A Comparison of Local and Global Strategies for Exploiting Field Inversion on Separated Flows at Low Reynolds Number

Original
A Comparison of Local and Global Strategies for Exploiting Field Inversion on Separated Flows at Low Reynolds Number / Muscara, L., Cisternino, M., Ferrero, A., Iob, A., Larocca, F.. - In: APPLIED SCIENCES. - ISSN 2076-3417. - 14:18(2024). [10.3390/app14188382]

Publisher:
MDPI

Published
DOI:10.3390/app14188382

applied sciences

MDPI

Article

A Comparison of Local and Global Strategies for Exploiting Field Inversion on Separated Flows at Low Reynolds Number

Luca Muscarà [1], Marco Cisternino [2], Andrea Ferrero [1,*], Andrea Iob [2] and Francesco Larocca [1]

[1] Dipartimento di Ingegneria Meccanica e Aerospaziale, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy; luca.muscara@polito.it (L.M.); francesco.larocca@formerfaculty.polito.it (F.L.)
[2] Optimad Srl Via Agostino da Montefeltro, 2, 10134 Turin, Italy; marco.cisternino@optimad.it (M.C.); andrea.iob@optimad.it (A.I.)
* Correspondence: andrea_ferrero@polito.it

Abstract: The prediction of separated flows at low Reynolds numbers is crucial for several applications in aerospace and energy fields. Reynolds-averaged Navier–Stokes (RANS) equations are widely used but their accuracy is limited in the presence of transition or separation. In this work, two different strategies for improving RANS simulations by means of field inversion are discussed. Both strategies require solving an optimization problem to identify a correction field by minimizing the error on some measurable data. The obtained correction field is exploited with two alternative strategies. The first strategy aims to the identification of a relation that allows to express the local correction field as a function of some local flow features. However, this regression can be difficult or even impossible because the relation between the assumed input variables and the local correction could not be a function. For this reason, an alternative is proposed: a U-Net model is trained on the original and corrected RANS results. In this way, it is possible to perform a prediction with the original RANS model and then correct it by means of the U-Net. The methodologies are evaluated and compared on the flow around the NACA0021 and the SD7003 airfoils.

Keywords: RANS; separation; transition; field inversion; machine learning; U-Net

check for updates

Citation: Muscarà, L.; Cisternino, M.; Ferrero, A.; Iob, A.; Larocca, F. A Comparison of Local and Global Strategies for Exploiting Field Inversion on Separated Flows at Low Reynolds Number. *Appl. Sci.* **2024**, *14*, 8382. https://doi.org/10.3390/app14188382

Academic Editor: Wei Huang

Received: 9 August 2024
Revised: 4 September 2024
Accepted: 9 September 2024
Published: 18 September 2024

1. Introduction

1.1. Background

Several applications in aerospace and energy production are characterized by flows with low or moderate values (10^4–10^5) of the Reynolds number. Some examples are represented by flows in advanced air mobility systems like propellers for unmanned aerial vehicles (UAV) [1–6], low-pressure gas turbines in turbofan engines [7,8] or small scale wind turbines [9,10]. In these flows a significant portion of the boundary layer can remain laminar and this increases the risk of separation. Transition to turbulence and separation are phenomena that cannot be easily predicted with the presently available computational techniques. In particular, the design process of industrial components requires to evaluation of a large number of different geometries with limited computational power. As a result, the use of expensive and accurate scale-resolving techniques like Direct Numerical Simulations (DNS) or Large Eddy Simulations (LES) is limited in the design process and is avoided in the preliminary design. On the contrary, Reynolds-averaged Navier–Stokes (RANS) models are widely used for industrial applications because of their limited computational requirements and they are expected to remain crucial in the aerospace industry for several years [11].

1.2. Motivation

The RANS models provide reasonable estimates of the average flow field but their accuracy is limited in the presence of transition or separation since they rely on some simplifications which are approximately satisfied only in fully turbulent and attached flows. In recent years, several research efforts have been devoted to the development of new turbulence models that can avoid the limitations of classical RANS models. On one side, the growing computational power allows us to perform scale-resolving simulations which can provide high-fidelity databases and insight into turbulence physics. On the other side, the development of new machine learning techniques also paves the way for the application of these techniques in the prediction of turbulent separated flows. The use of machine learning in fluid dynamics has been investigated with several strategies [12]: field inversion and machine learning [13,14], correction of the Boussinesq assumption in the computation of the Reynolds stresses [15,16], evaluation of the subgrid stresses in LES [17,18], RANS modeling of vortex breakdown in complex flows [19], transition modeling [8,20]. The original field inversion and machine learning (FIML) paradigm was originally introduced by [13], who proposed to correct the RANS error by introducing a multiplier in the production term of the turbulent eddy viscosity transport equation. This approach was applied and investigated for several applications and provided encouraging results. However, in some problems, it is difficult to perform the regression step through machine learning because it is not clear what input flow features should be considered and it could be impossible to find a set of local flow features that allows the evaluation of the local correction factor through a function. For this reason, the FIML paradigm is considered in this work but then an alternative approach is investigated.

1.3. Main Contribution

The idea behind this work is that field inversion can be used to exploit easily measurable data to correct the entire flow field and to improve the prediction of variables that cannot be easily measured. Then it is possible to exploit these improved fields in two ways: the first approach is the classical FIML approach in which machine learning is used to identify a function of some local flow features that can provide the local value of the correction [8,21,22]. In this work, we propose a second approach involving the training of a U-Net model, a type of convolutional neural network (CNN) commonly used to accelerate computations in computational fluid dynamics (CFD). The U-Net model has demonstrated significant potential in various CFD applications. For instance, the authors in [23] employed the U-Net model for fast and precise prediction and optimization of wind environments around buildings. Similarly, a modified U-Net architecture has been successfully applied to predict flow field information in porous media, showcasing the ability of deep learning to efficiently handle complex fluid dynamics problems [24]. Beyond flow field prediction, U-Net architectures have also been employed in thermal–hydraulic analysis for reactor simulations. An example of this is the Unet-IMR, an intelligent mesh refinement method that improves mesh quality by predicting and refining poor mesh regions, thereby enhancing simulation accuracy in complex fluid dynamics scenarios [25].

High-fidelity CFD simulations are essential for the analysis and optimization of components like heat exchangers in thermal management systems, particularly in electric aircraft propulsion. These simulations, however, are computationally intensive, especially 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 integrating a physics-informed approach, this CNN model achieved high accuracy in predicting velocity, pressure, and temperature distributions while adhering to governing equations and boundary conditions [26].

In our study, we utilize the U-Net model to learn and correct the turbulent eddy viscosity field obtained from the original RANS model, thereby producing an augmented flow field provided by the field inversion procedure. Unlike approaches that seek to develop analytical corrections for the RANS model, our method focuses on providing a global correction to the entire RANS flow field.

The motivation behind adopting the second strategy originates from the limitations of using a simple feed-forward neural network (FFNN). By using an FFNN, there is no guarantee that a function describing the correction field obtained through optimization will exist. This is primarily due to the complexity involved in selecting the input variables, a task that is challenging and may not yield a set of inputs that are universally optimal across different test cases. Another issue arises when integrating the model into CFD software. Since the training is based on a database containing results at convergence, the model must handle numerical transients and input values that were not encountered during training and may significantly deviate from those seen in the training data. This can lead to numerical instability, which can be difficult to mitigate, even by reducing the CFL number.

The second approach, using a U-Net model, addresses these concerns more effectively because it is applied as a post-processing tool. Unlike the FFNN, the U-Net does not attempt to establish a relationship between the correction field and specific input variables. Instead, it performs a global correction based on the original RANS model fields to produce an augmented field that represents the optimal solution. This corrected field is then imposed on the CFD solver, and if it is accurately determined, the solver converges within a few iterations. The only drawback of this method is the interpolation process required to obtain the appropriate input shape for the U-Net model. All the details of these methods are explained in the dedicated sections of this paper. We believe that the second approach represents the main contribution of this work, as it overcomes the disadvantages associated with the traditional use of a simple FFNN in the context of the FIML paradigm.

To implement these methods, we utilized two test cases: the NACA0021 and SD7003 airfoils. We selected these airfoils primarily because experimental results are available for these configurations. Furthermore, these airfoils are widely recognized for their significance in studying separation and transition phenomena at low Reynolds numbers, as they are used in various applications. This makes them ideal candidates for evaluating our strategies.

1.4. Work Outline

In Section 2, the proposed methodology is described. In particular, in Section 2.1, the baseline Spalart–Allmaras [27] RANS model is described. In Section 2.2, the field inversion procedure is explained, while Section 2.2.1 discusses the definition and properties of the goal function. In Section 2.3, the machine learning regression which is used to identify the local model correction is described, and Section 2.3.1 presents a discussion on the selection of input variables for the feed-forward neural network. In Section 2.4, the U-Net approach used to perform global corrections to the RANS prediction is introduced. Section 3 presents the findings of this study. In Section 3.1, the details of the computational mesh used for the simulations are provided. The numerical results for the SD7003 and NACA0021 test cases are discussed in Sections 3.2 and 3.3, respectively. For the SD7003 airfoil, Section 3.2.1 is dedicated to the results obtained using different definitions of the goal function. Additionally, Sections 3.2.2 and 3.2.3 showcase the results of the two different strategies, namely the feed-forward neural network and U-Net models. Similarly, for the NACA0021 airfoil, the results are presented in Sections 3.3.1 and 3.3.2. Finally, conclusions and future perspectives are reported in Section 4.

2. Methods

2.1. Baseline RANS Model

In our study, we utilize the Spalart–Allmaras (SA) turbulence model without the trip terms f_{t1} and f_{t2}. The original SA model allows for the imposition of transition locations via the trip term f_{t1} as defined in [28]. However, this approach is seldom used in the literature because the transition location is generally unknown a priori. Additionally, activating f_{t1} necessitates the use of a second term, f_{t2}, to delay natural transition and make f_{t1} effective. By omitting these trip terms, the SA model performs well at high Reynolds numbers but struggles to predict transition and separation at lower Reynolds numbers. This often results in an overestimation of turbulent eddy viscosity for such flows, making the model a suitable baseline for testing the field inversion approach and evaluating potential improvements.

To address situations where the turbulence eddy viscosity may become negative, particularly in coarse grids and transient states, we employ the negative Spalart–Allmaras model as proposed by Allmaras et al. in [27]. The negative SA model is designed to handle undershoots in the turbulence solution that might occur at the edges of boundary layers and wakes. This is achieved through a continuation of the SA model into the realm of negative $\tilde{\nu}$ solutions. This formulation ensures non-negative solutions and addresses issues related to under-resolved grids and non-physical transient states in discrete settings.

The governing equations used in our simulations are the mass-averaged RANS equations, which are expressed as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{1}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} \tag{2}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (\mathbf{u}(E + p)) = \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{u} - \mathbf{q}) \tag{3}$$

$$\frac{\partial \rho \tilde{\nu}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \tilde{\nu}) = \rho(P - D) + \frac{1}{\sigma} \nabla \cdot (\rho(\nu + \tilde{\nu}) \nabla \tilde{\nu}) + \frac{cb_2}{\sigma} \rho (\nabla \tilde{\nu})^2 - \frac{1}{\sigma}(\nu + \tilde{\nu}) \nabla \rho \cdot \nabla \tilde{\nu} \tag{4}$$

Since we are using the negative S-A model, in cases where $\tilde{\nu}$ becomes negative, the following equation is solved instead of (4):

$$\frac{\partial \rho \tilde{\nu}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \tilde{\nu}) = \rho(P_n - D_n) + \frac{1}{\sigma} \nabla \cdot (\rho(\nu + \tilde{\nu} f_n) \nabla \tilde{\nu}) + \frac{cb_2}{\sigma} \rho (\nabla \tilde{\nu})^2 - \frac{1}{\sigma}(\nu + \tilde{\nu} f_n) \nabla \rho \cdot \nabla \tilde{\nu} \tag{5}$$

where ρ, $\mathbf{u}$, p, E, ν, $\tilde{\nu}$, $\mathbf{x}$, and t denote density, velocity, pressure, total energy per unit volume, molecular viscosity, modified eddy viscosity, spatial position, and time, respectively. A fluid with a constant specific heat ratio γ and constant viscosity is considered. The equation for the energy is given by:

$$E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho |\mathbf{u}|^2 \tag{6}$$

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

$$\tau_{ij} = 2\rho(\nu + \tilde{\nu} f_{v1}) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right) \tag{7}$$

where ν denotes molecular viscosity, $\tilde{\nu}$ denotes modified eddy viscosity, and f_{v1} is a function of the viscosity ratio.

The production P and destruction D terms in the turbulence model are computed as follows:

$$P = c_{b1} \tilde{S} \tilde{\nu} \tag{8}$$

$$D = c_{w1} f_w \left(\frac{\tilde{\nu}}{d} \right)^2 \tag{9}$$

where f_w is given by:

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{1/6} \tag{10}$$

$$g = r + c_{w2}(r^6 - r) \tag{11}$$

$$r = min\left(\frac{\tilde{\nu}}{\tilde{S}^2 k^2 d^2}, r_{lim}\right) \tag{12}$$

The term $\tilde{S}$ represents the modified vorticity magnitude defined as:

$$\tilde{S} = \begin{cases} S + \overline{S} & : \overline{S} \geq -c_{v2} S \\ S + \frac{S({c_{v2}}^2 S + c_{v3}\overline{S})}{(c_{v3} - 2c_{v2})S - S} & : \overline{S} < -c_{v2} S \end{cases} \tag{13}$$

where S is the vorticity magnitude and $\overline{S}$ is:

$$\overline{S} = \frac{\tilde{\nu}}{k^2 d^2} f_{v2} \tag{14}$$

The functions f_{v1} and f_{v2} depend on the viscosity ratio $\chi = \frac{\tilde{\nu}}{\nu}$:

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \tag{15}$$

$$f_{v2} = 1 - \frac{1}{1 + \chi f_{v1}} \tag{16}$$

In the case of the negative model, the production P_n and destruction D_n term are written as follows:

$$P_n = c_{b1}(1 - c_{t3}) S \tilde{\nu} \tag{17}$$

$$D_n = -c_{w1}\left(\frac{\tilde{\nu}}{d}\right)^2 \tag{18}$$

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

$$f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3} \tag{19}$$

The constants σ, c_{b1}, c_{b2}, c_{v1}, and c_{w1}, c_{t3} and c_{n1} are defined in [27].

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

$$\mathbf{q} = -\left(\frac{c_p \mu}{\mathrm{Pr}} + \frac{c_p \rho \tilde{\nu}}{\mathrm{Pr}_t}\right) \nabla T \tag{20}$$

where T denotes the temperature, c_p the constant pressure specific heat capacity, Pr the Prandtl number, and Pr_t the turbulent Prandtl number. For the test cases considered, corresponding to experiments with air, the following values are assumed: $\gamma = 1.4$, $Pr = 0.72$, and $Pr_t = 0.9$.

2.2. Field Inversion

Recently, there has been a notable shift in turbulence modeling towards data-driven approaches, leveraging machine learning techniques to enhance the accuracy of current models. Several reviews have explored these advancements within the computational fluid dynamics (CFD) community, highlighting how these new algorithms can reduce costs and improve prediction accuracy [29–31]. By incorporating data-driven techniques, such as Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Gaussian Process Regression (GPR), and Genetic Expression Programming (GEP), turbulence models can potentially better predict turbulent flows and augment RANS models with additional experimental or simulation data.

According to [32], there are three primary approaches to implementing data-driven algorithms in turbulence modeling. Firstly, these algorithms are used to estimate parameters within turbulence models, often fine-tuned based on experimental data, particularly focusing on the Reynolds stress tensor and the Boussinesq hypothesis [33]. This approach includes methods that enhance neural networks with Galilean invariance, thereby increasing flexibility in handling fluid flow complexities [15,34–36]. Secondly, traditional closure models are replaced with fully connected neural networks (FCNN) or networks employing gated recurrent units (GRU) to address stability issues over longer time frames [37–39]. Finally, a third approach substitutes the entire PDE-based model with autoencoders or Physical Informed Neural Networks (PINNs) [40–43].

One particularly promising data-driven approach is the field inversion and machine learning paradigm [13,20,22], which is based on the solution of an optimization problem in which high fidelity data are used to find a correction field to the production term of a baseline RANS model. This correction field is then analyzed through machine learning tools to express the local correction as a function of local flow variables. FIML has shown significant potential in improving the accuracy of RANS models by integrating additional experimental or simulated data. Various studies have demonstrated its effectiveness across different applications. For example, Fidkowski [44] used FIML in gradient-based shape optimization by replacing RANS models, while Yang and Xiao [45] enhanced a four-equation transition model. Additionally, Ferrero et al. [8,21] applied FIML to improve RANS modeling in turbomachinery flows, and Brenner et al. [46] utilized it in incompressible turbulent flow to directly adjust turbulent viscosity.

In this work, we apply the field inversion approach to enhance the SA model. The classical field inversion approach modifies the production term of the SA model to improve its transition prediction capabilities. Specifically, the production term P is adjusted using a correction field β which is inferred through an optimization process. This process involves solving an inverse problem to minimize the discrepancy between model predictions and experimental data.

The modified SA model is expressed as:

$$\frac{\partial \rho\tilde{\nu}}{\partial t} + \nabla\cdot(\rho\mathbf{u}\tilde{\nu}) = \rho[h(\beta)P - D] + \frac{1}{\sigma}\nabla\cdot(\rho(\nu+\tilde{\nu})\nabla\tilde{\nu}) + \frac{cb_2}{\sigma}\rho(\nabla\tilde{\nu})^2 - \frac{1}{\sigma}(\nu+\tilde{\nu})\nabla\rho\cdot\nabla\tilde{\nu} \tag{21}$$

where $h(\beta)$ is a function mapping the correction field. The mapping function ensures that β remains non-negative, following the approach outlined in [46]. The equations defining this mapping function are provided below:

$$h(\beta) = \begin{cases} \beta + 1 & : \beta \geq 0 \\ \exp(\beta) & : \beta < 0 \end{cases} \tag{22}$$

The correction field β is determined through the optimization problem:

$$\beta^* = \arg\min_{\beta} G(\tilde{u}, u_{exp}) \text{ s.t.} \tag{23}$$

$$R(u(x,\beta(x)),\beta(x)) = 0 \tag{24}$$

where G denotes the objective function and R represents the residuals of the equations.

Since we are 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:

$$\frac{\partial \rho\tilde{\nu}}{\partial t} + \nabla\cdot(\rho\mathbf{u}\tilde{\nu}) = \rho(h(\beta)P_n - D_n) + \frac{1}{\sigma}\nabla\cdot(\rho(\nu+\tilde{\nu}f_n)\nabla\tilde{\nu}) + \frac{cb_2}{\sigma}\rho(\nabla\tilde{\nu})^2 - \frac{1}{\sigma}(\nu+\tilde{\nu}f_n)\nabla\rho\cdot\nabla\tilde{\nu} \tag{25}$$

To solve an optimization problem, evaluating the gradient of the objective function with respect to the correction field is essential. In the context of CFD, the discrete adjoint method is particularly efficient for this purpose.

The discrete adjoint method is effective because it allows for the calculation of gradients with respect to a large number of design variables using only one additional adjoint simulation, rather than requiring multiple forward simulations. This approach not only reduces computational costs but also enhances the accuracy and efficiency of gradient computations. By solving the adjoint equations discretely, we obtain precise sensitivity information crucial for refining the optimization process. The equations governing this approach are detailed in the following:

$$\frac{dG}{d\beta} = \frac{\partial G}{\partial \beta} + \psi^T \frac{\partial R}{\partial \beta} \tag{26}$$

$$\left[\frac{\partial R}{\partial \tilde{u}}\right]^T \psi + \left[\frac{\partial G}{\partial \tilde{u}}\right]^T = 0 \tag{27}$$

where ψ represents the adjoint field.

Classically, the discrete adjoint method involves solving a linear system directly. While this method can be effective, it may struggle with convergence issues in larger or more complex problems, particularly when dealing with viscous or three-dimensional flows. The process often requires high memory usage and can become inefficient as the number of Krylov subspaces increases with each optimization iteration.

To overcome these difficulties, a promising alternative is the use of pseudo-time stepping for solving adjoint Equations [47]. This approach leverages the mathematical concept of duality between flow and adjoint systems and allows the adjoint field to evolve in pseudo-time. By applying the same time-marching algorithms used for the primal equations, or other time integration schemes, this method provides flexibility in timestep selection. Unlike physical time, the choice of timestep in pseudo-time is arbitrary, enabling both fixed and increasing timestep techniques. This method not only mitigates convergence issues but also manages numerical transients effectively, making it a robust solution for handling complex adjoint problems. As a result, Equation (27) can be expressed as follows:

$$\Delta V \frac{\partial \psi}{\partial t} + \left[\frac{\partial R}{\partial \tilde{u}}\right]^T \psi + \left[\frac{\partial G}{\partial \tilde{u}}\right]^T = 0 \tag{28}$$

where ΔV represents the discrete cell volumes of the mesh.

Once the gradient is found, we update the correction field through the optimization iteration using a modified version of the Gauss–Newton method, as detailed in [46]. The following equations illustrate this process:

$$\beta^{k+1} = \beta^k - \omega \frac{G^k - G_{tol}}{\frac{dG}{d\beta}\Big|^k \cdot \frac{dG}{d\beta}\Big|^k} \frac{dG}{d\beta}\Big|^k \tag{29}$$

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

2.2.1. Goal Function Selection

The goal function that drives the optimization procedure in field inversion is crucial for finding physically consistent corrections. This function must meet some important criteria:

- It should be easy to measure with sufficient accuracy.
- It should be closely related to the performance of the system.
- It should be sensible to the correction introduced by field inversion.

The basic idea is that the goal function should represent the RANS error on a variable that can be easily measured or evaluated through scale-resolving simulations (such as force coefficients or pressure and skin friction distribution). This information is then used to find corrections for other variables (for example, the modified turbulent eddy viscosity), which cannot be directly measured in an experiment.

It is important to note that experimental results can also contain uncertainties, which may affect the optimization process. In a previous study [48], a sensitivity analysis was performed by perturbing the experimental data to identify the regions within the domain where these uncertainties have the greatest impact on the correction provided by field inversion. This approach can be used to define accuracy requirements that the reference experimental data should satisfy to be suitable for the field inversion procedure.

The goal function is augmented with a penalization term, often referred to as Tikhonov regularization. This term helps prevent extreme values in the correction field during optimization, ensuring that the correction field is active only in regions where it can significantly contribute to the reduction in the error. An example of a goal function based on the pressure distribution, including the penalization term, is shown below:

$$G = \int_{\partial\Omega} (Cp - Cp_{exp})^2 dl + \lambda \int_{\Omega} (\beta - \beta_0)^2 d\Omega \tag{30}$$

The first term is a line integral performed along the airfoil's wall, which evaluates the norm-2 error in the wall pressure distribution. The second term 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 key strengths of the field inversion approach is its flexibility in optimizing the correction field by defining a goal function that can be tailored to various objectives. This flexibility allows for the optimization of different aspects of the flow, whether it is a global performance metric or specific local characteristics. In our work, we utilized both local and global coefficients to define the goal function. Specifically, we focused on skin friction distribution, pressure distribution, lift and drag coefficients. By choosing an appropriate norm, we aggregated these coefficients into a single value that effectively guides the optimization process.

A discussion on the use of global (force coefficients) or local (pressure or skin friction distributions) goal functions is presented in Section 3.2.1. Global goal functions, based on integral quantities, can be more easily measured, but pressure and skin friction distributions can provide more physical insight.

We analyze two test cases: the NACA0021 and the SD7003 airfoil. All results for the turbulent viscosity and velocity fields are non-dimensionalized using the reference kinematic viscosity $\nu_{ref} = \frac{\mu_{ref}}{\rho_{ref}}$ and the reference velocity $V_{ref} = \sqrt{RT_{ref}}$, where μ_{ref} and ρ_{ref} are the reference dynamic viscosity and reference density, respectively, evaluated in the far-field. Here, R is the gas constant for air and T_{ref} is the reference static temperature in the far-field.

For the SD7003 airfoil, we have both local and global coefficients, allowing us to test the effects of defining the goal function using either global or local coefficients. We observed that using global coefficients (lift and drag coefficients) does not effectively predict flow separation, unlike using local coefficients such as skin friction and pressure coefficient.

2.3. Regression for Local Model Corrections

The FIML paradigm involves two main steps. The first step explained in Section 2.2 is an optimization procedure. This procedure aims to find a correction field that alters the source term of the SA equation, specifically the production term in this work. By solving this modified equation alongside the Navier–Stokes equations, we can find a solution that minimizes the objective function, thereby matching the experimental or high-fidelity data used as the objective.

The second step of the classical FIML approach involves using a machine learning algorithm, such as a neural network, to augment the result and find a correlation for the correction field. Specifically, a feed-forward neural network is typically used.

A FFNN is a type of artificial neural network where connections between nodes do not form cycles. It consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of units called neurons, which process input data by applying weights and activation functions to produce an output. The FFNN learns to map input variables to the desired output by adjusting the weights through a process called backpropagation during training.

In this work, we utilized a FFNN with 5 hidden layers, each consisting of 10 nodes. The FFNN takes 5 input features (discussed in Section 2.3.1) and outputs the correction field, as shown in Figure 1.

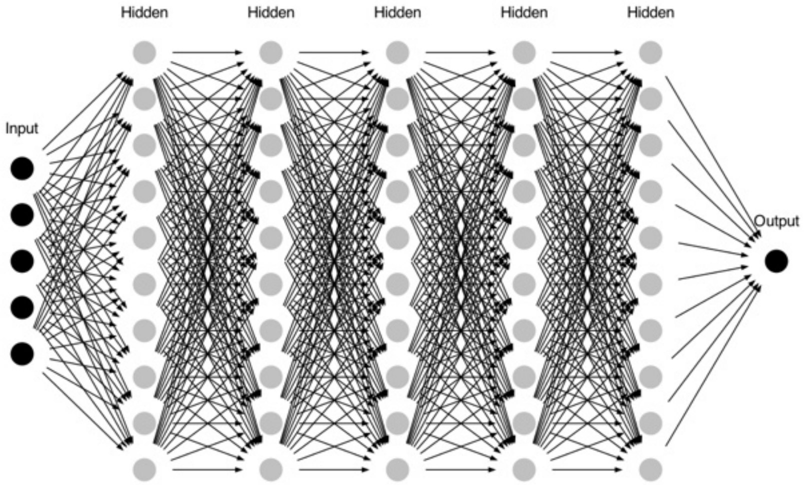


Figure 1. Feed-forward neural network architecture.

We used the Adam backpropagation algorithm, which offers adaptive learning rates, computational efficiency, robustness to hyperparameters, and bias correction, enhancing convergence speed and stability. The Rectified Linear Unit (ReLU) activation function was chosen for its simplicity, computational efficiency, ability to induce sparse activation, and mitigation of the vanishing gradient problem, making it suitable for deeper networks.

The dataset includes simulations from different angles of attack. For SD7003, data from two angles were used, while for NACA0021, six angles were included. The dataset was divided into 70% training and 30% validation.

A variable learning rate was employed for faster initial learning and stable convergence. Early stopping, based on validation loss, was used to prevent overfitting by stopping the training when the validation loss ceased to improve, ensuring the maintenance of the best model parameters.

The strength of the field inversion approach lies in its flexibility to use both global and local coefficients obtained experimentally or from high-fidelity data. Typically, measuring integral coefficients such as lift or drag is straightforward and depends exclusively on the fields calculated at the boundary layer. Consequently, the correction field will primarily alter the equation in the boundary layer region.

When constructing the training dataset for the neural network, it is advisable to select a specific region of the domain rather than using the entire domain. This is because the correction field will remain equal to its initial value in regions where it is not necessary to alter the original SA model. Focusing only on the regions of the domain where the correction field shows significant variations reduces the complexity of the problem and improves the efficiency of the training process.

The correction field is expected to differ from its initial value only near the body. To select the portion of the domain of interest, we utilize a modified version of the shield function that was initially adopted in the Detached Eddy Simulation (DES) [49] model to blend the LES and RANS models. The values of the shield function range between 0 and 1, with 0 indicating a region inside the boundary layer and 1 indicating a region outside the boundary layer. Using this parameter, we can target the region where the correction field is expected to be active.

By leveraging this approach, we select the dataset values from the region of the domain where the modified shield function $fd^{'}$ takes values less than or equal to 0.9. The definition of the modified shield function is given below.

$$rd = \frac{\nu + \tilde{\nu}}{d^2 k^2 |\nabla \mathbf{u}|} \tag{31}$$

where d is the distance function while $k = 0.41$ is the von Kármán costant.

$$fd^{'} = 1 - \tanh(rd^{0.5}) \tag{32}$$

2.3.1. Input Selection

The use of the FFNN aims to find a relation between some local flow features and the local correction, which ideally can be used in any CFD code to solve the modified SA equation over time. This is a critical step, as the choice of input variables greatly influences the results. General rules suggest selecting local variables that are Galilean-invariant and based on RANS simulations. However, the final selection of input variables is up to the researcher, who must consider the specific needs and objectives of the study.

A disadvantage of this approach is that an exact relation might not exist, or the right inputs to describe the correction field might not be found. Even worse, the FFNN might find a relation that does not represent physical behavior, leading to problems during simulations, such as numerical instability during the transient phase.

Different researchers have chosen various sets of input variables depending on their specific needs and objectives. The choice often depends on the particular flow characteristics being studied and the aspects of the turbulence model that require correction. In this work, we choose the following five input variables:

- Modified turbulent viscosity ratio:

$$\nu/(\tilde{\nu} + \nu) \tag{33}$$

- Reynolds thickness number:

$$Re_\theta = Re_v/2.193 \tag{34}$$

where

$$Re_v = \frac{\rho d^2}{\mu} S \tag{35}$$

- Streamwise pressure gradient:

$$\frac{(\nabla p \mathbf{u})d}{|\mathbf{u}|p} \tag{36}$$

- Magnitude of the turbulent viscosity gradient:

$$\frac{|\nabla \tilde{\nu}|d}{(\nu + \tilde{\nu})} \tag{37}$$

- Modified shield function:

$$fd^{'} \tag{38}$$

Since we use non-dimensional variables as input, in Equations (36) and (37), the streamwise pressure gradient and the magnitude of the turbulent viscosity gradient are non-dimensionalized using, respectively, the distance function, the pressure, and the sum of the kinematic viscosity and the turbulent viscosity.

The first input is an index of the intensity of the turbulent viscosity, as also used in [22]. The second input was chosen because it appears in a variant of the SA model called SA-BCM [50]. In this version, the authors use an algebraic function to evaluate an intermittency multiplier for the production term. Since the correction field acts similarly to an intermittency value, we include the Reynolds thickness number as an input to our FFNN.

The third input is 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 separation from the surface. This separation is a key factor in the transition from laminar to turbulent flow. Including the streamwise pressure gradient helps the model predict and correct for regions prone to separation, thus enhancing the turbulence model's accuracy. It could be argued that the use of the streamwise pressure gradient introduces a violation of the Galilean invariance: this problem can be avoided by substituting the absolute velocity in Equation (36) with the velocity relative to the closest solid wall. Since in this work, all the bodies are at rest with respect to the chosen frame of reference, the two velocities coincide.

The fourth input is the non-dimensional magnitude of the turbulent viscosity gradient, reflecting changes in turbulent viscosity and providing information on the spatial variation in turbulence.

Finally, we include the modified shield function, which was used to select the region for constructing the dataset. Including this in the inputs allows us to leverage variations within the boundary layer to establish a meaningful relationship with the correction field.

2.4. U-Net for Global Corrections

As mentioned in Section 2.3.1, establishing a physical relationship between the correction field and the selected inputs is not guaranteed. This can lead to instability during numerical transients which cannot be suppressed even reducing the CFL value. For example, we had good results in the SD7003 test case, but for the NACA0021, the same architecture and choice of inputs did not yield satisfactory outcomes. These results will be discussed in Sections 3.2.2 and 3.3.1.

Therefore, we explored a different approach to mitigate this problem. Staying within the neural network framework, we decided to use a different structure, specifically a convolutional neural network. We opted for the U-Net model due to its ability to handle complex spatial relationships effectively.

The U-Net model is a type of convolutional neural network designed for image segmentation tasks. It consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. The contracting path follows the typical architecture of a CNN, where repeated applications of convolutions and max-pooling operations reduce the spatial dimensions. The expanding path consists of upsampling operations and concatenations with corresponding feature maps from the contracting path, followed by convolutions. This structure allows the network to combine high-resolution features with context information from larger regions, making it particularly suitable for tasks where precise localization is required.

A schematic of the architecture used in this work is shown in Figure 2.

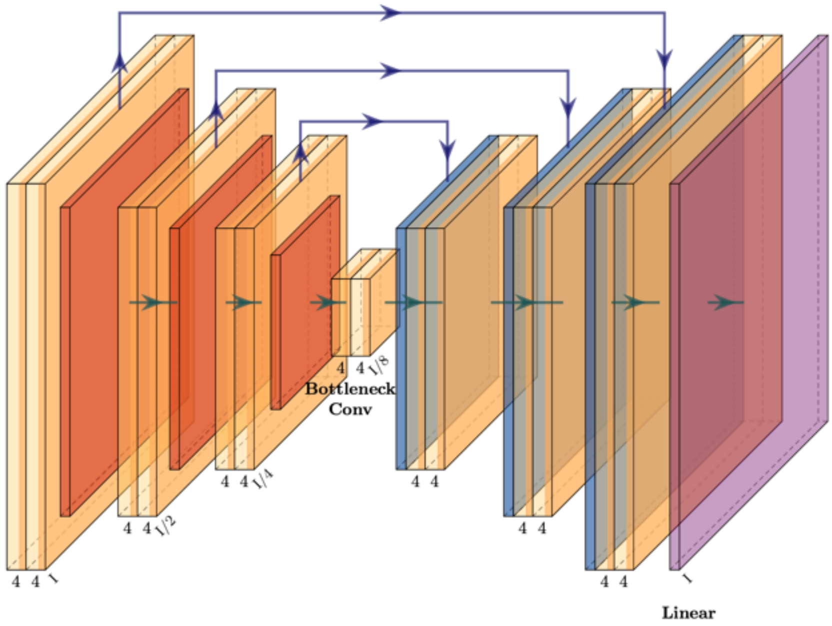


Figure 2. U-Net model architecture.

Inspired by the work of Thuerey et al. [51], we similarly use the U-Net model. The authors 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.

In this work, starting from a converged solution of the original SA model, we use the streamwise and spanwise velocity, pressure, temperature, and turbulent viscosity as inputs to our model with the objective of predicting the corrected turbulent viscosity field. This approach avoids the need to find a local function for the correction field. Instead, we predict the turbulent viscosity field directly, which is the field of interest.

With this approach, we no longer need to alter the SA equation. Instead, we delegate the task to the U-Net model. First of all, a RANS simulation with the original SA model is performed. Then the steady solution is provided to the U-net to compute the corrected eddy 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.

Since convolutional layers typically operate on structured uniform grids, but turbulent CFD simulations are often based on unstructured grids, a preprocessing step is necessary. We extract the fields of interest within a defined region around the airfoil and interpolate these values onto a new structured grid with lower resolution using linear interpolation. This preprocessing step ensures the data are in a suitable format for the U-Net model while managing computational costs.

If the U-Net model accurately predicts the optimal turbulent viscosity, the solver will quickly converge to the correct steady-state solution with minimal iterations, replicating the field inversion results.

There are some disadvantages to this approach. For example, the need to interpolate results onto a structured grid introduces errors in the dataset used for training the model. This step, although necessary for compatibility with convolutional layers, can affect the overall accuracy of the predictions.

According to our initial tests, this approach shows promising results. However, the model's ability to generalize across different airfoil shapes remains untested. Training the model on a variety of airfoil shapes would increase the parameter space, complicating the training process, especially given that we already account for variations in the angle of attack.

The proposed U-Net-based methodology offers a robust alternative to classical FFNN approaches for predicting turbulent viscosity fields in CFD simulations. By leveraging the strengths of the U-Net architecture and careful preprocessing of CFD data, we achieve accurate predictions while mitigating the issues of numerical instability and input variable selection inherent in traditional methods.

3. Results and Discussion

3.1. Computational Mesh

For both airfoils used in this study, the computational meshes were generated using GMSH software 4.11.1 [52]. The mesh for the NACA0021 airfoil is an O-type mesh, featuring a combination of structured and unstructured elements. Figure 3 provides an overall view of the mesh details.

The mesh for the NACA0021 airfoil was created using a combination of structured and unstructured elements to accurately capture the flow characteristics around the airfoil. To maintain solution accuracy while reducing computational cost, the far-field boundary was meshed with larger elements, resulting in a total of 14,824 elements. As in the study by [53], the domain boundaries were placed 20 chord lengths away from the airfoil to prevent any influence on the flow field.

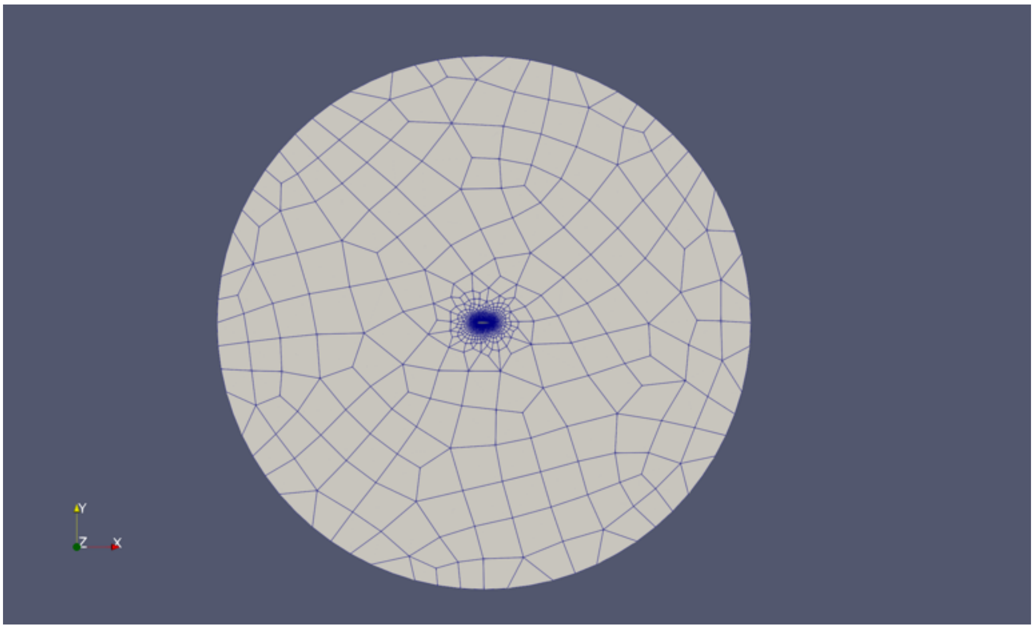


Figure 3. NACA0021 computational mesh.

A structured grid was applied near the airfoil's surface, particularly in the boundary layer region, to achieve precise resolution of the flow, especially close to the wall. The first cell height was determined to meet the minimum y+ requirements for the low Reynolds number turbulence model used in this study. A total of 27 structured layers with linear progression were employed to fully capture the boundary layer thickness; the details are shown in Figure 4. Additionally, transfinite lines were utilized to control point distribution along critical areas, such as the leading and trailing edges, enabling finer mesh refinement in these regions. Regarding the boundary conditions, a far-field boundary condition was applied where the flow enters, a pressure-outlet condition was used at the downstream boundary, and a no-slip boundary condition was imposed on the airfoil surface.

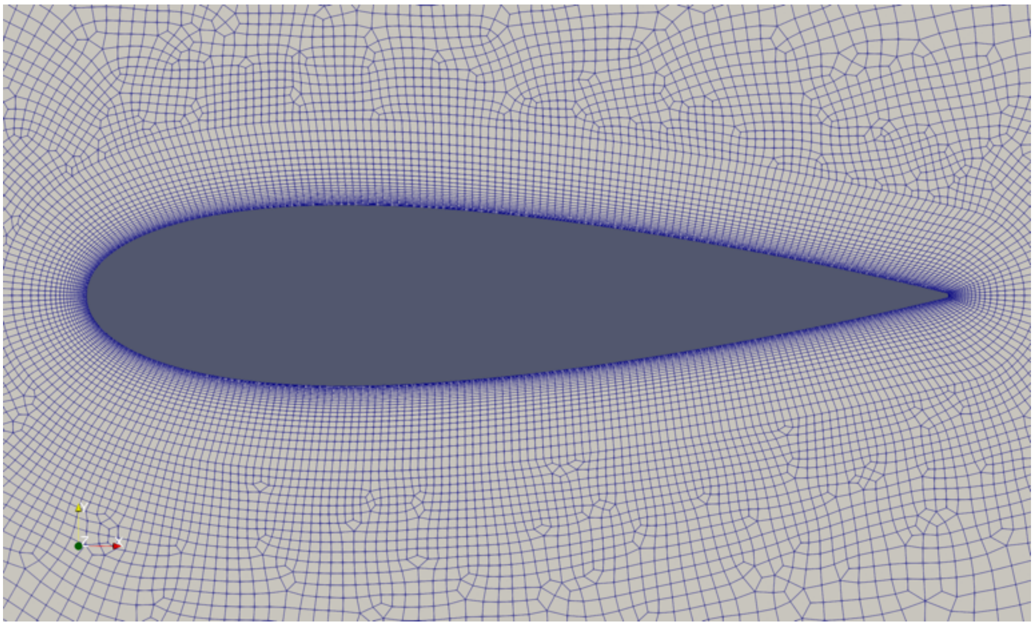


Figure 4. Detailed view of the boundary layer mesh around the NACA0021 airfoil.

For the SD7003 airfoil, a C-type mesh was employed. The overall configuration of this mesh is illustrated in Figure 5. The total number of elements is 149,120.

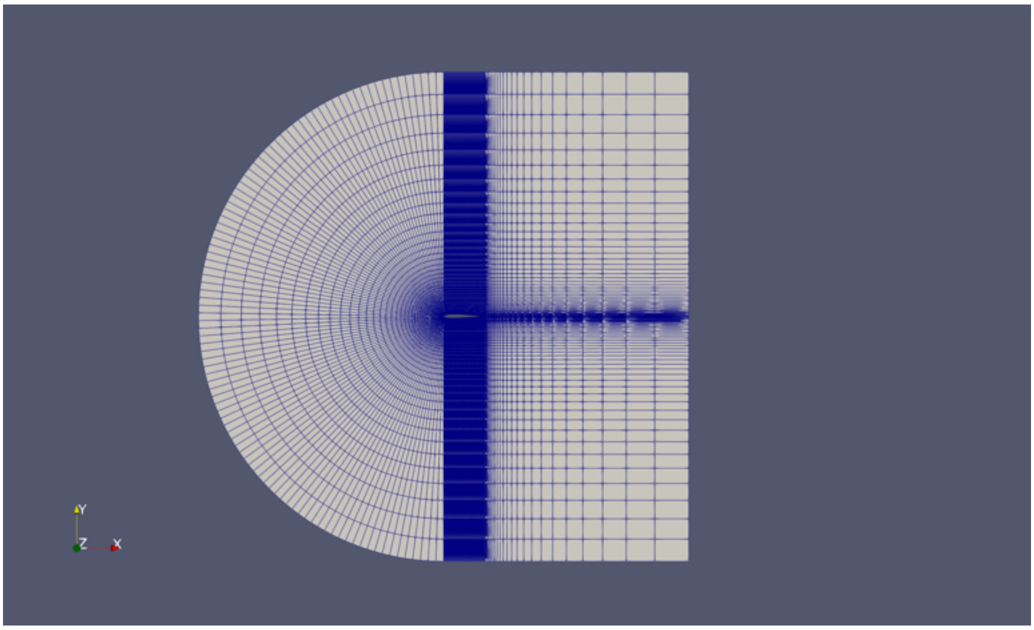


Figure 5. SD7003 computational mesh.

The mesh is entirely structured, with enhanced refinement near the wall to accurately capture boundary layer characteristics. The size of the first cell adjacent to the wall was determined based on the y+ value calculated using the Reynolds number for this simulation. Details of the mesh near the airfoil boundary are shown in Figure 6.

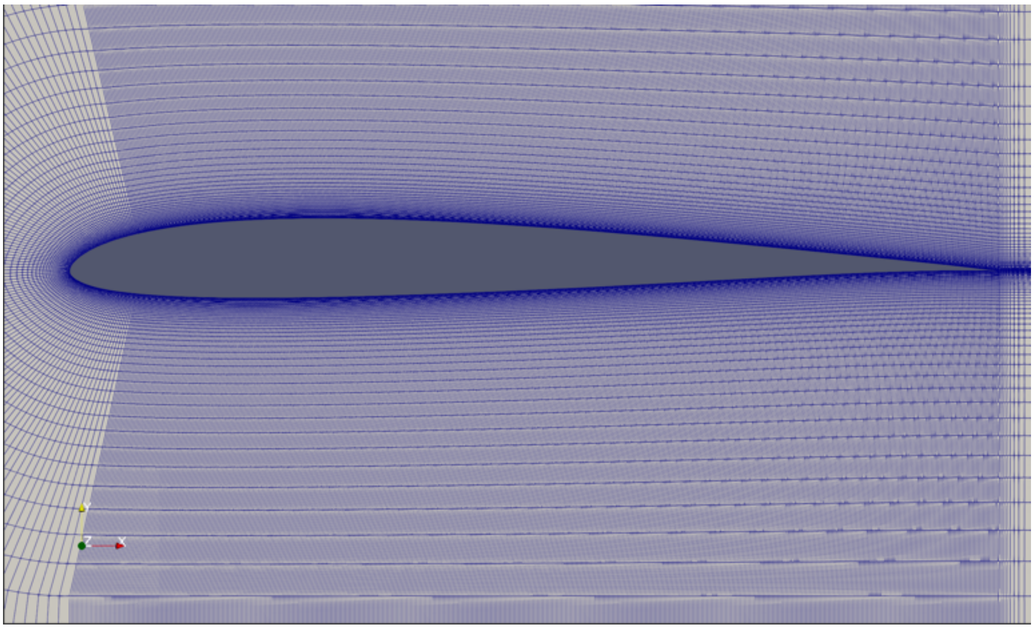


Figure 6. Detailed view of the boundary layer mesh around the SD7003 airfoil.

The boundary conditions used in this test are the same as the ones used for the NACA0021.

3.2. Flow around the SD7003 Airfoil

The SD7003 airfoil, designed by Selig and Donovan, is a well-known airfoil optimized for low Reynolds number applications, typically in the range of 10^4 to 10^5. This airfoil is widely used in small-scale aircraft, unmanned aerial vehicles (UAVs), and model aircraft due to its favorable aerodynamic performance at low speeds. The SD7003 airfoil performs well in low Reynolds number regimes, making it an ideal choice for applications where low-speed aerodynamics are critical. A key characteristic of the SD7003 is its tendency to form laminar separation bubbles. These bubbles occur when laminar flow separates from the suction side of the airfoil, then it becomes turbulent and subsequently reattaches. The presence of these laminar separation bubbles significantly impacts the airfoil's performance, particularly in terms of lift and drag characteristics.

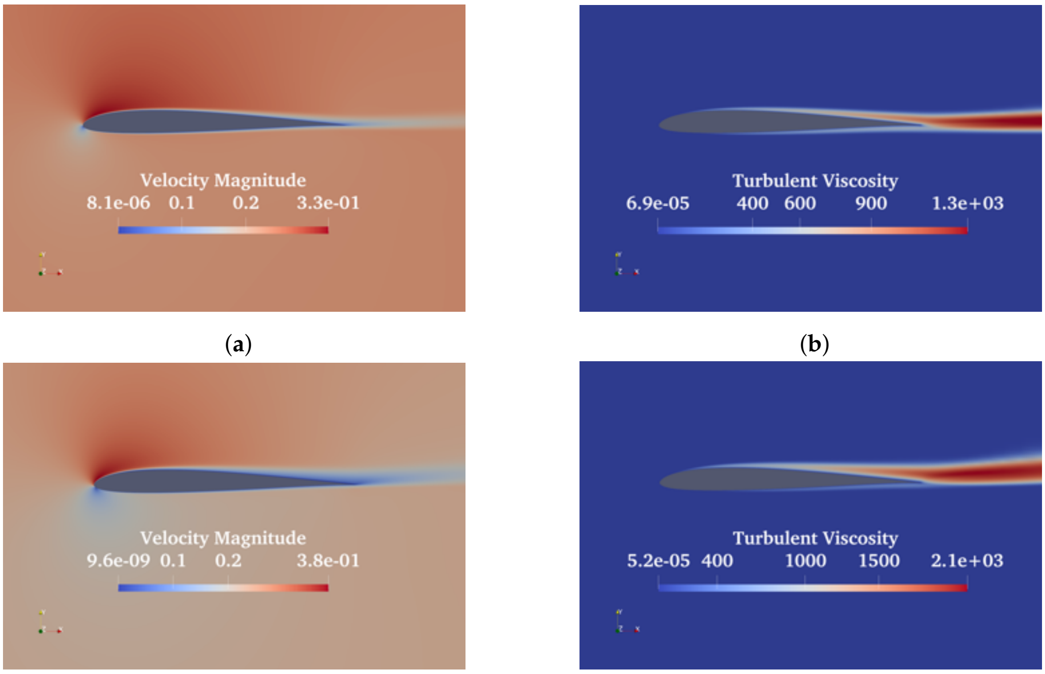
In this study, we utilized the SD7003 airfoil to analyze separation behavior using the Spalart–Allmaras turbulence model. The simulations were conducted at a Reynolds number of 60,000 and angles of attack of 4° and 8°. These conditions were chosen to investigate the flow separation and transition phenomena that are critical for low Reynolds number aerodynamics.

At an angle of attack of 4°, the airfoil is expected to exhibit moderate flow separation, primarily near the trailing edge. At a higher angle of attack of 8°, more pronounced separation and the formation of laminar separation bubbles are expected.

It is well known that the SA model has limitations in accurately predicting separation, especially 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 approaches to verify whether they can improve the baseline model.

By using field inversion, we can identify regions where the SA model deviates from high-fidelity data, allowing us to modify the turbulence model to better capture the separation phenomena.

The results of the original SA model are shown in Figure 7. The turbulent viscosity and velocity fields exhibit similar behavior at both angles of attack. However, we should expect a clear difference, as the separation is anticipated to start near the trailing edge at $\alpha = 4°$ and extend towards the leading edge at $\alpha = 8°$.



(**a**) (**b**)

(**c**) (**d**)

Figure 7. 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 $\alpha = 8°$.

More exhaustive results are presented in Table 1, where CL and CD are compared to the high-fidelity results from a LES simulation [54]. There is not a significant difference in terms of CL for both angles of attack, even though the correct separation is not observed. In the next section, we will also show results in terms of CP and CF. For CD at $\alpha = 8°$, we observed good agreement with the experimental results, while a relative error of 52% is present for $\alpha = 4°$.

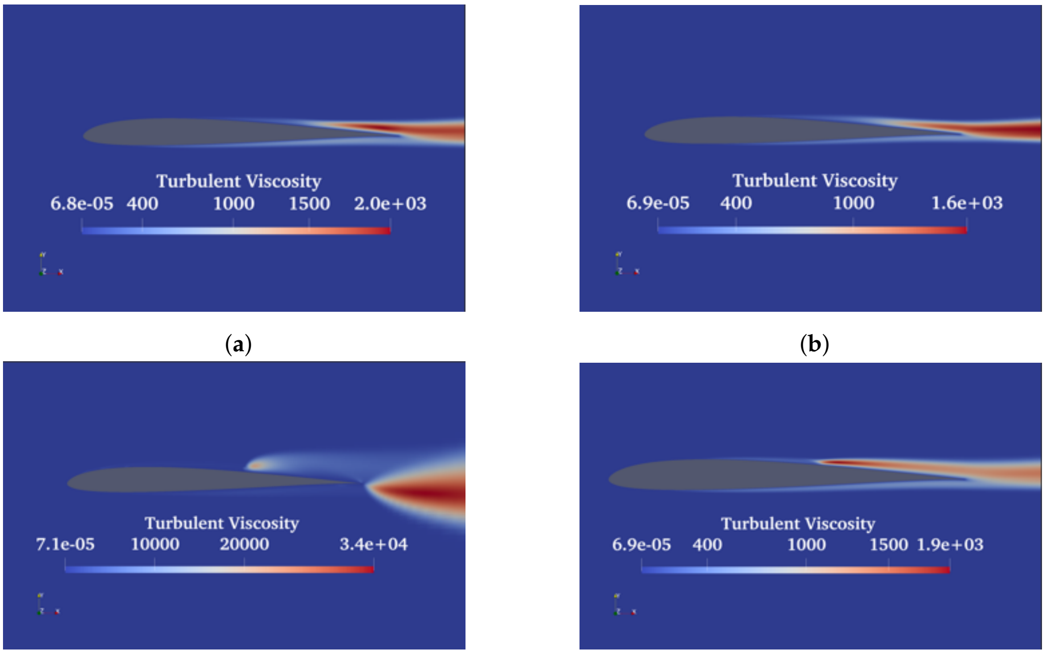
Table 1. Comparison of lift and drag coefficients for SD7003 airfoil.

		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

3.2.1. Comparison of Different Goal Functions

As previously mentioned, we have access to various high-fidelity values for this test case, including wall pressure distribution CP, skin friction distribution CF, lift CL and drag CD coefficients. This allows us to define four different objective functions. We tested all these goal functions and compared the results across the different optimizations. Each optimization alters the turbulent viscosity field differently, consequently affecting all other fields.

The differences are visible in Figure 8 below.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. Turbulent viscosity results after field inversion optimization for the SD7003 airfoil at $\alpha = 4°$. (**a**) CL optimization, (**b**) CD optimization, (**c**) CP optimization, and (**d**) CF optimization.

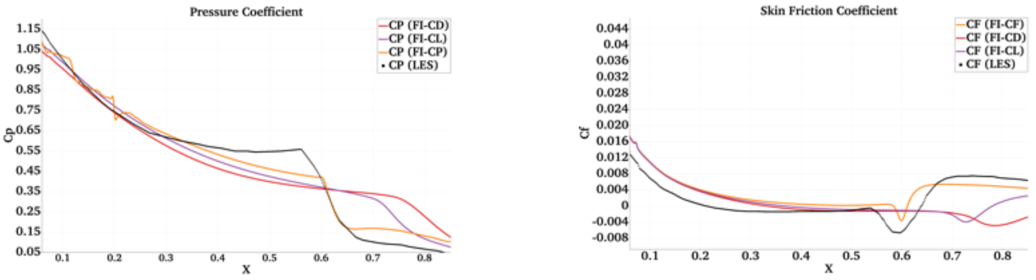
As a result, we observe different outcomes in terms of both local and global coefficients. There is an evident difference in the shape of the pressure and skin friction coefficients, as illustrated in Figure 9.

We also illustrate in Table 2 below the results in terms of CL and CD for each optimization.

Additionally, Figures 10 and 11, along with Table 3, illustrate the differences in turbulent viscosity, as well as local and global coefficients for $\alpha = 8°$.

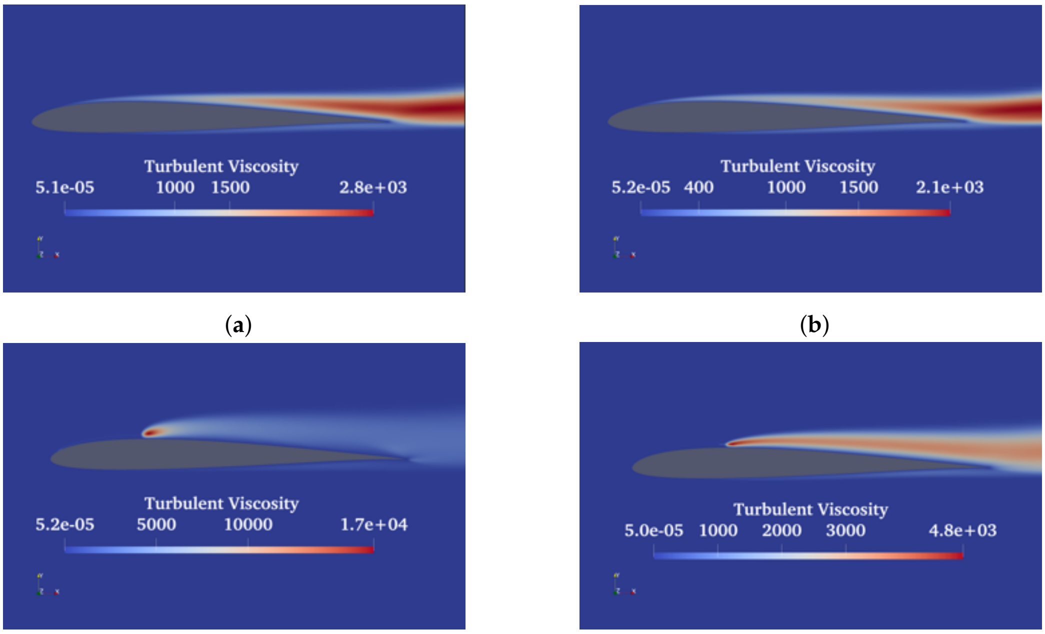
Table 2. Comparison of lift and drag coefficients for SD7003 airfoil at $\alpha = 4^\circ$.

	CL	CD
LES	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



(**a**) (**b**)

Figure 9. Comparison of local coefficients for different goal function definitions for SD7003 at $\alpha = 4^\circ$. (**a**) Pressure coefficient. (**b**) Skin friction coefficient on the suction side.

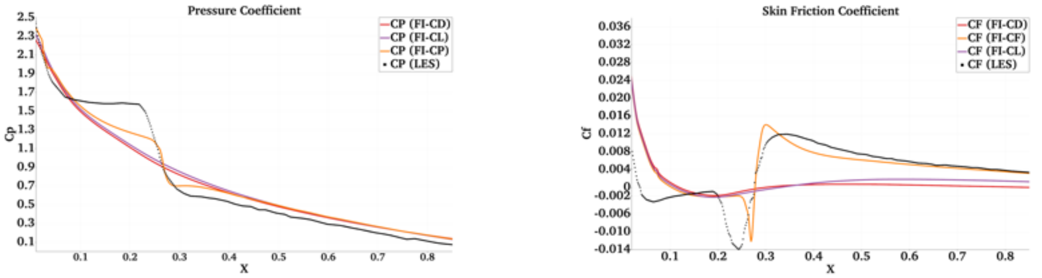


(**a**) (**b**)

(**c**) (**d**)

Figure 10. Turbulent viscosity results after field inversion optimization for the SD7003 airfoil at $\alpha = 8^\circ$. (**a**) CL optimization, (**b**) CD optimization, (**c**) CP optimization, and (**d**) CF optimization.

From both angles of attack, we observed the same effects. The use of objective functions based on the error on global coefficients yields good results in terms of global coefficients prediction but fails to predict the correct shape distributions, missing import flow features like separation. On the contrary, the use of local distributions in the goal function provides more physical insights and allows us to achieve corrected fields that are closer to the reference one.



(**a**) (**b**)

Figure 11. Comparison of local coefficients for different goal function definitions for SD7003 at $\alpha = 8°$. (**a**) Pressure coefficient. (**b**) Skin friction coefficient on the suction side.

Table 3. Comparison of lift and drag coefficients for SD7003 airfoil at $\alpha = 8°$.

	CL	CD
LES [54]	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

3.2.2. Local Model Training

In this section, we present the results obtained using the FFNN for the SD7003 test case. As mentioned earlier, we have ground truth data for two angles of attack for this test case. In Section 3.2.1, we demonstrated the superiority of using local coefficients instead of global ones by showing the different results of field inversion using different goal functions. However, to maintain coherence with the NACA0021 test case for which only global coefficients are available, we train the neural network using only global coefficients. Specifically, we use the results obtained by using the CL error as the objective function. These results are used to train both the FFNN and U-Net models.

The details of the FFNN were already explained in Section 2.3. The loss function is defined as the mean squared error between the simulated correction field and the one obtained from the field inversion process. In Figure 12, we show the trend of the train and validation loss. As explained, we use an early stopping method to automatically stop the training in order to avoid overfitting. We used the R-squared (R^2) coefficient to determine the goodness of the neural network training. The R^2 value measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It is calculated as:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{39}$$

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

$$SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{40}$$

and SS_{tot} is the total sum of squares defined as:

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \overline{y_i})^2 \tag{41}$$

In these equations, n, y_i, $\hat{y}_i$ and $\overline{y_i}$ are, respectively, the total number of data points, the ground truth value, the predicted value and the mean of the ground truth value. The R^2 values range from 0 to 1, where 1 indicates perfect prediction and 0 indicates that the model does not explain any of the variability. Thus, higher R^2 values indicate better fit and more accurate predictions by the model. The value obtained in this training was $R^2 = 0.862$.

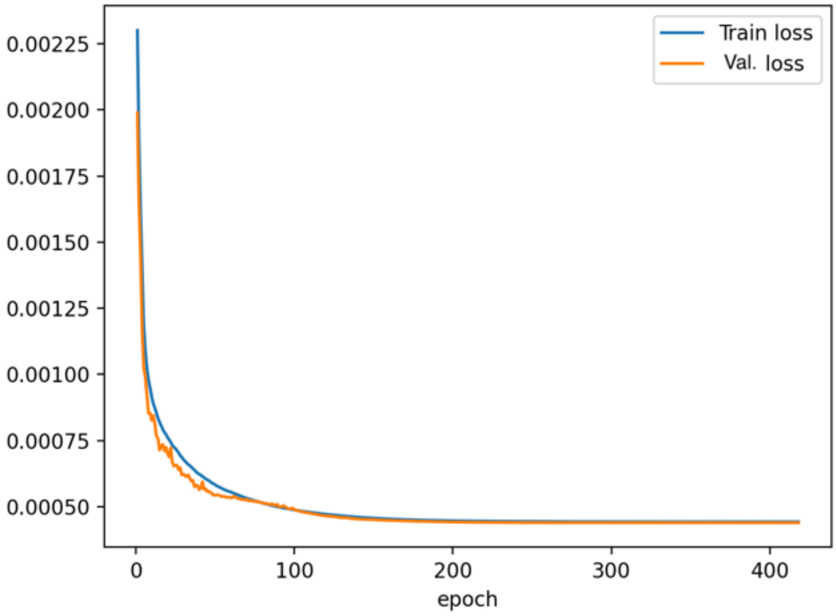


Figure 12. Training and validation loss for the FFNN model trained on SD7003 data.

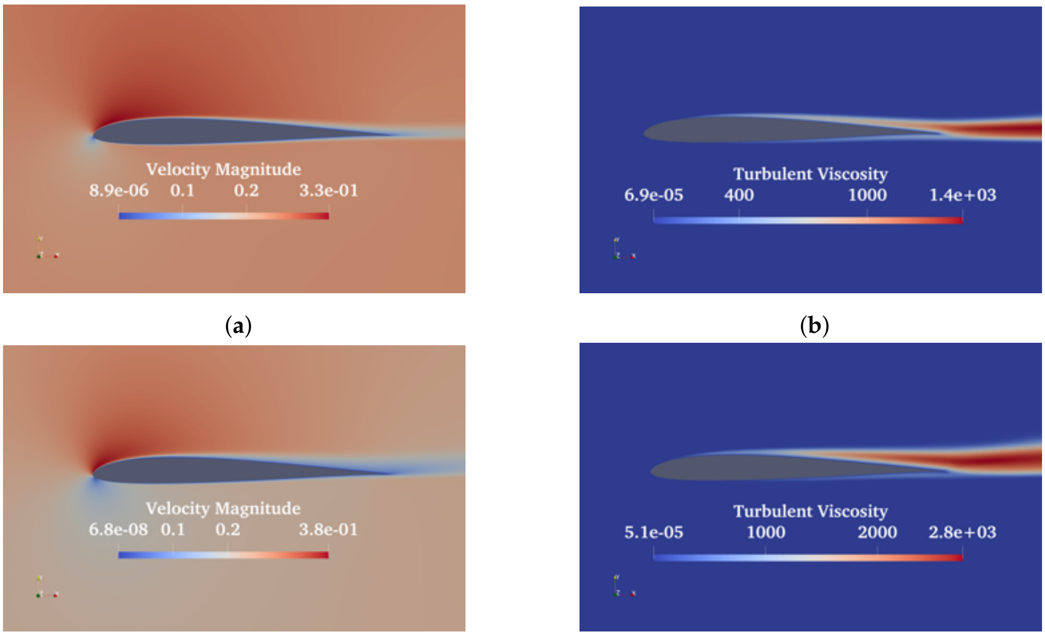
Figure 13 shows the results of using the FFNN implemented in the CFD solver. The simulation starts from the converged solution of the original SA model. To better understand the results, Table 4 presents the results in terms of CL and CD. It is evident that for $\alpha = 8°$, we had a good match in terms of CL with the optimized result of the field inversion. This is not the case for $\alpha = 4°$, where it is clear that the trained model did not learn the correct relation to predict the correct value of CL, which remains the same as the original SA model.

By examining these results, it becomes apparent that while the FFNN showed promising results for certain angles of attack, it did not consistently achieve the desired accuracy across different conditions. This suggests that the model may require further tuning or a different approach to improve its generalization and prediction capabilities.

We left an angle of attack $\alpha = 6°$ out of the training set and did not perform field inversion for this angle. Instead, we used it as a test to understand how well the trained model can predict an angle not included in the training set. Figure 14 shows the results in terms of velocity and turbulent viscosity fields. Table 5 provides a comparison of the global coefficients between high-fidelity, original SA, and augmented SA models. For this test case, we observed an improvement in CL but not in CD, which was expected since we trained the model with data optimized for CL.

Table 4. Comparison of lift and drag coefficients for SD7003 airfoil.

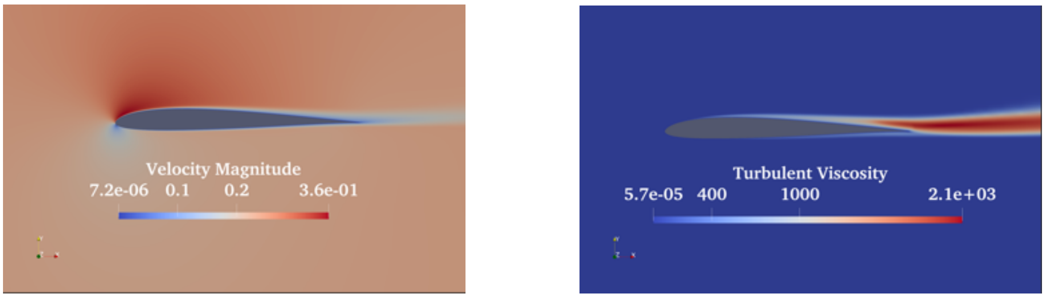
		CL	**CD**
$\alpha = 4°$	LES	0.59	0.021
	SA	0.551	0.0099
	SA (FI−CL)	0.586	0.0147
	SA (FFNN−CL)	0.550	0.0103
$\alpha = 8°$	LES	0.92	0.043
	SA	0.895	0.0435
	SA (FI−CL)	0.917	0.0436
	SA (FFNN−CL)	0.917	0.0435



(**a**) (**b**)

(**c**) (**d**)

Figure 13. Results of the FFNN-augmented Spalart–Allmaras model for the SD7003 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 4^\circ$. (**c**) Velocity magnitude and (**d**) turbulent viscosity at $\alpha = 8^\circ$.



(**a**) (**b**)

Figure 14. Results of the FFNN-augmented Spalart–Allmaras model for the SD7003 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 6^\circ$.

Finally, these results indicate that the classical FFNN approach is not sufficiently robust to capture all corrections for the angles of attack used in the training. However, it performed surprisingly well for an angle of attack outside the training set. It is worth noting that the error in terms of CL for the SD7003 airfoil was already low in the original SA model. This test case served as our initial exploration of the model's capabilities, and we were limited by a small dataset with only two angles of attack. In Section 3.3, we explore another test case with more experimental results at different angles of attack, providing a more comprehensive evaluation of the model's performance.

Table 5. Comparisonof lift and drag coefficients for SD7003 airfoil at $\alpha = 6^\circ$.

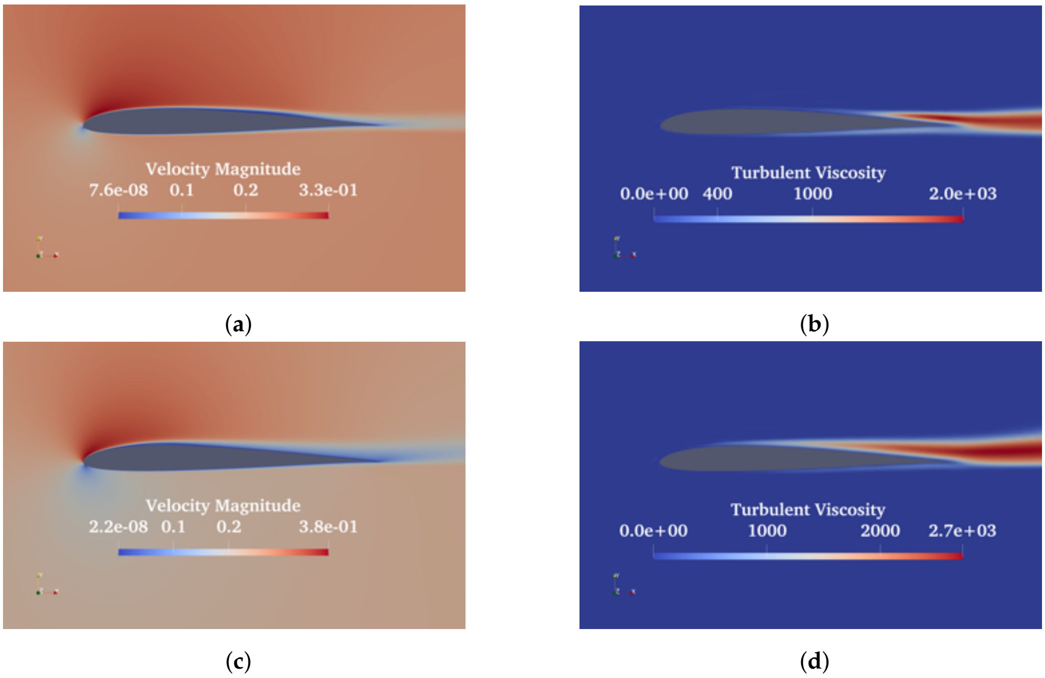
	CL	CD
LES [54]	0.748	0.0319
SA	0.735	0.023
SA (FFNN−CL)	0.746	0.0237

3.2.3. U-Net Model Training

In this section, we present the results obtained using an alternative approach, as explained in Section 2.4. We trained a U-Net model using the data from the two angles of attack for which field inversion was performed. The inputs to the U-Net model are the converged results of the original model, including velocity, pressure, temperature, and turbulent viscosity fields. The U-Net model then generates a turbulent viscosity field, which is compared with the corrected field obtained through field inversion optimization.

In this approach, the U-Net model leverages the converged results of the original SA model to determine the correct turbulent viscosity field, which is then imposed in the solution. After imposing this field in the solution, we allow the other fields to evolve to achieve an improved steady solution.

In Figure 15, (b) and (d) show the corrected turbulent viscosity obtained from the U-Net model, while (a) and (c) represent the velocity magnitude after imposing this corrected turbulent viscosity in the RANS equations. The turbulent viscosity from the U-Net model is in agreement with that obtained from field inversion, as shown in Figures 8a and 10a.



(**a**) (**b**) (**c**) (**d**)

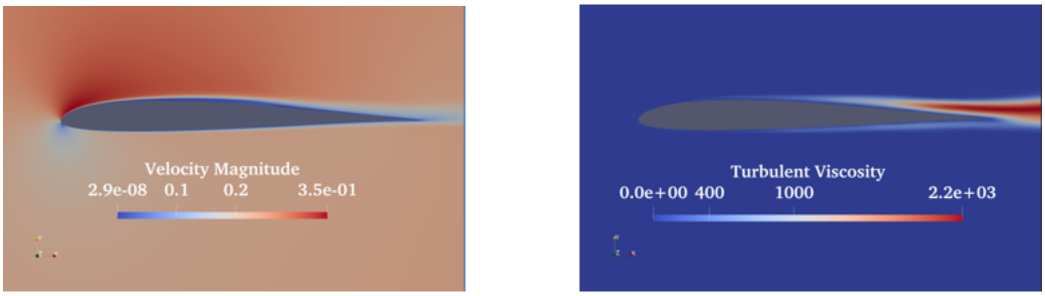
Figure 15. Results of the U-Net-augmented Spalart–Allmaras model for the SD7003 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 4°$. (**c**) Velocity magnitude and (**d**) turbulent viscosity at $\alpha = 8°$.

Table 6 provides a detailed comparison of the results in terms of CL and CD. Since the data were optimized to match CL, the CL values obtained in this simulation align closely with those from the optimization. For $\alpha = 4°$, the CD value is consistent with the optimized value but still differs from the experimental data. For $\alpha = 8°$, the CD is higher compared to the experimental data. This discrepancy is expected, as the turbulent viscosity field was specifically optimized to match CL. Nevertheless, in the case of $\alpha = 4°$, the CD shows an improvement compared to the original model.

We use the $\alpha = 6°$ case to test the performance of the U-Net model outside the training data. In Figure 16, we show the velocity and turbulent viscosity fields. From the velocity field, it is evident that the separation moves towards the leading edge compared to the $\alpha = 4°$ case. More detailed results are shown in Table 7. The results from the U-Net model still show a discrepancy in CL similar to the original SA model, but this time CL is overestimated. Notably, the CD value in this case is close to the LES value.

Table 6. Comparison of lift and drag coefficients for SD7003 airfoil.

		CL	CD
$\alpha = 4°$	LES [54]	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 [54]	0.92	0.043
	SA	0.895	0.0435
	SA (FI−CL)	0.917	0.0436
	SA (U-Net−CL)	0.917	0.0494



(**a**) (**b**)

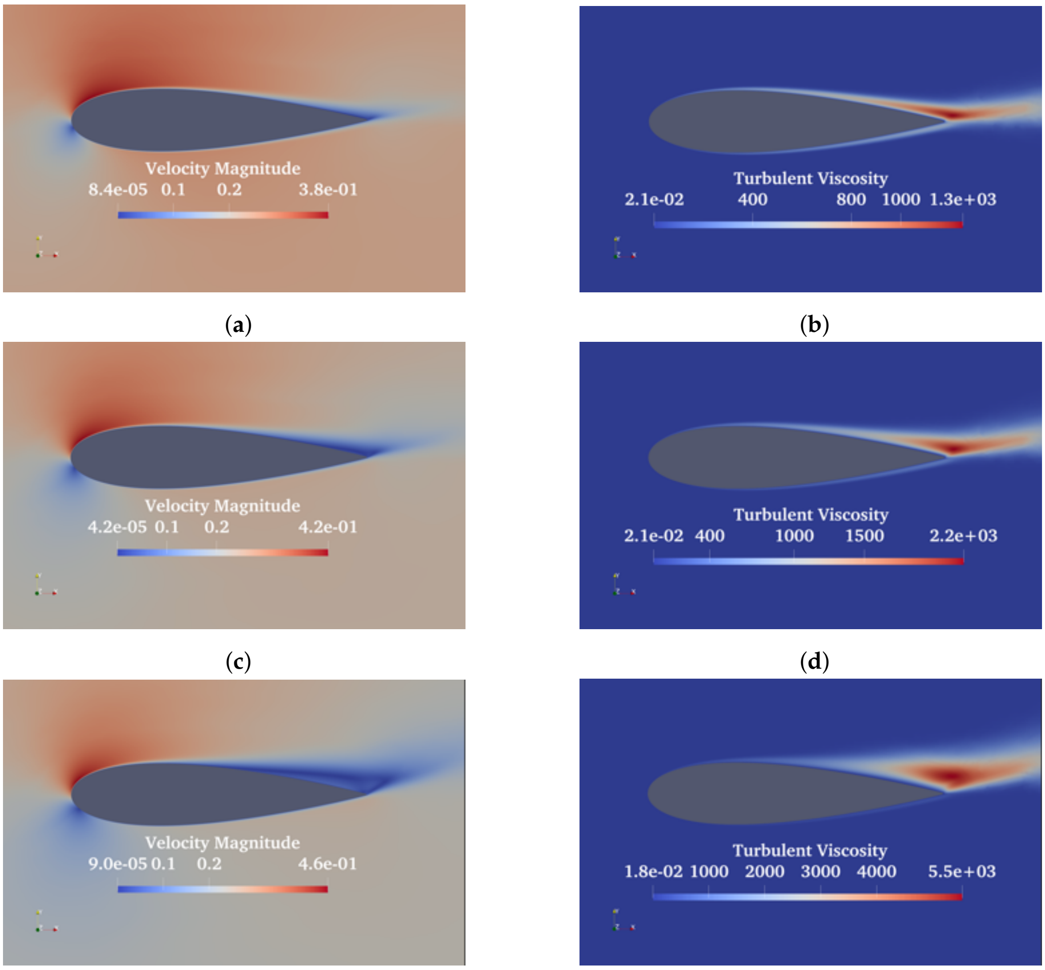
Figure 16. Results of the U-Net-augmented Spalart–Allmaras model for the SD7003 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 6°$.

Table 7. Comparison of lift and drag coefficients for SD7003 airfoil at $\alpha = 6°$.

	CL	CD
LES [54]	0.748	0.0319
SA	0.735	0.023
SA (U-Net−CL)	0.759	0.033

3.3. Flow around the NACA0021 Airfoil

We also evaluated the FIML approach using the NACA0021 airfoil. This airfoil is a notable test case frequently used to assess the performance of transitional RANS models. For this analysis, we referenced the work of Choudhry et al. [53]. The mentioned study investigated the flow around a two-dimensional NACA0021 airfoil at low Reynolds numbers and turbulence intensities to examine long separation bubbles and their impact on performance. Two transition models were evaluated: the correlation-based $\gamma - Re_\theta$ intermittency model and laminar–kinetic-energy-based $k - k_L - \omega$ model. The authors compared experimental and computed lift and drag coefficients for the NACA0021 airfoil at a Reynolds number of 120,000. The $\gamma - Re_\theta$ model under-predicted both lift and drag coefficients, with increased divergence at higher angles. It also predicted a gradual stall behavior typical of thick airfoils at high Reynolds numbers. Conversely, the $k - k_L - \omega$ model showed better agreement with experimental lift coefficients in the attached-flow regime and accurately predicted drag coefficients and stall behavior, capturing the abrupt stall due to the separation bubble bursting at low Reynolds numbers. However, a deviation from the experimental lift coefficient is still present at a higher angle of attack. In this work, our objective is to understand the ability of the FIML paradigm to improve the SA model in order to predict flow separation and achieve good results in terms of lift and drag coefficients at higher angles of attack. For this test case, we had experimental CL and CD for $\alpha \in [0°, 20°]$. We used six angles of attack, specifically $\alpha = 8°, 10°, 12°, 14°, 16°$ and $20°$. In Figure 17, we show the velocity and turbulent viscosity fields of the original SA model for $\alpha = 8°, 12°, \text{and} 16°$.



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 17. Results of the Spalart–Allmaras model for the NACA0021 test case. (**a**) Velocity magnitude and (**b**) turbulent viscosity at $\alpha = 8^\circ$. (**c**) Velocity magnitude and (**d**) turbulent viscosity at $\alpha = 12^\circ$. (**e**) Velocity magnitude and (**f**) turbulent viscosity at $\alpha = 16^\circ$.

The velocity field shows a growing separation starting from the trailing edge of the airfoil, which leads to an increase in turbulent viscosity in this region. However, this is insufficient for predicting accurate CL and CD values.

In Figure 18, we show the results of running the original SA model for various angles of attack, covering both pre-stall and post-stall conditions. The CL behavior indicates that as the angle of attack increases, the original model failed to accurately capture the stall and subsequent separation.

We begin by performing an optimization based on the experimental values of CL. In Figure 19, we present the velocity and turbulent viscosity fields after the optimization has been completed for angles of attack $\alpha = 8^\circ, 12^\circ, \text{and} 16^\circ$. Notably, there is a significant difference, particularly for $\alpha = 16^\circ$. At this angle of attack, a marked separation is evident in the velocity field, and the shape of the turbulent viscosity distribution is also substantially altered.

Figure 20 illustrates the comparison between experimental data, the original SA model, and the augmented SA model. The augmented model incorporates the correction field obtained after the optimization using field inversion. An improvement in CL is observed with the augmented SA model compared to the original SA model. This improvement is noticeable in the pre-stall region but becomes significantly more pronounced in the post-stall region, where flow separation is prominent.

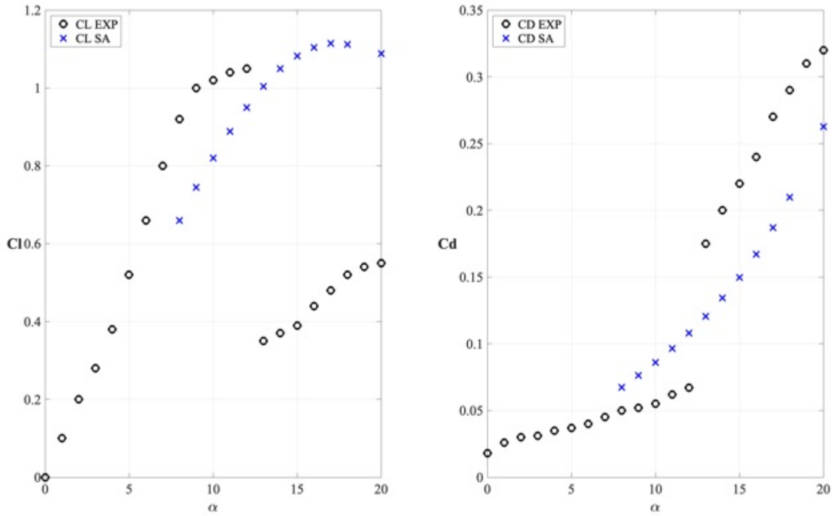
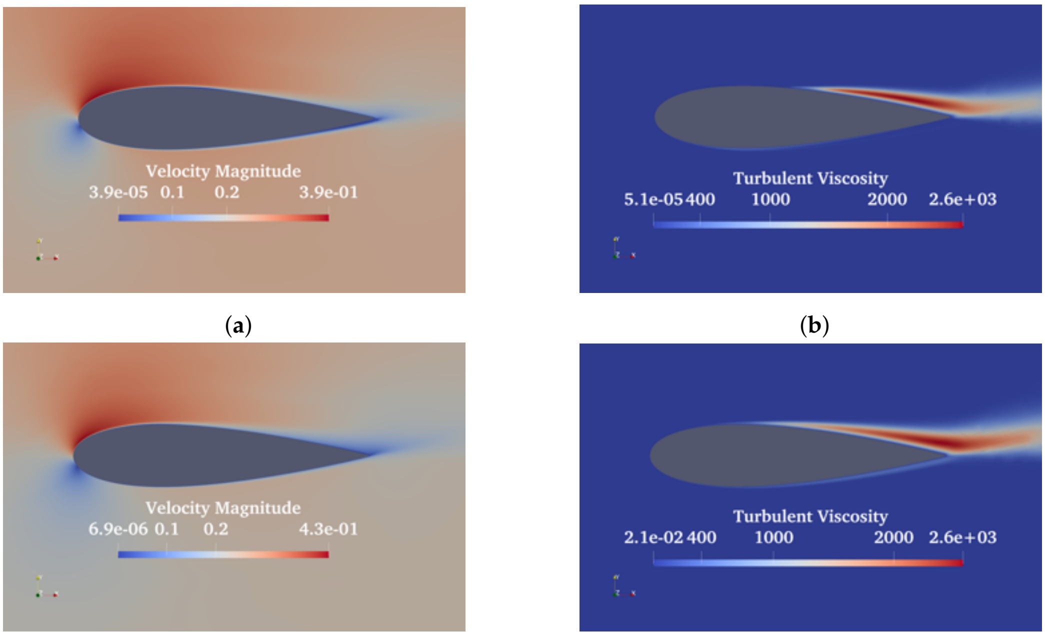


Figure 18. Lift and drag coefficients predicted for the NACA0021 test case by the Spalart–Allmaras model at various angles of attack.

We also perform the optimization using CD as the goal function. Below we present the results of this optimization with CD as the target.

In Figure 21, we observe significant differences in both the velocity and turbulent viscosity fields compared to the original model, as well as the results from the optimization using CL. The shape of the turbulent viscosity field is completely different from that obtained with the CL optimization. This indicates that using different objective functions leads to distinct correction fields. Ideally, it would be desirable for different objective functions to yield the same results, but we have already observed this behavior in the SD7003 test case.

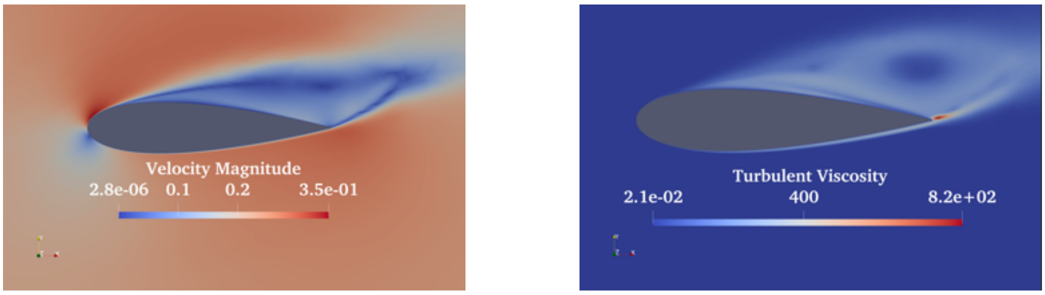
In Figure 22, we present the results in terms of CL and CD. As expected, the CD values match exactly with the experimental data. However, the CL values degrade, even when compared to the original SA model.



(**a**) (**b**)

(**c**) (**d**)

Figure 19. *Cont.*



(**e**) (**f**)

Figure 19. Results of field inversion optimization for CL 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 $\alpha = 16°$.

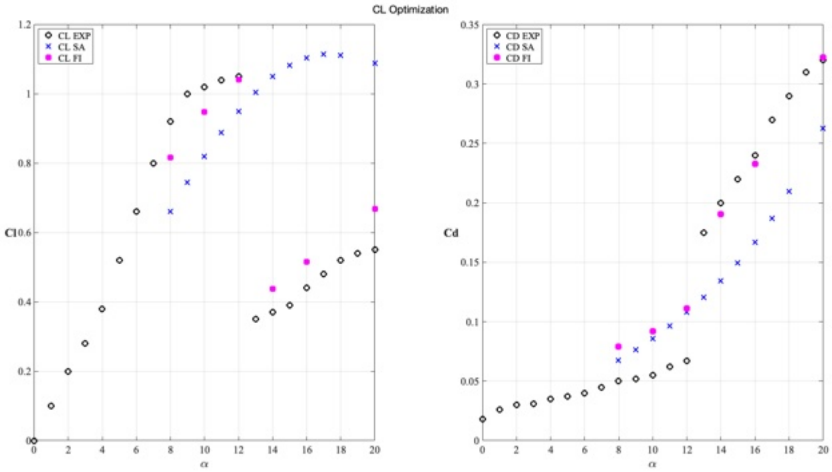
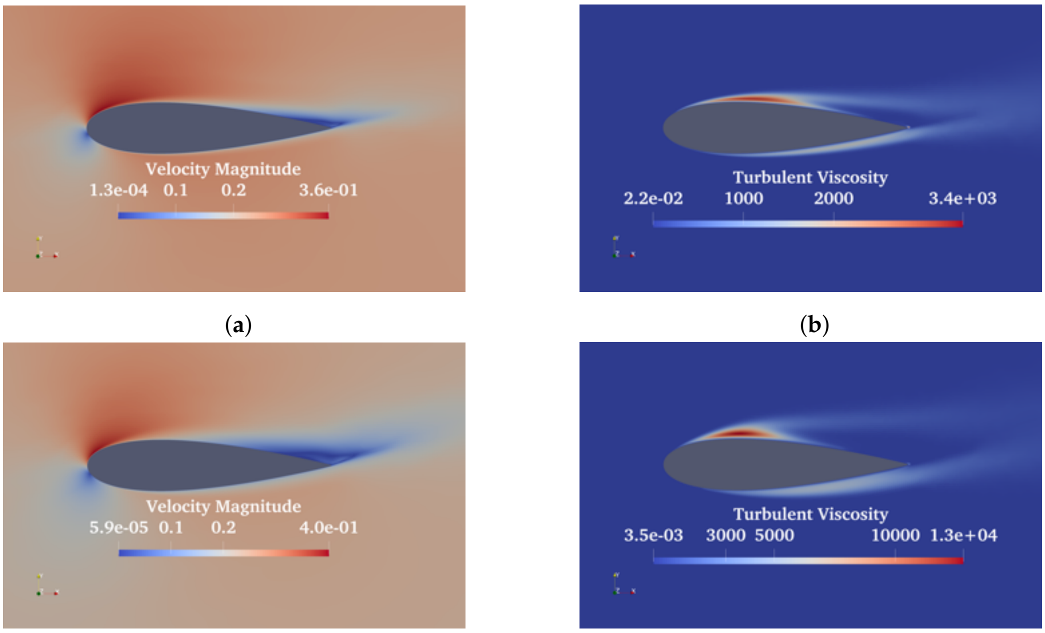


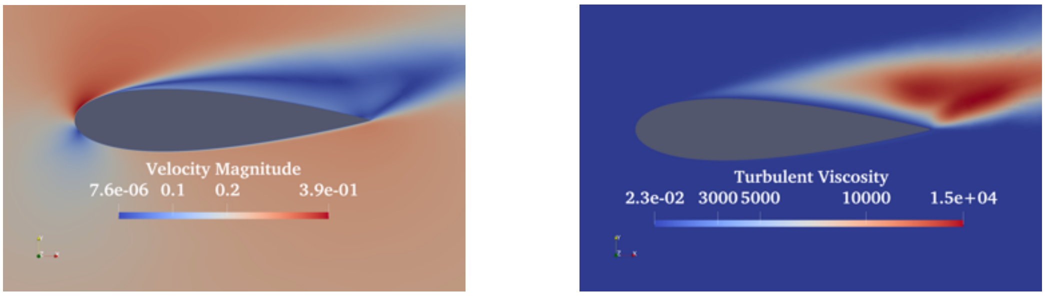
Figure 20. Lift and drag coefficients predicted for the NACA0021 test case after field inversion optimization for CL at various angles of attack.



(**a**) (**b**)

(**c**) (**d**)

Figure 21. *Cont*.



(**e**) (**f**)

Figure 21. Results of field inversion optimization for CD 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 $\alpha = 16°$.

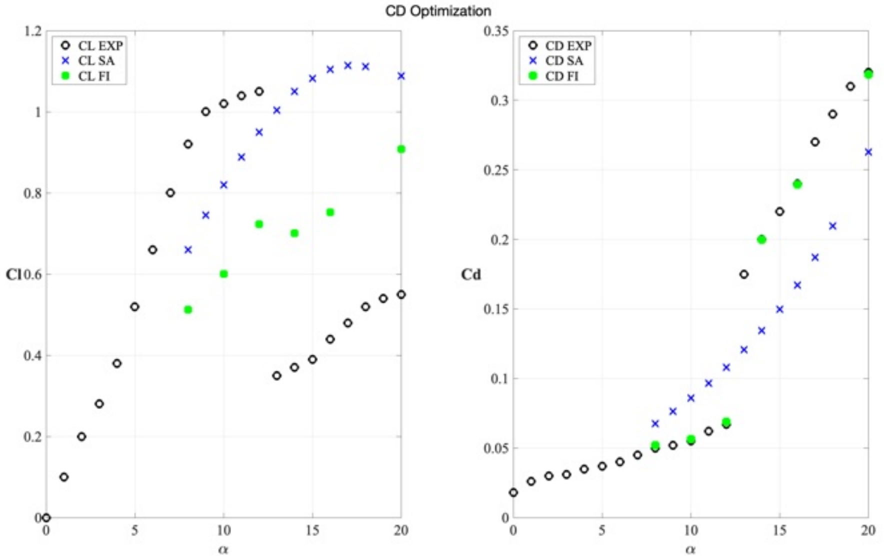


Figure 22. Lift and drag coefficients predicted for NACA0021 test case after field inversion optimization for CD at various angles of attack.

3.3.1. Local Model Training and Testing

As in the SD7003 test case, we also use data optimized for the CL in the NACA0021 test case for testing the classical FIML paradigm. In this section, we present the results obtained by using the FFNN. The architecture and input variables are consistent with those used for the SD7003 test case. As shown previously, we have more angles of attack available for training and additional angles of attack to test the model outside the training set.

Figure 23 illustrates the train and validation loss function through the epochs.

By including more angles of attack in the training set, we aim to improve the robustness and generalizability of the model. This expanded dataset should allow the FFNN better to capture the complex aerodynamic behaviors across different flow conditions.

For this test case, we did not apply any mapping function to the correction field. Avoiding negative values of β reduces the ability to capture the experimental results effectively during optimization. Unlike the SD7003 test case, the trained model did not perform well in prediction, despite achieving a high R^2 value of 0.758. This high value led us to expect good results at least at the training points.

As mentioned in Section 2.3.1, the model might find a relationship that does not correctly represent the physical behavior, potentially introducing instability during numerical transients. This issue was evident in this case, even when starting from a converged solution of the original SA model. Although reducing the CFL number can mitigate instability, the solver could not find a new converged solution that accurately predicts the corrected turbulent viscosity field.

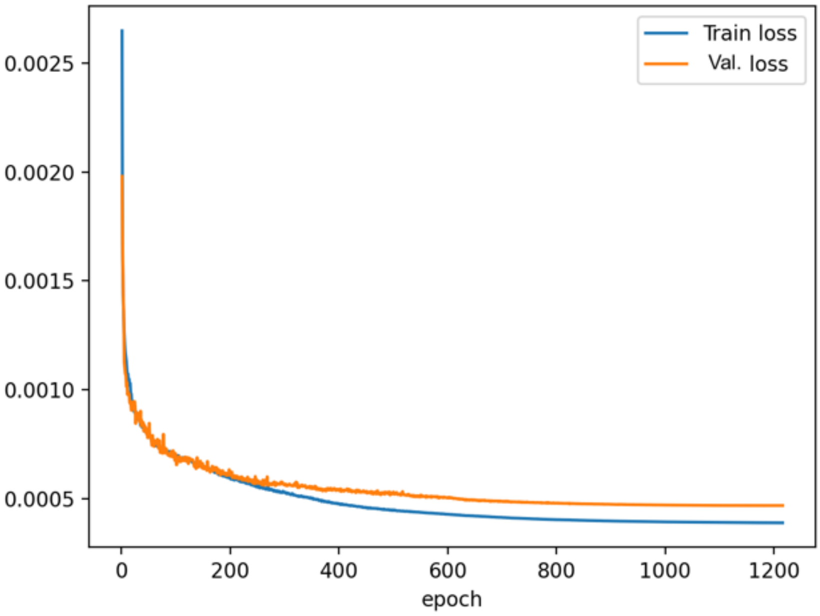


Figure 23. Training and validation loss for the FFNN model trained on NACA0021 data.

Figure 24 presents the results obtained at different angles of attack using the FFNN. There is an improvement in CL at higher angles of attack after incipient stall ($\alpha = 12°$), but a significant discrepancy still exists with the experimental results. We did not observe any improvement for angles of attack before stall, and at $\alpha = 12°$, the model performs worse than the original SA model. We tested two additional angles of attack, one before the incipient stall ($\alpha = 11°$) and one in the post-stall region ($\alpha = 17°$). For these angles, which were not part of the training set, we observed the same behavior.

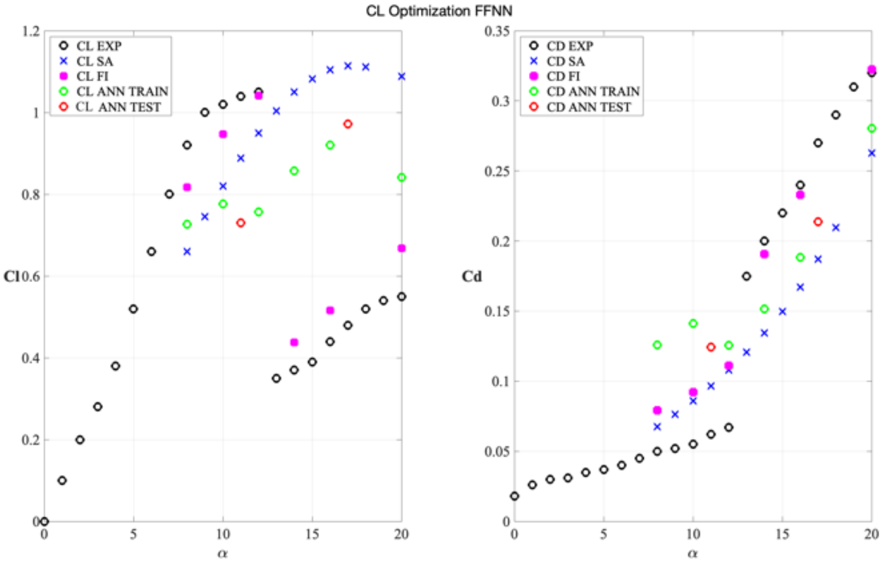


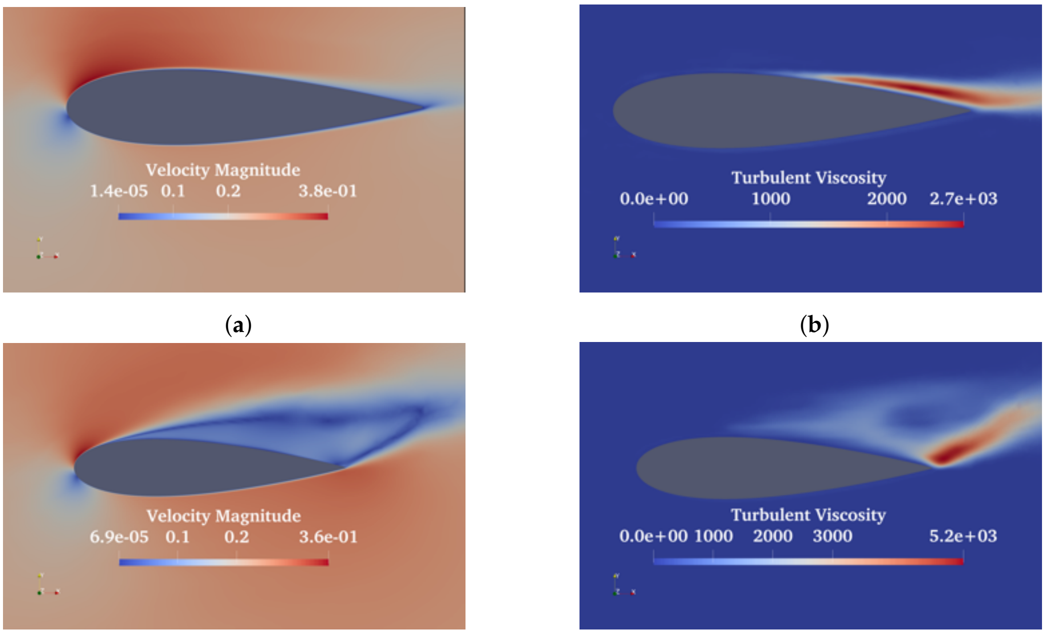
Figure 24. Lift and drag coefficients predicted with FFNN-augmented Spalart–Allmaras model for the NACA0021 test case at various angles of attack.

3.3.2. U-Net Model Training and Testing

In this section, we present the results obtained using the U-Net model. As previously explained, this model helps us to avoid the instability of the classical FFNN approach. In our initial approach, we attempted to find the correlation of the correction field with some local flow variables. This correlation might not exist, leading to instability problems during the numerical transient, even if we started from the converged SA original model. By using the U-Net model, we leverage the converged results of the original SA model to find the correct turbulent viscosity field, which we then impose as the solution. We allow the other fields to evolve to find a new steady solution while the eddy viscosity field remains frozen as predicted by the U-Net model. Below, we show the results for $\alpha = 8°$ and $\alpha = 16°$.

From Figure 25, the results are in agreement with those obtained using field inversion, as shown in Figure 19. Despite some differences in the turbulent viscosity field at $\alpha = 16°$, we can clearly see that the separation in the velocity field is closer to the field inversion results than to the original SA model.

To demonstrate the improvement gained with the U-Net model, Figure 26 compares the results in terms of CL and CD among the experimental data, the original model, the field inversion augmented model, and the U-Net model. Additionally, we tested the model for angles of attack not included in the training process: $\alpha = 9°, 11°, 13°, 15°, 17°$ and $18°$. We did not observe significant improvements in the pre-stall region, but we did observe substantial improvements in the post-stall region, even for the untrained angles of attack. Thus, we can conclude that the new augmented model has good predictive capabilities even in the presence of transition and separation.



(**a**) (**b**)

(**c**) (**d**)

Figure 25. Results of the U-Net-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 $\alpha = 16°$.

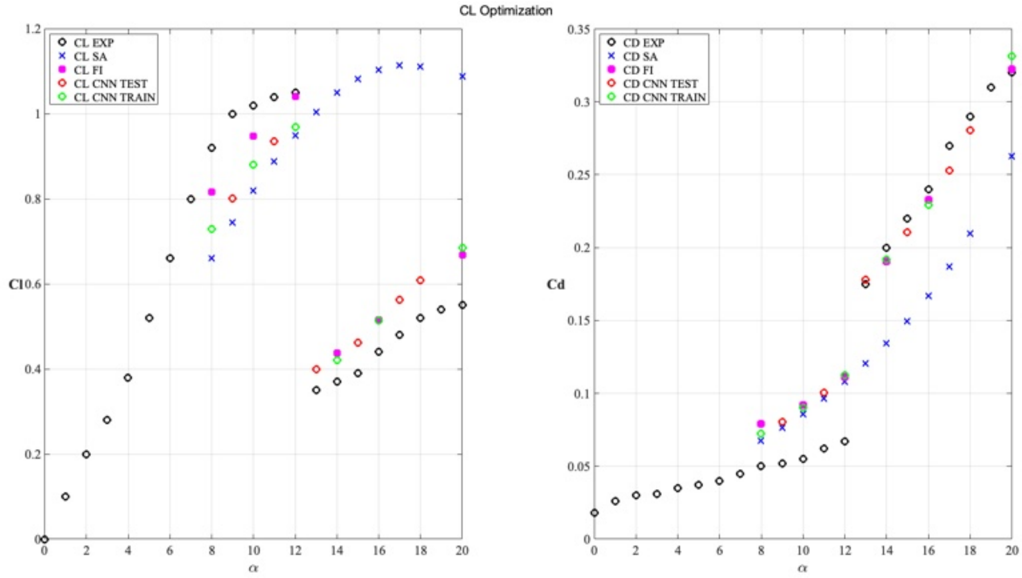


Figure 26. Lift and drag coefficients predicted with U-Net-augmented Spalart–Allmaras model for the NACA0021 test case at various angles of attack.

4. Conclusions

In this work, the potential of field inversion for improving the prediction of the flow field around airfoils at low Reynolds number was investigated and critically discussed. First of all, a discussion on the choice of the goal function which drives field inversion was performed: the minimization of the error on a physical distribution (for example wall static pressure) provided better results with respect to the error on an integral quantity.

Two different strategies for exploiting the results of field inversion are then discussed. The first considered approach is the classical local FIML paradigm which was originally proposed by [13]. This approach is promising but the results obtained on the considered test cases enlightened some criticalities. First of all, the regression step of the FIML paradigm requires us to assume a list of input variables for the FFNN: this choice is not trivial and it is possible to find problems where the correction is not a function of the chosen input variables because it is possible to find in the database different values of the correction for the same values of the input variables. Furthermore, even if the chosen set of input variables provides a suitable regression, this function can lead to numerical instabilities when integrated into a RANS solver and used for predictive CFD simulations. This behavior can be explained by the fact that the FFNN is trained on a set of steady solutions and so, during the numerical transient of a predictive simulation, it can encounter input values that are far from the training range. In the present work, it was observed that a significant source of instability is related to the development of negative corrections during the transients: in order to solve this problem, a mapping function on the correction is introduced during the field inversion optimization in order to guarantee the positivity of the correction. This significantly improved the numerical behavior of the corrected model during the predictions but it limited its correction capability.

This last limitation was not observed in the global U-Net approach. This approach has already demonstrated promising results. For instance, Wang et al. in [55] utilized a U-Net model as a surrogate, referring to it as a physics-informed CNN (PI-CNN) because they enhanced the loss function with physical constraints. They applied this model to predict the turbulent mixing layer over time at various Reynolds numbers. In contrast, our work employs a different strategy. Rather than providing an augmented RANS model like the FIML paradigm, our method offers a post-processing procedure that globally adjusts the flow field predicted by the baseline RANS model. Since the corrected eddy viscosity is directly imposed as a frozen field in the CFD simulation, the numerical instabilities, which represent a problem for FIML, are not observed in the global U-Net approach. In the present implementation, the U-Net correction acts on a Cartesian grid and so it is necessary to interpolate the CFD solution provided by the baseline RANS model from the original unstructured mesh to a Cartesian mesh and back. Interpolating data onto a Cartesian mesh for U-Net can lead to significant errors if the mesh is too coarse, particularly in the boundary layer region. This issue is also highlighted in [56], where the authors opted to use a simple FFNN to circumvent the problem. In our study, we integrated the FFNN within the CFD software, but this introduced instability, unlike in their work where the FFNN served as a surrogate model to predict flow quantities based on airfoil shape, Mach number, and angle of attack. The interpolation issue can be mitigated by using a sufficiently fine Cartesian mesh, limited to the region of interest (typically near the solid wall where the goal function is computed), or by employing U-Nets that operate on unstructured meshes. This latter approach aligns with the method proposed by [57], in which the authors used a graph convolutional network (GCN), a variant of the convolutional neural network, as described in [58]. This extension is left for future work.

In this study, we evaluated the proposed methodology only in 2D problems because in this way it was possible to perform several tests due to the low cost of the 2D simulations. However, there are no theoretical limitations to the extension of the method in 3D since it would be sufficient to modify the architecture of the U-Net model.

The two strategies were investigated on two test cases characterized by transition and separation (NACA0021 and SD7003 airfoils): these flow features are not correctly captured by the chosen baseline Spalart–Allmaras model which was developed for fully turbulent flows. However, the field inversion shows its potential by providing a correction field which can significantly improve the prediction in both cases. The correction field was then exploited with both the local FIML and the global U-Net approaches: both strategies are able to improve the prediction of the original RANS model but the FIML showed some instability problems which limited the exploitation of field inversion results. On the contrary, the global U-Net approach avoided these instability problems and provided significant improvements in the predicted flow field. Furthermore, the global U-Net approach is less intrusive since does not require us to implement the correction to the production terms in the turbulent eddy viscosity equation inside the CFD solver.

Author Contributions: Conceptualization, L.M., A.F. and F.L.; methodology, L.M. and A.F.; software, L.M., M.C. and A.I.; supervision, A.F. and F.L.; visualization, L.M.; writing—original draft preparation, L.M.; writing—review and editing, L.M. and A.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Italian Ministry of University and Research through the PRIN research project 2022B2X937, 'NextGenSProDesT: Next Generation Space Propulsion Design Techniques', and supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions, grant agreement 872442 (ARIA).

Data Availability Statement: The data presented in this study are available upon request from the corresponding author. The full dataset is not publicly available due to the extensive number of simulations performed for this work; however, specific data will be provided upon request.

Acknowledgments: The authors would like to thank OPTIMAD Srl for providing the CFD software, immerFlow, and the necessary computational resources that made all the simulations possible. Additionally, we extend our gratitude to NUREA for their assistance in understanding the U-Net model applied in this work. Special thanks to Laura Balasso for sharing her extensive knowledge of machine learning models and for all her valuable suggestions. We also wish to thank Sandro Malusà and Kostantinos Samouchos for their insightful discussions on the various applications of machine learning models in the field of CFD.

Conflicts of Interest: Authors Marco Cisternino and Andrea Iob were employed by the company Optimad Srl Via Agostino da Montefeltro. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zhu, H.; Jiang, Z.; Zhao, H.; Pei, S.; Li, H.; Lan, Y. Aerodynamic performance of propellers for multirotor unmanned aerial vehicles: Measurement, analysis, and experiment. *Shock Vib.* **2021**, *2021*, 9538647. [CrossRef]
2. Zhang, W.; Xu, B.; Zhang, H.; Xiang, C.; Fan, W.; Zhao, Z. Analysis of aerodynamic characteristics of propeller systems based on Martian atmospheric environment. *Drones* **2023**, *7*, 397. [CrossRef]
3. Winslow, J.; Otsuka, H.; Govindarajan, B.; Chopra, I. Basic understanding of airfoil characteristics at low Reynolds numbers (10^4–10^5). *J. Aircr.* **2018**, *55*, 1050–1061. [CrossRef]
4. Grava, A.; Picillo, M.; Serpieri, J.; Iuso, G.; Bernardos, L.; Cafiero, G. Aerodynamic investigation of a drone propeller in cross-flow. In Proceedings of the AIAA SCITECH 2024 Forum, Orlando, FL, USA, 8–12 January 2024; p. 0244.
5. Yao, Y.; Ma, D.; Zhang, L.; Yang, X.; Yu, Y. Aerodynamic optimization and analysis of low Reynolds number propeller with gurney flap for ultra-high-altitude unmanned aerial Vehicle. *Appl. Sci.* **2022**, *12*, 3195. [CrossRef]
6. Liu, X.; Zhao, D.; Oo, N.L. Comparison studies on aerodynamic performances of a rotating propeller for small-size UAVs. *Aerosp. Sci. Technol.* **2023**, *133*, 108148. [CrossRef]
7. Michálek, J.; Monaldi, M.; Arts, T. 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. *J. Turbomach.* **2012**, *134*, 061009. [CrossRef]
8. Ferrero, A.; Iollo, A.; Larocca, F. Field inversion for data-augmented RANS modelling in turbomachinery flows. *Comput. Fluids* **2020**, *201*, 104474. [CrossRef]
9. Koca, K.; Genç, M.S.; Açıkel, H.H.; Çağdaş, M.; Bodur, T.M. Identification of flow phenomena over NACA 4412 wind turbine airfoil at low Reynolds numbers and role of laminar separation bubble on flow evolution. *Energy* **2018**, *144*, 750–764. [CrossRef]
10. Chawla, J.S.; Suryanarayanan, S.; Puranik, B.; Sheridan, J.; Falzon, B.G. Efficiency improvement study for small wind turbines through flow control. *Sustain. Energy Technol. Assess.* **2014**, *7*, 195–208. [CrossRef]
11. Bush, R.H.; Chyczewski, T.S.; Duraisamy, K.; Eisfeld, B.; Rumsey, C.L.; Smith, B.R. Recommendations for future efforts in RANS modeling and simulation. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019; p. 0317.
12. Duraisamy, K.; Iaccarino, G.; Xiao, H. Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **2019**, *51*, 357–377. [CrossRef]
13. Parish, E.J.; Duraisamy, K. A paradigm for data-driven predictive modeling using field inversion and machine learning. *J. Comput. Phys.* **2016**, *305*, 758–774. [CrossRef]
14. Singh, A.P.; Duraisamy, K.; Zhang, Z.J. Augmentation of turbulence models using field inversion and machine learning. In Proceedings of the 55th AIAA Aerospace Sciences Meeting, Grapevine, TX, USA, 7–13 January 2017; p. 0993.
15. Ling, J.; Kurzawski, A.; Templeton, J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **2016**, *807*, 155–166. [CrossRef]
16. Weatheritt, J.; Sandberg, R. A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. *J. Comput. Phys.* **2016**, *325*, 22–37. [CrossRef]
17. Reissmann, M.; Hasslberger, J.; Sandberg, R.D.; Klein, M. Application of gene expression programming to a posteriori LES modeling of a Taylor Green vortex. *J. Comput. Phys.* **2021**, *424*, 109859. [CrossRef]
18. Shin, J.; Hansinger, M.; Pfitzner, M.; Klein, M. A priori analysis on deep learning of filtered reaction rate. *Flow Turbul. Combust.* **2022**, *109*, 383–409. [CrossRef]
19. Di Fabbio, T.; Rajkumar, K.; Tangermann, E.; Klein, M. Towards the understanding of vortex breakdown for improved RANS turbulence modeling. *Aerosp. Sci. Technol.* **2024**, *146*, 108973. [CrossRef]
20. Duraisamy, K.; Zhang, Z.J.; Singh, A.P. New approaches in turbulence and transition modeling using data-driven techniques. In Proceedings of the 53rd AIAA Aerospace Sciences Meeting, Kissimmee, FL, USA, 5–9 January 2015; p. 1284.
21. Ferrero, A.; Iollo, A.; Larocca, F.; Loffredo, M.; Menegatti, E. Field inversion and machine learning strategies for improving rans modelling in turbomachinery. In Proceedings of the ETC 2021-14th European Turbomachinery Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Gdansk, Poland, 12–16 April 2021.
22. Srivastava, V.; Duraisamy, K. Generalizable physics-constrained modeling using learning and inference assisted by feature-space engineering. *Phys. Rev. Fluids* **2021**, *6*, 124602. [CrossRef]
23. You, Y.; Yu, F.; Mao, N. Fast Prediction and Optimization of Building Wind Environment Using CFD and Deep Learning Method. *Appl. Sci.* **2024**, *14*, 4087. [CrossRef]
24. Yu, Y.; Chen, S.; Wei, H. Modified UNet with attention gate and dense skip connection for flow field information prediction with porous media. *Flow Meas. Instrum.* **2023**, *89*, 102300. [CrossRef]
25. Miao, X.; Wang, Z.; Ren, S.; Zhang, L.; Li, H.; Dong, L.; Chen, D.; Hu, C. Intelligent mesh refinement based on U-NET for high-fidelity CFD simulation in numerical reactor. *Nucl. Eng. Des.* **2023**, *411*, 112411. [CrossRef]
26. Bokil, G.R.; Geyer, T.F.; Merbold, S.; Kazula, S. Physics-Guided Convolutional Neural Network for Flow Prediction in Heat Exchangers in Electrified Aircraft. In Proceedings of the AIAA AVIATION FORUM AND ASCEND 2024, Las Vegas, NV, USA, 30 July–1 August 2024; p. 4108.
27. Allmaras, S.R.; Johnson, F.T. Modifications and clarifications for the implementation of the Spalart–Allmaras turbulence model. In Proceedings of the Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, HI, USA, 9–13 July 2012; Volume 1902, pp. 1–11.

28. Spalart, P.; Allmaras, S. A one-equation turbulence model for aerodynamic flows. In Proceedings of the 30th Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 6–9 January 1992; p. 439.
29. Brunton, S.L.; Noack, B.R.; Koumoutsakos, P. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **2020**, *52*, 477–508. [CrossRef]
30. Garnier, P.; Viquerat, J.; Rabault, J.; Larcher, A.; Kuhnle, A.; Hachem, E. A review on deep reinforcement learning for fluid mechanics. *Comput. Fluids* **2021**, *225*, 104973. [CrossRef]
31. Viquerat, J.; Meliga, P.; Larcher, A.; Hachem, E. A review on deep reinforcement learning for fluid mechanics: An update. *Phys. Fluids* **2022**, *34*, 111301. [CrossRef]
32. Beck, A.; Kurz, M. A perspective on machine learning methods in turbulence modeling. *GAMM-Mitteilungen* **2021**, *44*, e202100002. [CrossRef]
33. Pope, S.B. A more general effective-viscosity hypothesis. *J. Fluid Mech.* **1975**, *72*, 331–340. [CrossRef]
34. Jiang, C.; Mi, J.; Laima, S.; Li, H. A novel algebraic stress model with machine-learning-assisted parameterization. *Energies* **2020**, *13*, 258. [CrossRef]
35. Zhao, Y.; Akolekar, H.D.; Weatheritt, J.; Michelassi, V.; Sandberg, R.D. RANS turbulence model development using CFD-driven machine learning. *J. Comput. Phys.* **2020**, *411*, 109413. [CrossRef]
36. Wang, J.X.; Wu, J.L.; Xiao, H. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Phys. Rev. Fluids* **2017**, *2*, 034603. [CrossRef]
37. Bhushan, S.; Burgreen, G.W.; Brewer, W.; Dettwiller, I.D. Development and validation of a machine learned turbulence model. *Energies* **2021**, *14*, 1465. [CrossRef]
38. Meng, Q.; Jiang, Z.; Wang, J. Artificial neural network-based subgrid-scale models for LES of compressible turbulent channel flow. *Theor. Appl. Mech. Lett.* **2023**, *13*, 100399. [CrossRef]
39. Kurz, M.; Beck, A. A machine learning framework for LES closure terms. *arXiv* **2020**, arXiv:2010.03030.
40. Morton, J.; Kochenderfer, M.J.; Witherden, F.D. Parameter-conditioned sequential generative modeling of fluid flows. *AIAA J.* **2021**, *59*, 825–841. [CrossRef]
41. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
42. Eivazi, H.; Tahani, M.; Schlatter, P.; Vinuesa, R. Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations. *Phys. Fluids* **2022**, *34*, 075117. [CrossRef]
43. Jin, X.; Cai, S.; Li, H.; Karniadakis, G.E. NSFnets (Navier–Stokes flow nets): Physics-informed neural networks for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **2021**, *426*, 109951. [CrossRef]
44. Fidkowski, K.J. Gradient-based shape optimization for unsteady turbulent simulations using field inversion and machine learning. *Aerosp. Sci. Technol.* **2022**, *129*, 107843. [CrossRef]
45. Yang, M.; Xiao, Z. Improving the k–ω–γ–Ar transition model by the field inversion and machine learning framework. *Phys. Fluids* **2020**, *32*, 064101. [CrossRef]
46. Brenner, O.; Piroozmand, P.; Jenny, P. Efficient assimilation of sparse data into RANS-based turbulent flow simulations using a discrete adjoint method. *J. Comput. Phys.* **2022**, *471*, 111667. [CrossRef]
47. Nielsen, E.J.; Lu, J.; Park, M.A.; Darmofal, D.L. An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids. *Comput. Fluids* **2004**, *33*, 1131–1155. [CrossRef]
48. Ferrero, A.; Larocca, F.; Pennecchi, F.R. Uncertainty propagation in field inversion for turbulence modelling in turbomachinery. In Proceedings of the 2020 IEEE 7th International Workshop on Metrology for AeroSpace (MetroAeroSpace), Pisa, Italy, 22–24 June 2020; pp. 303–308.
49. Spalart, P.R.; Deck, S.; Shur, M.L.; Squires, K.D.; Strelets, M.K.; Travin, A. A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theor. Comput. Fluid Dyn.* **2006**, *20*, 181–195. [CrossRef]
50. Cakmakcioglu, S.C.; Bas, O.; Mura, R.; Kaynak, U. A revised one-equation transitional model for external aerodynamics. In Proceedings of the AIAA Aviation 2020 Forum, Online, 15–19 June 2020; p. 2706.
51. Thuerey, N.; Weißenow, K.; Prantl, L.; Hu, X. Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows. *AIAA J.* **2020**, *58*, 25–36. [CrossRef]
52. Geuzaine, C.; Remacle, J.F. Gmsh: A three-dimensional finite element mesh generator with built-in pre-and post-processing facilities. *Int. J. Numer. Methods Eng.* **2009**, *79*, 1309–1331. [CrossRef]
53. Choudhry, A.; Arjomandi, M.; Kelso, R. A study of long separation bubble on thick airfoils and its consequent effects. *Int. J. Heat Fluid Flow* **2015**, *52*, 84–96. [CrossRef]
54. Galbraith, M.; Visbal, M. Implicit large eddy simulation of low Reynolds number flow past the SD7003 airfoil. In Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 7–10 January 2008; p. 225.
55. Wang, Y.; Shelyag, S.; Schlüter, J. A Predictive Model for Turbulence Evolution and Mixing Using Machine Learning. *IEEE Access* **2024**, *12*, 115182–115196. [CrossRef]
56. Nemati, M.; Jahangirian, A. A Data-Driven Machine Learning Approach for Turbulent Flow Field Prediction Based on Direct Computational Fluid Dynamics Database. *J. Appl. Fluid Mech.* **2023**, *17*, 60–74.

57. Belbute-Peres, F.D.A.; Economon, T.; Kolter, Z. Combining differentiable PDE solvers and graph neural networks for fluid flow prediction. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 2402–2411.
58. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.