying flow conditions.

6.4.2 U-Net Results on the NACA0012 Airfoil

This section presents the results obtained using the U-Net model on the NACA0021 test case. As previously discussed, this model was chosen to mitigate the instability issues associated with the classical FFNN approach. In the initial methodology, an attempt was made to establish a correlation between the correction field and specific local flow variables. However, the existence of such a direct correlation is not always guaranteed, which can lead to instability during numerical transients. These instabilities may persist even when the simulation is initialized from the converged solution of the original SA model, as illustrated in Figure 5.15. By employing the U-Net model, the goal is to enhance the robustness of the correction field prediction by leveraging spatial information more effectively. Unlike the FFNN, which rely solely on pointwise data, the U-Net model processes entire field distributions, potentially leading to more stable and physically consistent corrections. The following results assess the effectiveness of this approach and its ability to mitigate the issues that were present with earlier models.

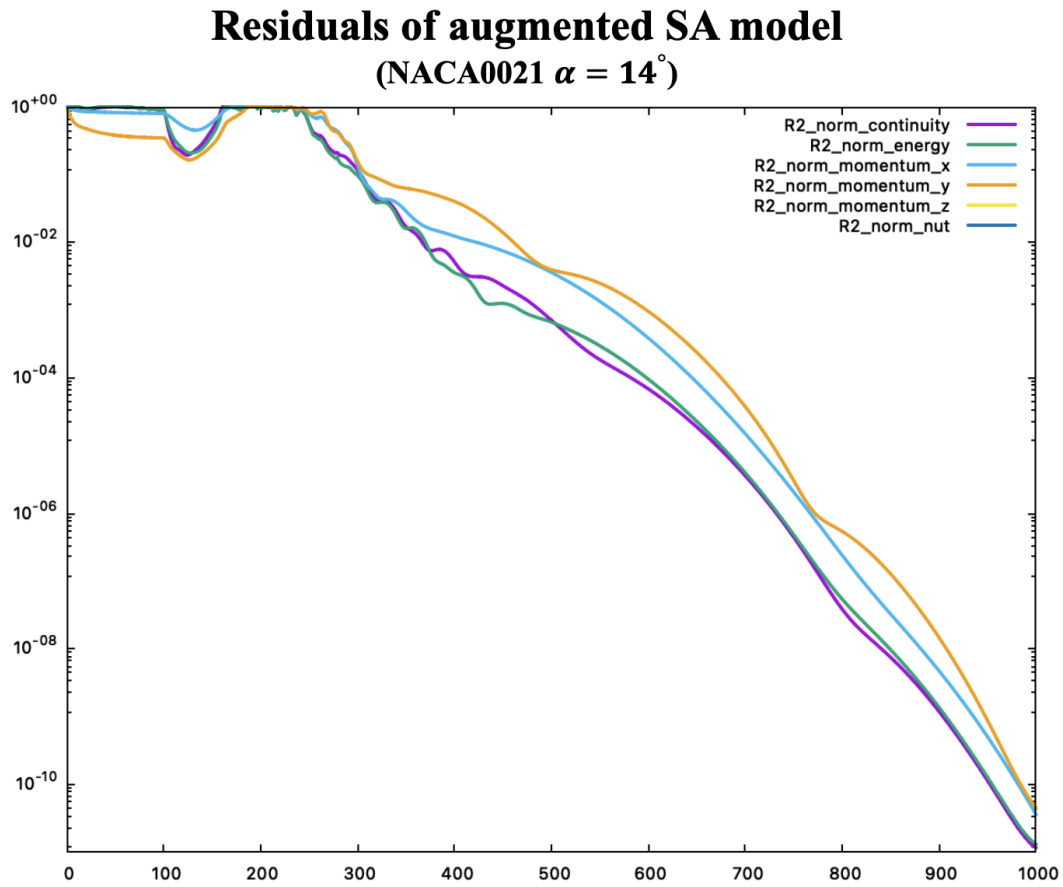


Figure 6.5: Residuals with the U-Net approach.

Figure 6.5 demonstrates the performance improvement achieved with the U-Net model compared to the traditional FFNN approach. The residuals shown in the figure correspond to the same test case ($\alpha = 14°$), allowing for a direct comparison between the two methodologies. The results do not highlight instability behavior, which was a major issue in the FFNN-based approach due to its sensitivity to local flow variables. Unlike the FFNN model, which struggled with numerical transients and often led to divergence, the U-Net model approach exhibits a more stable behavior throughout the simulation. This improvement can be attributed to the ability of the U-Net to capture spatial dependencies throughout the flow field rather than relying on point-wise correlations, which may not always be well-defined.

The results obtained for $\alpha = 8°$ and $\alpha = 16°$ are shown below.

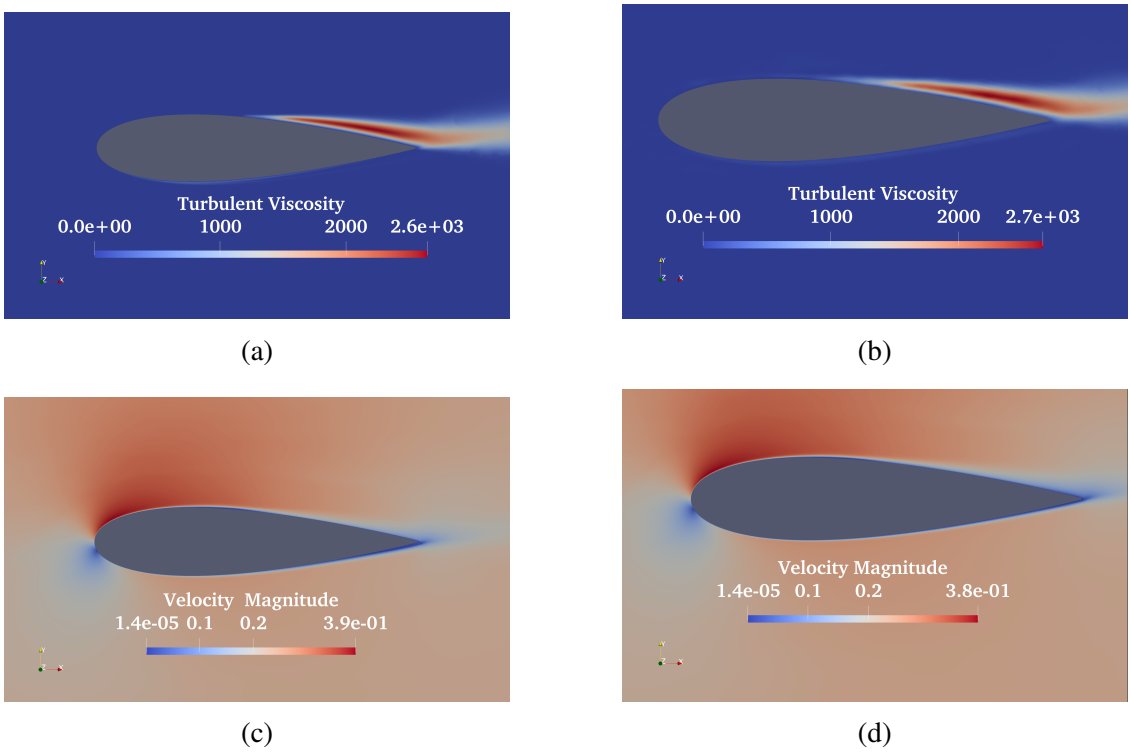
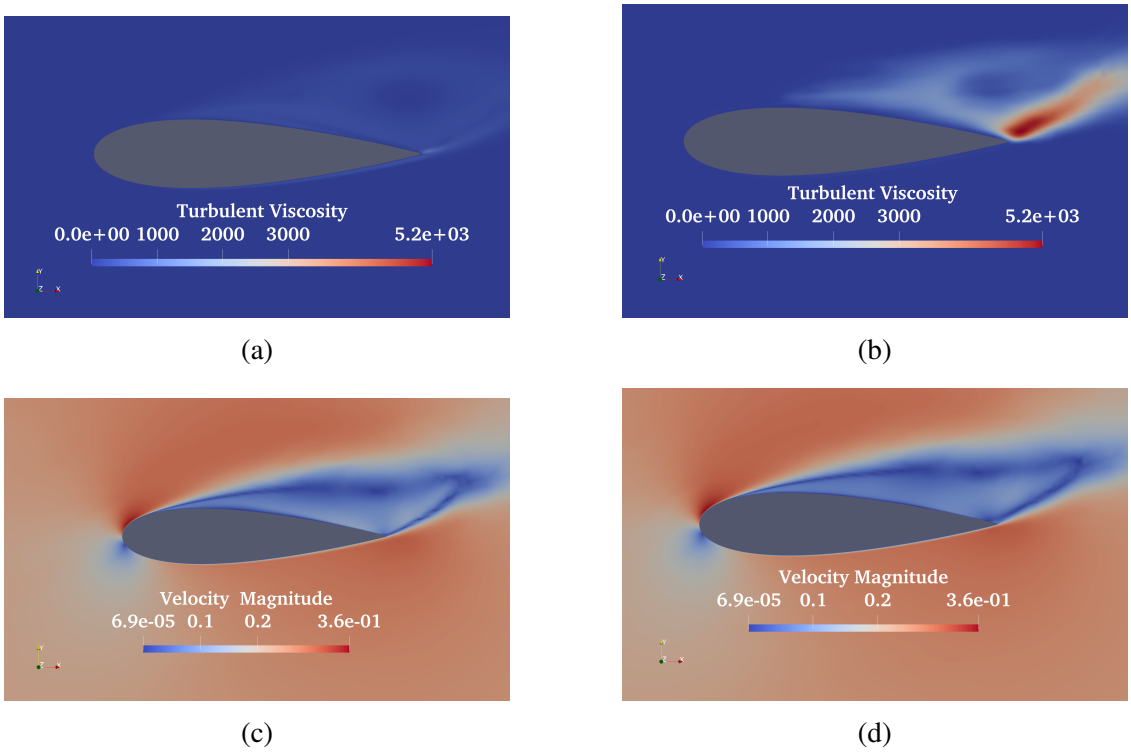


Figure 6.6: Results of the U-Net-augmented Spalart–Allmaras model for the NACA0021 test case at $\alpha = 8°$. (**a**)Turbulent viscosity FI and (**b**) Turbulent viscosity U-Net. (**c**) Velocity magnitude FI and(**d**) Velocity magnitude U-Net.



(a) (b)

(c) (d)

Figure 6.7: Results of the U-Net-augmented Spalart–Allmaras model for the NACA0021 test case at $\alpha = 16°$. (**a**)Turbulent viscosity FI and (**b**) Turbulent viscosity U-Net. (**c**) Velocity magnitude FI and(**d**) Velocity magnitude U-Net.

Figures 6.6 and 6.7 present the results obtained with the U-Net model for angles of attack $\alpha = 8°$ and $\alpha = 16°$. At $\alpha = 8°$, both the turbulent viscosity and the velocity magnitude fields predicted by the U-Net are in close agreement with those obtained through field inversion. At $\alpha = 16°$, however, the turbulent viscosity field predicted by the U-Net exhibits noticeable discrepancies, particularly near the trailing edge, where higher viscosity values are observed. Despite this, the corresponding velocity magnitude field remains in good agreement with the field inversion results. In particular, the predicted separation point in the velocity field more closely matches the field inversion than the original SA model.

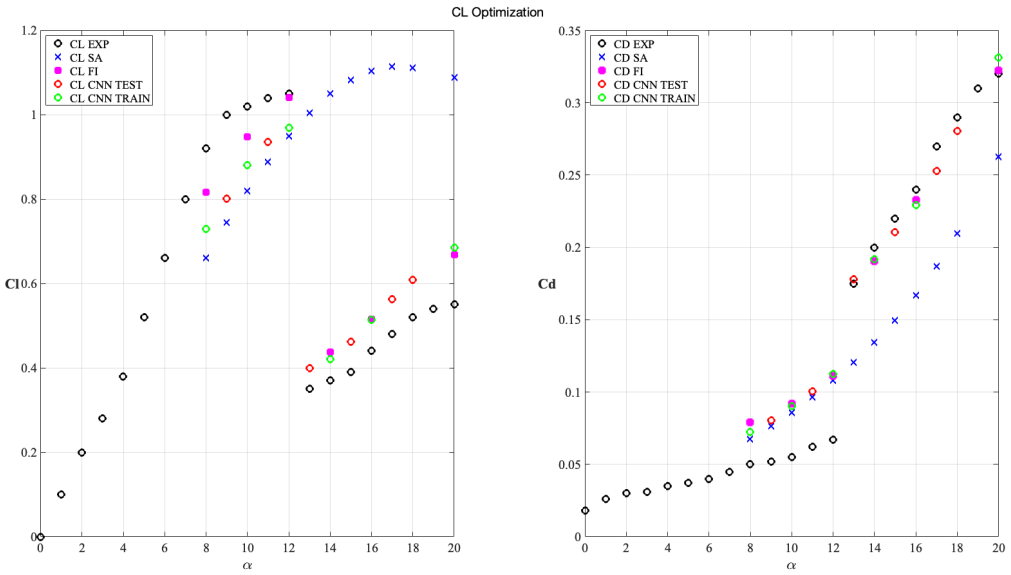


Figure 6.8: Lift and drag coefficients predicted with U-Net-augmented Spalart–Allmaras model for the NACA0021 test case at various angles of attack.

To further highlight the improvements achieved with the U-Net model, Figure 6.8 compares the C_L and C_D results across four models: experimental data, the original SA model, the field inversion-augmented model, and the U-Net model. The model was also tested at angles of attack that were not part of the training set ($\alpha = 9°, 11°, 13°, 15°, 17°$ and $18°$). While no significant improvements were observed in the pre-stall region, substantial improvements were evident in the post-stall region, even for the untrained angles of attack. These findings indicate that the U-Net-augmented model demonstrates strong predictive capabilities, especially in capturing transition and separation phenomena, making it a promising approach for improving flow predictions in complex aerodynamic scenarios.

Chapter 7

Conclusions

This thesis summarizes the work carried out over three years of research. The main goal was to explore the use of data-driven models in the study of turbulent flows, particularly in cases involving transition and separation phenomena. The objective was to understand both the potential and limitations of this approach in such complex scenarios. In the first year, a literature review was conducted to assess the state of the art in this field. This led to the decision to use a framework known as Field Inversion and Machine Learning. Since the aim was to improve existing turbulence models where they typically fail, the thesis begins with a detailed explanation of turbulence and transition phenomena. It also includes an overview of traditional turbulence modeling approaches such as the Reynolds-Averaged Navier-Stokes equations, and zero-, one-, and two-equation models. Transitional models are also discussed. The first year focused heavily on implementing the baseline turbulence model used in this work, the Negative Spalart-Allmaras model. Chapter 3 provides details on both the original and negative versions of the SA model, as well as a variant known as the Spalart-Allmaras Bas-Cakmakcioglu Mura model. The chapter concludes with a description of the test cases selected for the study. Three different cases (T3A transitional flat plate, backward-facing step, SD7003, and NACA0021 airfoils) were used to compare the performance of the standard SA model and its data-augmented versions. The core of the thesis begins in Chapter 4, where the Field Inversion and Machine Learning framework is explained in detail. This includes the implementation of the adjoint model used to compute gradients of the objective function, as well as various mapping functions. Details on the type of goal functions used are also explained. The chapter ends with results from the standard test cases and an example of multi-objective optimization. Chapters 5 and 6 focus on the two data-driven models used in this work. The first approach was a simple feed-forward neural network, which aimed to learn a relationship between the correction field from field inversion and the flow features. This part of the research carried out mainly in the second year, focused on the backward-facing step case and was presented at the ParCFD 2023 conference. Special attention was given to selecting input features for the feed-forward neural network, inspired by previous studies and the Spalart-Allmaras Bas-Cakmakcioglu Mura model.

A technique for selecting the region of interest in the computational domain was also developed to improve training and reduce overfitting. However, results from the feed-forward neural network model were not always successful. In particular, the NACA0012 test case showed poor performance due to numerical instability. This led to a change in direction during the third year, where a new approach was explored using the U-Net model, a neural network architecture originally designed for image segmentation but increasingly used in CFD. In this thesis, the U-Net was used differently than in previous studies and differently from the feed-forward neural network. The U-Net was used as a post-processing tool to directly predict the corrected turbulent viscosity field. This approach showed promising results, especially for the NACA0012 case, where it helped eliminate numerical instabilities and matched well with the results from field inversion. Despite its advantages, the U-Net model also has limitations. Since it requires image-like input data, the CFD results, typically on an unstructured mesh, had to be interpolated onto a structured grid. This interpolation step can introduce errors, especially near the boundary layer where mesh resolution is high. One way to reduce these errors would be to use a finer structured grid with cell sizes matching those of the unstructured mesh, but this would result in high computational costs and was not pursued in this work. The methodology proposed in this thesis requires further investigated. Specifically, a feedback loop should be implemented in which the final results are passed back through the U-Net to evaluate convergence properties and determine whether the model remains stable or instead induces continuous changes. Finally, one suggestion for future research is the use of Graph Neural Networks, which can work directly on unstructured meshes and may overcome some of the limitations of the U-Net. Some early work in this area, such as the study by Belbute-Peres et al. [6], seems promising. It is my hope that this thesis will encourage further exploration of data-driven methods in CFD, leading to smarter models and better predictions of turbulent phenomena, one of the most fascinating and complex challenges in fluid dynamics.

Bibliography

[1] BJ Abu-Ghannam and R Shaw. "Natural transition of boundary layers—the effects of turbulence, pressure gradient, and flow history". In: *Journal of Mechanical Engineering Science* 22.5 (1980), pp. 213–228.

[2] Steven R Allmaras and Forrester T Johnson. "Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model". In: *Seventh international conference on computational fluid dynamics (ICCFD7)*. Vol. 1902. Big Island, HI. 2012, pp. 1–11.

[3] Barrett Baldwin and Timothy Barth. "A one-equation turbulence transport model for high Reynolds number wall-bounded flows". In: *29th aerospace sciences meeting*. 1991, p. 610.

[4] Barrett Baldwin and Harvard Lomax. "Thin-layer approximation and algebraic model for separated turbulent flows". In: *16th aerospace sciences meeting*. 1978, p. 257.

[5] Andrea Beck and Marius Kurz. "A perspective on machine learning methods in turbulence modeling". In: *GAMM-Mitteilungen* 44.1 (2021), e202100002.

[6] Filipe De Avila Belbute-Peres, Thomas Economon, and Zico Kolter. "Combining differentiable PDE solvers and graph neural networks for fluid flow prediction". In: *international conference on machine learning*. PMLR. 2020, pp. 2402–2411.

[7] Shanti Bhushan et al. "Development and validation of a machine learned turbulence model". In: *Energies* 14.5 (2021), p. 1465.

[8] Gaurav R Bokil et al. "Physics-Guided Convolutional Neural Network for Flow Prediction in Heat Exchangers in Electrified Aircraft". In: *AIAA AVIATION FORUM AND ASCEND 2024*. 2024, p. 4108.

[9] Oliver Brenner, Pasha Piroozmand, and Patrick Jenny. "Efficient assimilation of sparse data into RANS-based turbulent flow simulations using a discrete adjoint method". In: *Journal of Computational Physics* 471 (2022), p. 111667.

[10] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. "Machine learning for fluid mechanics". In: *Annual review of fluid mechanics* 52 (2020), pp. 477–508.

[11] Robert H Bush et al. “Recommendations for future efforts in RANS modeling and simulation”. In: *AIAA scitech 2019 forum*. 2019, p. 0317.

[12] Samet C Cakmakcioglu et al. “A revised one-equation transitional model for external aerodynamics”. In: *AIAA Aviation 2020 Forum*. 2020, p. 2706.

[13] Jasvipul S Chawla et al. “Efficiency improvement study for small wind turbines through flow control”. In: *Sustainable Energy Technologies and Assessments* 7 (2014), pp. 195–208.

[14] Soufiane Cherroud et al. “Sparse Bayesian learning of explicit algebraic Reynolds-stress models for turbulent separated flows”. In: *International Journal of Heat and Fluid Flow* 98 (2022), p. 109047.

[15] Amanullah Choudhry, Maziar Arjomandi, and Richard Kelso. “A study of long separation bubble on thick airfoils and its consequent effects”. In: *International Journal of Heat and Fluid Flow* 52 (2015), pp. 84–96.

[16] Paola Cinnella. “Data-driven turbulence modeling”. In: *arXiv preprint arXiv:2404.09074* (2024).

[17] SJ Dhawan and R Narasimha. “Some properties of boundary layer flow during the transition from laminar to turbulent motion”. In: *Journal of Fluid Mechanics* 3.4 (1958), pp. 418–436.

[18] Tony Di Fabbio et al. “Towards the understanding of vortex breakdown for improved RANS turbulence modeling”. In: *Aerospace Science and Technology* 146 (2024), p. 108973.

[19] J Dunham. “Prediction of boundary layer transition on turbomachinery blades”. In: *AGARD meeting on boundary layers in turbomachines, 1972*. 1972.

[20] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. “Turbulence modeling in the age of data”. In: *Annual review of fluid mechanics* 51.1 (2019), pp. 357–377.

[21] Karthikeyan Duraisamy, Ze J Zhang, and Anand Pratap Singh. “New approaches in turbulence and transition modeling using data-driven techniques”. In: *53rd AIAA Aerospace sciences meeting*. 2015, p. 1284.

[22] Hamidreza Eivazi et al. “Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations”. In: *Physics of Fluids* 34.7 (2022), p. 075117.

[23] Andrea Ferrero. “Computational fluid dynamics for aerospace propulsion systems: an approach based on discontinuous finite elements”. PhD thesis. Politecnico di Torino, 2015.

[24] Andrea Ferrero, Angelo Iollo, and Francesco Larocca. “Field inversion for data-augmented RANS modelling in turbomachinery flows”. In: *Computers & Fluids* 201 (2020), p. 104474.

[25] Andrea Ferrero, Angelo Iollo, and Francesco Larocca. "RANS closure approximation by artificial neural networks". In: *ETC 2019-13th European Turbomachinery Conference on Turbomachinery Fluid Dynamics and Thermodynamics*. 2019.

[26] Andrea Ferrero, Francesco Larocca, and Francesca Romana Pennecchi. "Uncertainty propagation in field inversion for turbulence modelling in turbomachinery". In: *2020 IEEE 7th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. IEEE. 2020, pp. 303–308.

[27] Andrea Ferrero et al. "Field inversion and machine learning strategies for improving rans modelling in turbomachinery". In: *ETC 2021-14th European Turbomachinery Conference on Turbomachinery Fluid Dynamics and Thermodynamics*. 2021.

[28] Krzysztof J Fidkowski. "Gradient-based shape optimization for unsteady turbulent simulations using field inversion and machine learning". In: *Aerospace Science and Technology* 129 (2022), p. 107843.

[29] Marshall Galbraith and Miguel Visbal. "Implicit large eddy simulation of low Reynolds number flow past the SD7003 airfoil". In: *46th AIAA aerospace sciences meeting and exhibit*. 2008, p. 225.

[30] Paul Garnier et al. "A review on deep reinforcement learning for fluid mechanics". In: *Computers & Fluids* 225 (2021), p. 104973.

[31] Michael B Giles. *On the iterative solution of adjoint equations*. 2002.

[32] Alessandro Grava et al. "Aerodynamic investigation of a drone propeller in crossflow". In: *AIAA SCITECH 2024 Forum*. 2024, p. 0244.

[33] Gotthlif Heinrich Ludwig Hagen. *Über den Einfluss der Temperatur auf die Bewegung des Wassers in Röhren–*. Druckerei der Königl. Akademie der Wissenschaften, 1854.

[34] Hohum. *Studies of Water passing Obstacles and falling*. Wikimedia Commons. Public Domain. 2023. URL: https://commons.wikimedia.org/wiki/File:Studies_of_Water_passing_Obstacles_and_falling.jpg?uselang=en#Licensing (visited on 08/25/2025).

[35] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.

[36] Chao Jiang et al. "A novel algebraic stress model with machine-learning-assisted parameterization". In: *Energies* 13.1 (2020), p. 258.

[37] Xiaowei Jin et al. "NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations". In: *Journal of Computational Physics* 426 (2021), p. 109951.

[38] Arne V Johansson and Magnus Hallbäck. "Modelling of rapid pressure—strain in Reynolds-stress closures". In: *Journal of Fluid Mechanics* 269 (1994), pp. 143–168.

[39] Duraisamy Karthik and Paul Durbin. "Transition modeling using data driven approaches". In: *Proceedings of the Summer Program*. 2014, p. 427.

[40] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[41] Kemal Koca et al. "Identification of flow phenomena over NACA 4412 wind turbine airfoil at low Reynolds numbers and role of laminar separation bubble on flow evolution". In: *Energy* 144 (2018), pp. 750–764.

[42] Norbert Kroll et al. "Flow simulation and shape optimization for aircraft design". In: *Journal of Computational and applied mathematics* 203.2 (2007), pp. 397–411.

[43] Marius Kurz and Andrea Beck. "A machine learning framework for LES closure terms". In: *arXiv preprint arXiv:2010.03030* (2020).

[44] Robin B Langtry and Florian R Menter. "Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes". In: *AIAA journal* 47.12 (2009), pp. 2894–2906.

[45] Brian Edward Launder and Bahrat I Sharma. "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc". In: *Letters in heat and mass transfer* 1.2 (1974), pp. 131–137.

[46] Yann LeCun et al. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.

[47] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance". In: *Journal of Fluid Mechanics* 807 (2016), pp. 155–166.

[48] JL Lions. *Optimal control of systems governed by partial differential equations*. 1971.

[49] Xiran Liu, Dan Zhao, and Nay Lin Oo. "Comparison studies on aerodynamic performances of a rotating propeller for small-size UAVs". In: *Aerospace Science and Technology* 133 (2023), p. 108148.

[50] Guokun Lyu et al. "Adjoint-based climate model tuning: Application to the planet simulator". In: *Journal of Advances in Modeling Earth Systems* 10.1 (2018), pp. 207–222.

[51] Joaquim RRA Martins, Juan J Alonso, and James J Reuther. "High-fidelity aerostructural design optimization of a supersonic business jet". In: *Journal of Aircraft* 41.3 (2004), pp. 523–530.

[52] RE Mayle and A Schulz. *The path to predicting bypass transition*. Vol. 78729. American Society of Mechanical Engineers, 1996.

[53] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5 (1943), pp. 115–133.

[54] Qingjia Meng, Zhou Jiang, and Jianchun Wang. "Artificial neural network-based subgrid-scale models for LES of compressible turbulent channel flow". In: *Theoretical and Applied Mechanics Letters* 13.1 (2023), p. 100399.

[55] Florian R Menter, Martin Kuntz, Robin Langtry, et al. "Ten years of industrial experience with the SST turbulence model". In: *Turbulence, heat and mass transfer* 4.1 (2003), pp. 625–632.

[56] FR Menter, T Esch, and S Kubacki. "Transition modelling based on local variables". In: *Engineering Turbulence Modelling and Experiments 5*. Elsevier, 2002, pp. 555–564.

[57] Xue Miao et al. "Intelligent mesh refinement based on U-NET for high-fidelity CFD simulation in numerical reactor". In: *Nuclear Engineering and Design* 411 (2023), p. 112411.

[58] Jan Michálek, Michelangelo Monaldi, and Tony Arts. "Aerodynamic performance of a very high lift low pressure turbine airfoil (T106C) at low Reynolds and high Mach number with effect of free stream turbulence intensity". In: (2012).

[59] Aashwin A Mishra and Sharath S Girimaji. "Toward approximating non-local dynamics in single-point pressure–strain correlation closures". In: *Journal of Fluid Mechanics* 811 (2017), pp. 168–188.

[60] Jeremy Morton, Mykel J Kochenderfer, and Freddie D Witherden. "Parameter-conditioned sequential generative modeling of fluid flows". In: *AIAA Journal* 59.3 (2021), pp. 825–841.

[61] Riccardo Mura and Samet C Cakmakcioglu. "A revised one-equation transitional model for external aerodynamics-part i: Theory, validation and base cases". In: *AIAA Aviation 2020 Forum*. 2020, p. 2714.

[62] Luca Muscarà et al. "A Comparison of Local and Global Strategies for Exploiting Field Inversion on Separated Flows at Low Reynolds Number". In: *Applied Sciences* 14.18 (2024), p. 8382.

[63] Eric J Nielsen et al. "An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids". In: *Computers & Fluids* 33.9 (2004), pp. 1131–1155.

[64] Roberto Pacciani et al. “An assessment of the laminar kinetic energy concept for the prediction of high-lift, low-Reynolds number cascade flows”. In: *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy* 225.7 (2011), pp. 995–1003.

[65] Eric J Parish and Karthik Duraisamy. “A paradigm for data-driven predictive modeling using field inversion and machine learning”. In: *Journal of computational physics* 305 (2016), pp. 758–774.

[66] Matteo Pini. “Turbomachinery design optimization using adjoint method and accurate equations of state”. In: (2014).

[67] Olivier Pironneau et al. *Optimal Shape Design for Elliptic Systems [electronic resource]*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984.

[68] Stephen B Pope. “A more general effective-viscosity hypothesis”. In: *Journal of Fluid Mechanics* 72.2 (1975), pp. 331–340.

[69] Stephen B Pope. “Turbulent flows”. In: *Measurement Science and Technology* 12.11 (2001), pp. 2020–2021.

[70] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational physics* 378 (2019), pp. 686–707.

[71] Maximilian Reissmann et al. “Application of gene expression programming to a-posteriori LES modeling of a Taylor Green vortex”. In: *Journal of Computational Physics* 424 (2021), p. 109859.

[72] Osborne Reynolds. “III. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels”. In: *Proceedings of the royal society of London* 35.224-226 (1883), pp. 84–99.

[73] Sal Rodriguez. *Applied Computational Fluid Dynamics and Turbulence Modeling: Practical Tools, Tips and Techniques*. Springer Nature, 2019.

[74] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241.

[75] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386.

[76] Nikolaus Rott. “Note on the history of the Reynolds number”. In: *Annual review of fluid mechanics* 22.1 (1990), pp. 1–12.

[77] Namhee Ryu and Seungjae Min. "Multiobjective optimization with an adaptive weight determination scheme using the concept of hyperplane". In: *International Journal for Numerical Methods in Engineering* 118.6 (2019), pp. 303–319.

[78] Konstantinos Samouchos. "The cut-cell method for the prediction of 2D/3D flows in complex geometries and the adjoint-based shape optimization". PhD thesis. National Technical University of Athens School of Mechanical Engineering, 2022.

[79] Scewing. *Tsunami by hokusai 19th century*. Wikimedia Commons. Public Domain. 2015. URL: https://commons.wikimedia.org/wiki/File:Tsunami_by_hokusai_19th_century.jpg?uselang=en#filelinks (visited on 08/25/2025).

[80] Hermann Schlichting and Joseph Kestin. *Boundary layer theory*. Vol. 121. Springer, 1961.

[81] Tsan-Hsing Shih et al. "A new $k-\epsilon$ eddy viscosity model for high Reynolds number turbulent flows". In: *Computers & fluids* 24.3 (1995), pp. 227–238.

[82] Eiji Shima and Keiichi Kitamura. "Parameter-free simple low-dissipation AUSM-family scheme for all speeds". In: *AIAA journal* 49.8 (2011), pp. 1693–1709.

[83] Junsu Shin et al. "A priori analysis on deep learning of filtered reaction rate". In: *Flow, Turbulence and Combustion* 109.2 (2022), pp. 383–409.

[84] Anand Pratap Singh, Karthikeyan Duraisamy, and Ze Jia Zhang. "Augmentation of turbulence models using field inversion and machine learning". In: *55th AIAA Aerospace Sciences Meeting*. 2017, p. 0993.

[85] A.M.O. Smith et al. *Transition, Pressure Gradient and Stability Theory*. ARC-19322. Douglas Aircraft Company, El Segundo Division, 1956. URL: https://books.google.fr/books?id=qXsxGwAACAAJ.

[86] Philippe Spalart and Steven Allmaras. "A one-equation turbulence model for aerodynamic flows". In: *30th aerospace sciences meeting and exhibit*. 1992, p. 439.

[87] Philippe R Spalart et al. "A new version of detached-eddy simulation, resistant to ambiguous grid densities". In: *Theoretical and computational fluid dynamics* 20 (2006), pp. 181–195.

[88] Charles G Speziale, Sutanu Sarkar, and Thomas B Gatski. "Modelling the pressure–strain correlation of turbulence: an invariant dynamical systems approach". In: *Journal of fluid mechanics* 227 (1991), pp. 245–272.

[89] Vishal Srivastava and Karthik Duraisamy. "Generalizable physics-constrained modeling using learning and inference assisted by feature-space engineering". In: *Physical Review Fluids* 6.12 (2021), p. 124602.

[90] Ralf Stark and Gerald Hagemann. “Current status of numerical flow prediction for separated nozzle flows”. In: (2007).

[91] Nils Thuerey et al. “Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows”. In: *AIAA Journal* 58.1 (2020), pp. 25–36.

[92] JL Van Ingen. “A suggested semi-empirical method for the calculation of the boundary layer transition region”. In: *University of Techn., Dept. of Aerospace Eng., Report UTH-74* (1956).

[93] Venkat Venkatakrishnan. “On the accuracy of limiters and convergence to steady state solutions”. In: *31st Aerospace sciences meeting*. 1993, p. 880.

[94] Jonathan Viquerat et al. “A review on deep reinforcement learning for fluid mechanics: an update”. In: *Physics of Fluids* 34.11 (2022), p. 111301.

[95] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. “Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data”. In: *Physical Review Fluids* 2.3 (2017), p. 034603.

[96] Yuhang Wang, Sergiy Shelyag, and Jörg Schlüter. “A Predictive Model for Turbulence Evolution and Mixing Using Machine Learning”. In: *IEEE Access* (2024).

[97] Jack Weatheritt and Richard Sandberg. “A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship”. In: *Journal of Computational Physics* 325 (2016), pp. 22–37.

[98] Bernard Widrow and Michael A Lehr. “30 years of adaptive neural networks: perceptron, madaline, and backpropagation”. In: *Proceedings of the IEEE* 78.9 (1990), pp. 1415–1442.

[99] David C Wilcox. “Reassessment of the scale-determining equation for advanced turbulence models”. In: *AIAA journal* 26.11 (1988), pp. 1299–1310.

[100] David C Wilcox. “Turbulence modeling for CFD. La Canada, CA: DCW Industries”. In: *Inc, November* 34 (2006).

[101] DC Wilcox. “Turbulence Modeling for CFD”. In: *DCW industries, La Canada* (1998).

[102] Justin Winslow et al. “Basic understanding of airfoil characteristics at low Reynolds numbers (10 4–10 5)”. In: *Journal of aircraft* 55.3 (2018), pp. 1050–1061.

[103] VSASTBCG Yakhot et al. “Development of turbulence models for shear flows by a double expansion technique”. In: *Physics of Fluids A: Fluid Dynamics* 4.7 (1992), pp. 1510–1520.

[104] Muchen Yang and Zhixiang Xiao. “Improving the k–ω–γ–Ar transition model by the field inversion and machine learning framework”. In: *Physics of Fluids* 32.6 (2020), p. 064101.

[105] Yuan Yao et al. “Aerodynamic optimization and analysis of low Reynolds number propeller with gurney flap for ultra-high-altitude unmanned aerial Vehicle”. In: *Applied Sciences* 12.6 (2022), p. 3195.

[106] Yong You, Fan Yu, and Ning Mao. “Fast Prediction and Optimization of Building Wind Environment Using CFD and Deep Learning Method”. In: *Applied Sciences* 14.10 (2024), p. 4087.

[107] Yang Yu, Sheng Chen, and Heng Wei. “Modified UNet with attention gate and dense skip connection for flow field information prediction with porous media”. In: *Flow Measurement and Instrumentation* 89 (2023), p. 102300.

[108] Wangwang Zhang et al. “Analysis of aerodynamic characteristics of propeller systems based on Martian atmospheric environment”. In: *drones* 7.6 (2023), p. 397.

[109] Yaomin Zhao et al. “RANS turbulence model development using CFD-driven machine learning”. In: *Journal of Computational Physics* 411 (2020), p. 109413.

[110] Hang Zhu et al. “Aerodynamic performance of propellers for multirotor unmanned aerial vehicles: Measurement, analysis, and experiment”. In: *Shock and Vibration* 2021.1 (2021), p. 9538647.