National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Symbolic Regression in Turbulence and Transition Modeling. Applications in Aerodynamic Shape Optimization

Diploma Thesis

Kalliopi Batsi

Supervisors:
Kyriakos C. Giannakoglou, Professor NTUA
Dr. V. Asouti, Adjunct Lecturer NTUA

Athens, 2026

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Kyriakos C. Giannakoglou, for providing me the opportunity to conduct research on such an innovative and challenging topic. His continuous guidance and support throughout the various stages of this work have been fundamental to its completion.

I am also deeply grateful to the members of the Parallel CFD & Optimization (PCOpt) Unit for their collaboration and support. In particular, I would like to thank Dr. Varvara Asouti for her time and continuous mentorship, as well as her help with with any technical issues that arised.

I would also like to thank Angelos, Andreas and Aris for our invaluable cooperation and the time we spent together while working on our research in the PCOpt Unit. Lastly, I am profoundly grateful to my family and the people closest to me for their sincere support throughout my entire academic journey. Their encouragement and presence have been a constant source of motivation, contributing significantly to the successful completion of this work.

National Technical University of Athens
School of Mechanical Engineering
Fluids Department
Parallel CFD & Optimization Unit

Symbolic Regression in Turbulence and Transition Modeling. Applications in Aerodynamic Shape Optimization

Diploma Thesis

Kalliopi Batsi

Supervisors:

Kyriakos C. Giannakoglou, Professor NTUA

Dr. V. Asouti, Adjunct Lecturer NTUA

Athens, July 2026

Abstract

The accurate prediction of turbulent flows and laminar-to-turbulent transition remains one of the most significant challenges in Computational Fluid Dynamics (CFD). While high-fidelity simulations, such as Large Eddy Simulation (LES) or Detached Eddy Simulation (DES), offer superior accuracy, they are accompanied by a prohibitive computational cost. This renders the need for faster RANS closure models essential, which do not sacrifice the calculation's accuracy to a great degree. Traditional transport equation-based turbulence and transition models add partial differential equations to the system, whereas algebraic models are based on determining the eddy viscosity directly from local flow quantities without solving additional transport equations. However, they are based on simplifications, such as Prandtl's "mixing-length" hypothesis, which renders them inaccurate for complex flows.

Data-driven approaches have been proved to be very promising in terms of turbulence and transition surrogate modeling. More specifically, DNN-based surrogates have exhibited great accuracy, although they are very sensitive to overfitting and lack of generalization. Furthermore, their "black-box" nature, leaves no room for interpretation of the surrogate model. Moreover, an analytical differentiation, in view of the model's integration within an optimization process, is not possible.

This diploma thesis explores an innovative data-driven approach by constructing algebraic turbulence and transition surrogate models using Symbolic Regression (SR). By leveraging the PySR algorithm, this work aims to replace computationally expensive transport equations with closed-form algebraic expressions that relate local flow and geometric quantities to target variables, such as the intermittency factor γ and the turbulent eddy viscosity ν_t. This ensures both physical transparency and analytical differentiability, whilst reducing the risk of overfitting, thus eliminating the disadvantages of previous data-driven approaches.

The methodology is validated across diverse cases. For transition modeling, the selected cases include the NLF(1)-0416 airfoil, which was designed by NASA for natural laminar flow, and the ONERA OAT15A airfoil, which is utilized for studying transonic flow and buffeting phenomena. Additionally, the two airfoils are used in combination to derive a single, more generalized algebraic expression for the intermittency field. The SR-based surrogates to the Piotrowski-Zingg variation of the $\gamma - Re_\theta$ model are integrated within a RANS simulation in the PUMA solver for evaluation. In terms of turbulence modeling, the framework is applied to the NACA 0012 airfoil and the Ahmed Body with a 35° slant. These cases serve to evaluate the Eddy Viscosity Adaptation - Symbolic Regression (EVA-SR) framework, aiming to improve the modeling of separation and recirculation regions. The resulting algebraic surrogates are integrated into the OpenFOAM RANS solver to replace the Spalart-Allmaras model, and their performance is subsequently evaluated within a Shape Optimization (ShpO) loop to test the advantages of a dynamically updating eddy viscosity field compared to a frozen field approach.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης

Συμβολική Παλινδρόμηση στη Μοντελοποίηση της Τύρβης και της Μετάβασης στην Αεροδυναμική Βελτιστοποίηση Μορφής

Διπλωματική Εργασία

Καλλιόπη Μπατσή

Επιβλέποντες: Κυριάκος Χ. Γιαννακόγλου, Καθηγητής ΕΜΠ
Δρ. Β. Ασούτη, Εντεταλμένη Διδάσκουσα ΕΜΠ

Αθήνα, Ιούλιος 2026

Περίληψη

Η ακριβής πρόβλεψη των τυρβωδών ροών και της μετάβασης από τη στρωτή στην τυρβώδη ροή παραμένει μία από τις σημαντικότερες προκλήσεις στην Υπολογιστική Ρευστοδυναμική (ΥΡΔ). Αν και οι προσομοιώσεις υψηλής πιστότητας, όπως η Προσομοίωση Μεγάλων Δινών ή η Προσομοίωση Αποκομμένων Δινών, προσφέρουν ανώτερη ακρίβεια, συνοδεύονται από απαγορευτικό υπολογιστικό κόστος. Αυτό καθιστά απαραίτητη την ανάγκη για ταχύτερα μοντέλα, τα οποία δεν θυσιάζουν σε μεγάλο βαθμό την ακρίβεια του υπολογισμού. Τα παραδοσιακά μοντέλα τύρβης και μετάβασης που βασίζονται σε εξισώσεις μεταφοράς προσθέτουν μερικές διαφορικές εξισώσεις στο σύστημα, ενώ τα αλγεβρικά μοντέλα βασίζονται στον προσδιορισμό της τυρβώδους συνεκτικότητας απευθείας από τοπικές ροϊκές ποσότητες, χωρίς την επίλυση επιπλέον εξισώσεων μεταφοράς. Ωστόσο, βασίζονται σε απλουστεύσεις, όπως η υπόθεση του «μήκους ανάμιξης» του Prandtl, γεγονός που τα καθιστά ανακριβή για σύνθετες ροές.

Οι προσεγγίσεις που βασίζονται σε δεδομένα έχουν αποδειχθεί υποσχόμενες όσον αφορά τη δημιουργία υποκατάστατων μοντέλων τύρβης και μετάβασης. Πιο συγκεκριμένα, τα υποκατάστατα μοντέλα που βασίζονται σε Βαθιά Νευρωνικά Δίκτυα έχουν επιδείξει μεγάλη ακρίβεια, ωστόσο είναι ιδιαίτερα ευαίσθητα στην υπερπροσαρμογή και στην αδυναμία γενίκευσης. Επιπλέον, η φύση τους ως «μαύρο κουτί» δεν αφήνει

περιθώρια για την ερμηνεία του υποκατάστατου μοντέλου. Ακόμη, δεν είναι δυνατή μια αναλυτική διαφόριση, με σκοπό την ενσωμάτωση του μοντέλου σε μια διαδικασία βελτιστοποίησης.

Η παρούσα διπλωματική εργασία διερευνά μια καινοτόμο προσέγγιση βασισμένη σε δεδομένα, κατασκευάζοντας αλγεβρικά υποκατάστατα μοντέλα τύρβης και μετάβασης με τη χρήση Συμβολικής Παλινδρόμησης. Αξιοποιώντας τον αλγόριθμο PySR, η εργασία αυτή στοχεύει στην αντικατάσταση των υπολογιστικά κοστοβόρων εξισώσεων μεταφοράς με αλγεβρικές εκφράσεις κλειστής μορφής, οι οποίες συσχετίζουν τοπικές ροϊκές και γεωμετρικές ποσότητες με μεταβλητές-στόχους, όπως ο παράγοντας διάλειψης γ και η τυρβώδης κινηματική συνεκτικότητα ν_t. Αυτό εξασφαλίζει τόσο φυσική διαφάνεια όσο και αναλυτική παραγωγισιμότητα, μειώνοντας παράλληλα τον κίνδυνο υπερπροσαρμογής, εξαλείφοντας έτσι τα μειονεκτήματα των προηγούμενων προσεγγίσεων που βασίζονταν σε δεδομένα.

Η μεθοδολογία επικυρώνεται σε ποικίλες εφαρμογές. Για τη μοντελοποίηση της μετάβασης, αυτές περιλαμβάνουν την αεροτομή NLF(1)-0416, η οποία σχεδιάστηκε από τη NASA για φυσική στρωτή ροή, και την αεροτομή ONERA OAT15A, η οποία χρησιμοποιείται για τη μελέτη διηχητικών ροών και φαινομένων αλληλεπίδρασης του τυρβώδους οριακού στρώματος με το κρουστικό κύμα. Επιπλέον, οι δύο αεροτομές χρησιμοποιούνται σε συνδυασμό για την εξαγωγή μιας ενιαίας, πιο γενικευμένης αλγεβρικής έκφρασης για το πεδίο του παράγοντα διάλειψης. Τα υποκατάστατα μοντέλα που βασίζονται στη Συμβολική Παλινδρόμηση για την παραλλαγή Piotrowski-Zingg του μοντέλου $\gamma - Re_\theta$ ενσωματώνονται σε μια προσομοίωση RANS στον επιλυτή PUMA για αξιολόγηση. Όσον αφορά τη μοντελοποίηση τύρβης, το πλαίσιο εφαρμογής χρησιμοποιείται στην αεροτομή NACA 0012 και στο Ahmed Body με κλίση 35°. Αυτές οι εφαρμογές χρησιμεύουν στη βελτίωση της μοντελοποίησης των περιοχών αποκόλλησης και ανακυκλοφορίας. Τα προκύπτοντα αλγεβρικά υποκατάστατα μοντέλα ενσωματώνονται στον RANS επιλυτή του OpenFOAM για την αντικατάσταση του μοντέλου Spalart-Allmaras, και η απόδοσή τους αξιολογείται στη συνέχεια μέσα σε έναν βρόχο Βελτιστοποίησης Σχήματος, προκειμένου να ελεγχθούν τα πλεονεκτήματα ενός δυναμικά ανανεούμενου πεδίου τυρβώδους συνεκτικότητας σε σύγκριση με την προσέγγιση του «παγωμένου» πεδίου.

List of Abbreviations

AoA	Angle of Attack.
CFD	Computational Fluid Dynamics.
CPU	Central Processing Unit.
DES	Detatched Eddy Simulation.
DNN	Deep Neural Network.
EA	Evolutionary Algorithm.
EVA	Eddy Viscosity Adaptation.
FAE	Field Adjoint Equations.
FEV	Frozen Eddy Viscosity.
FISR	Field Inversion Symbolic Regression.
GPU	Graphics Processing Unit.
LES	Large Eddy Simulation.
ML	Machine Learning.
MOO	Multi-Objective Optimization.
MSE	Mean Squared Error.
NLF	Natural Laminar Flow.
NTUA	National Technical University of Athens.
PCOpt	Parallel CFD & Optimization.
PUMA	Parallel Unstructured Adjoint Multi-row.
RANS	Reynolds Averaged Navier-Stokes.
SA	Spalart-Allmaras.
ShpO	Shape Optimization.
SOO	Single-Objective Optimization.
SR	Symbolic Regression.

Contents

Chapter 1

Introduction

1.1 Turbulence and Transition Modeling in CFD

Turbulent flow is by definition a stochastic phenomenon, characterized by chaotic spatial and temporal changes in the flow quantities. The unpredictability of turbulent phenomena creates the need for statistical methods and empirical models, in order to study them, as high-fidelity methods, such as Direct Numerical Simulation (DNS), are in many cases computationally prohibitive. Traditional turbulence models in Computational Fluid Dynamics (CFD) are categorized depending on the number of differential equations they add to the system, as follows:

- **Algebraic or Zero-Equation models** are based on Prandtl's "mixing length" hypothesis [37], which suggests that the turbulent eddy viscosity ν_t must vary with distance from the wall. Notable examples include the Cebeci-Smith [7] and Baldwin-Lomax [4] models, which determine the eddy viscosity directly from local flow quantities without solving additional transport equations.

- **One-Equation models** introduce a single additional transport equation, typically for the turbulent kinetic energy or a modified eddy viscosity. The Spalart-Allmaras (SA) model [45] is the most prominent example in external aerodynamics. It offers a robust compromise between computational cost and accuracy for boundary layer flows under adverse pressure gradients. The SA model also includes the f_{t1} term, meant to provide a basic modeling of the laminar-to-turbulent transition.

- **Two-Equation models** are the most widely used in industrial CFD, solving two separate transport equations. The most common formulations are the $k-\epsilon$ [24], $k-\omega$ [53] and $k-\omega-SST$ [28] models. These provide a more complete description of the turbulent state, though they remain limited by the Boussinesq hypothesis and often struggle with predicting the onset of transition without specific modifications.

Fluid flow rarely exists in a binary state of laminar or turbulent. It undergoes a complex transition process. This regime is highly sensitive to freestream pressure gradients and wall roughness. Accurately predicting the transition point is critical for aerodynamic design, as it dictates the sudden rise in skin friction and thermal loading. The so-called ”onset of turbulence” can occur through three fundamental mechanisms: natural, bypass and separation-induced transition [21]. Natural transition, most prominent in wings, is driven by the growth of ”Tollmien - Schlichting” instabilities within in the laminar boundary layer, which eventually break down into turbulent flow [39]. Bypass transition [40], appearing mainly in gas turbines, refers to the bypassing of the classical linear instability phase due to high intensity of external flow turbulence. Separation-induced transition [27] occurs mainly in gas turbines, low-pressure turbines and compressors in the separated boundary layer, where the subsequent reattachment of the flow often forms a laminar-separation bubble. A fourth category of transition is forced transition, where the laminar boundary layer is intentionally disturbed by artificial surface roughness or mechanical ”trips”. If the roughness elements are sufficiently large, the generated disturbances result in an immediate transition to a fully turbulent state. This method is often employed in experimental aerodynamics to ensure a predictable transition point across varying flow conditions [14].

Laminar-to-turbulent transition modeling plays a crucial role in maintaining accuracy in CFD simulations, as transitional phenomena directly affect the skin friction, lift and drag coefficients, which are highly important for aerodynamic design. Without a robust transition model, no accurate prediction of the aforementioned integral quantities can be achieved and, thus, any related design or optimization process is rendered unsuccessful. Historically, transition was accounted for using empirical

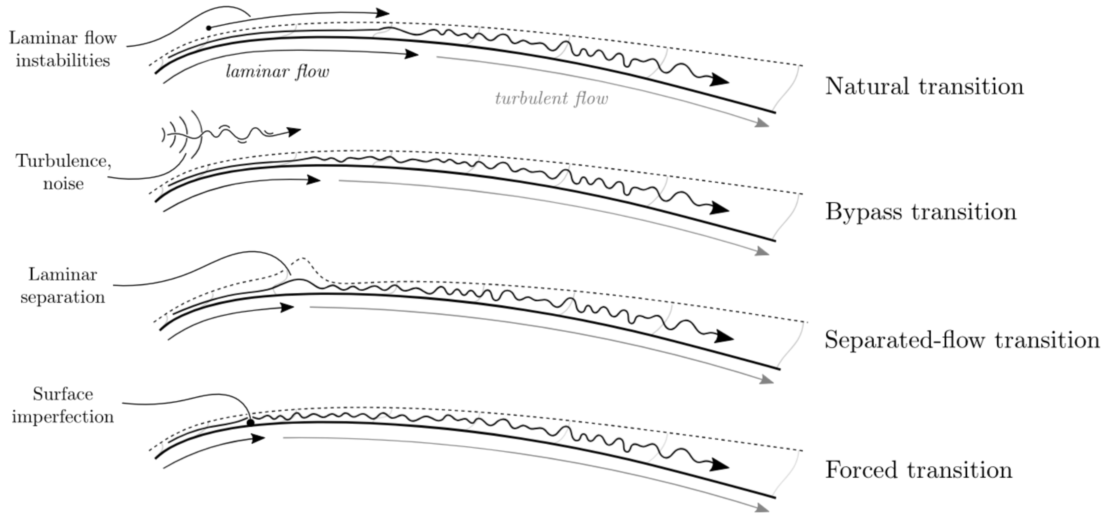


Figure 1.1: *Schematic drawing of four types of laminar-to-turbulent transition of the boundary layer [11].*

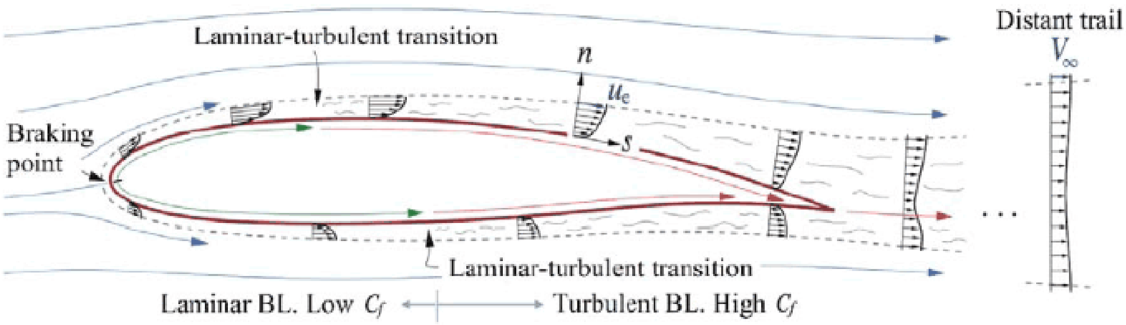


Figure 1.2: *Laminar to turbulent transition in the boudary layer of an airfoil [6].*

correlations or the e^N semi-empirical method [51] based on linear stability theory [39]. The e^N method is still a standard regarding natural laminar flows (NLF). However, modern CFD often utilizes local correlation-based transition models (LCTM), such as the $\gamma - Re_\theta$ model [29] or the one-equation γ model [30], which are integrated directly into the Reynolds-Averaged Navier-Stokes (RANS) equations. These models rely on the concept of intermittency γ to trigger the transition process [15], effectively bridging the gap between laminar and turbulent regimes in a numerically robust manner.

1.2 CFD-based Optimization

CFD-based optimization is the process of solving fluid-mechanical design problems by utilizing CFD solvers as the primary evaluation tool. Within this framework, the optimization strategy can either be gradient-free (such as Evolutionary Algorithms) or gradient-based (such as the continuous adjoint formulation), while the objectives are usually aerodynamic performance metrics. Noteworthy examples include maximizing the lift to drag ratio of an airfoil and minimizing of the total pressure loss in internal flows. CFD-based Optimization can refer to Shape, Topology or flow-control optimization. The first aims to find the optimal shape of a given geometry in order to achieve the optimal solution to the problem. In topology optimization the algorithm determines where material should and should not exist within a given design volume. Flow-control optimization aims to optimize by manipulating the flow without interfering with the shape of the geometry. This approach typically involves passive techniques, such as vortex generators and active methods, such as synthetic jets, suction or blowing, order to modulate the transition onset and reduce pressure drag.

1.3 Symbolic Regression

Regression analysis refers to the process of determining the relationship between a dependent and one or multiple indepedent variables, when only a set of data points is provided. In the context of science and engineering, it serves the purpose of yielding simple mathematical expressions that describe complex physical problems, whose behavior cannot otherwise easily be predicted. Depending on the structure of the regression algorithm and the nature of the search space, several regression types can be identified. The most basic form is **Linear Regression**, which is based on the assumption of linear dependency between the input and output variables. While computationally efficient, it is considered insufficient for capturing non-linear and more complex distributions. **Non-Linear Regression** refers to the process of fitting the data to a pre-defined non-linear expression (e.g. higher-order polynomials, exponential, logarithmic) by determining its coefficients. However, its success depends heavily on the researcher's a-priori knowledge of the ”correct” functional form to be used for the fit. The first two forms are strict mathematical methodologies, which can easily be calibrated for simple systems but often fail to capture the high-dimensional, non-linear distributions often encountered in physical problems. **"Black-box" Machine Learning-based Regression** is approached in the past decades through Deep Neural Networks (DNNs), which can accurately approximate nearly any complex distribution. The so-called "black-box" nature lies within their inability to provide explicit mathematical relationships, rendering them impossible to interpret and analytically differentiate for optimization purposes.

Symbolic Regression (SR) serves as the central methodology of this thesis for the yielding of algebraic expressions by fitting them to provided data. The difference with traditional linear or non-linear regression is that the form of the algebraic expression is not pre-determined. This suggests that the algorithm does not seek to find the coefficients that fit the data into an already decided mathematical expression, but also to optimize the form of the expression itself. This fact renders this method very useful, when no clue of the underlying mathematical relationship between the variables exists, or when the physical phenomena are too complex to be captured by traditional empirical correlations. SR fits the data to the simplest functional equation, meaning that sometimes better generalization can be achieved. This renders SR-generated expressions more successful in extrapolation to unseen data, unlike DNNs, which are great interpolators but often struggle with extrapolation, due to overfitting.

Any SR algorithm works by creating an "expression-tree" to describe the canditate algebraic expression, as illustrated in Fig. 1.3. Each node contains either an independent variable, a mathematical operator (either unary or binary) or a constant. A unary operator is a mathematical operator. which is fed with only one input (e.g. the trigonometric, exponential and logarithmic functions), whereas a binary operator needs two separate inputs (e.g. the power, addition, subtraction, multiplication and division functions). The algorithm seeks to optimize both the structure of the tree and the numerical constants within the nodes to achieve the best fit to the data while simultaneously minimizing complexity. In this context, the expression's complexity is defined as the number of nodes in the expression tree.

The most prominently used SR packages are described in the following sections.

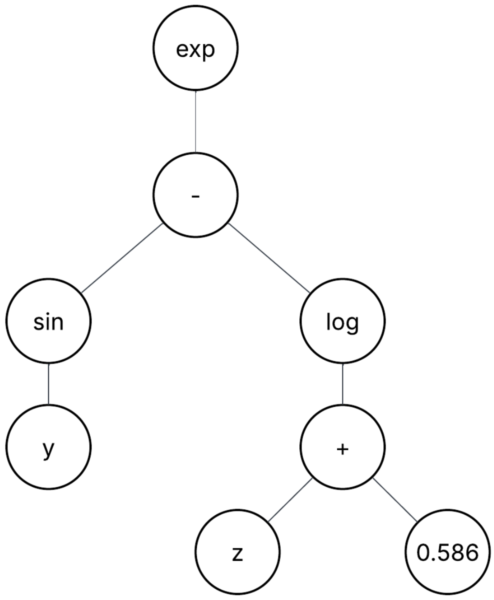


Figure 1.3: *Example of Expression Trees used in Symbolic Regression Algorithms.*

They are distinguished mainly by their search mechanisms, which employ either gradient-free or gradient-based methods, in order to optimize the design variables, which are the mathematical operators, coefficients and constants in the expression tree.

PySR [9] is a high-performance open-source library that has become a standard in engineering. A high-speed Julia backend for the core search process is implemented, while a user-friendly Python interface is maintained. The search mechanism consists of a Multi-Population EA, where independent populations of expressions, the so-called islands, evolve in parallel and periodically migrate to share genetic information. A key feature of PySR is its integration of Simulated Annealing in the mutation process to navigate complex search spaces and an optimization loop for the expression's constants, typically employing the BFGS algorithm, to tune the coefficients.

Gene Expression Programming (GEP) [52] refers to a specialized evolutionary logic that has successfully been used in Explicit Algebraic Reynolds Stress Models (EARSM). Unlike standard Genetic Programming, GEP utilizes fixed-length linear chromosomes that are subsequently expressed as non-linear expression trees. This separation between the genotype and the phenotype allows for highly efficient genetic operations during the search for optimal expressions.

Eureqa [41] was the software that popularized symbolic regression across scientific research. It was designed to uncover fundamental physical laws directly from experimental data. The engine prioritizes the yielding of the simplest possible expressions, a principle known as parsimony. This ensures that the resulting formulas are physically interpretable and not overfitted to the training data.

Deep Symbolic Regression (DSR) [34] introduces neural-guided search. Instead of relying on traditional evolutionary operators, DSR employs a Recurrent Neural Network (RNN), which is trained using Reinforcement Learning, specifically risk-seeking policy gradients. This strategy favors expressions that maximize accuracy while respecting predefined mathematical constraints.

GPLearn [47] is a widely used Python library for integrating SR into general data science workflows. It is fully compatible with the scikit-learn API, allowing the user to implement symbolic regression as a standard estimator. It follows a classical Genetic Programming ”survival of the fittest” approach.

Package	Primary Algorithm	Key Advantage
PySR	Multi-Population EA	Speed & Constant Optimization
GEP	Geno-/Phenotype Evolution	Proven success in EARSM
Eureqa	Genetic Programming	Historical standard for physical laws
DSR	Reinforcement Learning	Neural-guided search
GPLearn	Genetic Programming	Easy use & scikit-learn compatibility

Table 1.1: *Summary of commonly used Symbolic Regression packages.*

1.4 Thesis Motivation

The motivation for this thesis stems from one of the most significant challenges in CFD. The accurate prediction of turbulent flows and laminar-to-turbulent transition demands, in certain cases, high-fidelity model simulations with very high computational cost, like Large Eddy Simulation (LES) [43] or Detached Eddy Simulation (DES) [46]. This renders them practically impossible for being used in industrial-level applications [12]. However, losing accuracy in the name of minimizing the computational cost is simply unacceptable, as this can lead to a completely false prediction of aerodynamic quantities, such as the lift and drag coefficients, which are of uttermost importance for the design and optimization process in any industrial case. Thus, Data-Driven Turbulence and Transition Modeling has been gaining increasing popularity over the years [13, 5].

In view of maintaining high accuracy, while speeding up the computations, Deep Neural Networks (DNNs) are widely implemented in turbulence and transition modeling [25]. They are trained on experimental or high-fidelity simulations data and can offer impressively accurate predictions, and can also be tuned to fit certain cases or geometries almost perfectly. Their disadvantage lies within their "black-box" nature. This means that their prediction, while very trustworthy in terms of accuracy, offers no information concerning the physical problem and the quantities used in the calculation of the final result. Consequently, the model cannot be manipulated or generalized to data outside of the training set quite easily.

The previously-described gap, concerning the need for faster, human-interpretable turbulence and transition models, holds the potential to be bridged by an innovative tool, namely Symbolic Regression [9, 52]. Falling under the category of Interpretable Machine Learning (ML), SR can provide analytical expressions for any quantity of interest, such as the transition intermittency γ and the turbulent eddy viscosity ν_t. The acquisition of an algebraic expression leads to the ability to interpret and manipulate the quantities involved, while also easily implementing the equation in methods such as the continuous adjoint method for optimization purposes [55]. The continuous adjoint method, which is widely used for calculating sensitivities with respect to design variables, requires the linearization of the turbulence and transition model equations [18]. For complex neural architectures, calculating these derivatives — while possible through automatic differentiation — remains a "black-box" process that adds significant computational cost and memory requirements [21]. The explicit nature of SR-derived expressions simplifies this process considerably. Having a symbolic expression allows for the straightforward derivation of the adjoint counterparts, facilitating a more efficient and "frozen-free" optimization workflow where the transition or turbulence model is fully coupled with the adjoint solver. This approach holds a great advantage in shape optimization cycles, as the surrogate model uses geometrical information and updates the flowfield dynamically, contributing to a more accurate optimization process, compared to a frozen field approach. Finally, the "symbolic" nature of these models addresses the persistent issue of over-fitting.

While deep learning models typically require vast amounts of high-fidelity data to learn the underlying physics, SR often thrives on sparser datasets by seeking the simplest functional form that satisfies the data [34]. This suggests that SR-generated expressions may possess superior generalization capabilities when applied to unseen geometries or flow conditions that differ from the training data [54, 19]. This thesis explores this potential, aiming to develop surrogate models that are not only computationally efficient and accurate, but also physically interpretable and adaptable to diverse cases.

1.5 Thesis Workflow

The workflow followed in the subsequent chapters of this diploma thesis is organized into five distinct stages, as illustrated in the general methodology overview:

1. **Reference Data Generation:** High-fidelity CFD simulations are performed to generate the reference fields (e.g. intermittency γ or eddy viscosity ν_t).

2. **Input Variable Selection:** Local flow and geometric variables are extracted and normalized to serve as independent variables for the regression process.

3. **Symbolic Regression Training:** The SR algorithm is employed offline to search the solution space and derive algebraic relations between the input variables defined in Step 2 and the output variables, which are either the intermittency factor or the eddy viscosity, depending on the application. It seeks a trade-off between predictive accuracy (MSE) and minimal complexity (parsimony principle), yielding a Pareto front of algebraic expressions.

4. **Implementation in the CFD Solver:** The selected algebraic surrogate is integrated into the RANS solver (PUMA or OpenFOAM). This replaces the traditional closure model (either turbulence or transition), allowing the flow field to be updated dynamically based on the new algebraic surrogate. The results are evaluated based on post-processed quantities such as the aerodynamic coefficients (C_D and C_L) and boundary layer-related quantities (C_f), as well as by visualizing the computed fields.

5. **Adjoint-based ShpO:** In ShpO for turbulent flows, the algebraic surrogate

model is employed within the optimization framework, dynamically updating the eddy viscosity field in each optimization cycle, making changes in the flow field due to shape updates detectable, as opposed to a frozen eddy viscosity optimization loop.

The algebraic surrogates developed in this work offer transparency, as the explicit nature of the yielded expressions allow for immediate inspection and interpretability. Furthermore, they can seamlessly be integrated into the RANS solvers to minimize the computational cost, as well as within the continuous adjoint fomulation, providing exact sensitivity derivatives after analytical differentiation. Lastly, compared to the more common DNN-based suroggate modeling, these surrogates are less prone to overfitting, due to the algorithm's goal to always seek the simplest functional form.

1.6 Thesis Outline

This thesis aims to construct analytical turbulence and transition surrogate models using Symbolic Regression and to test their accuracy in CFD RANS calculations. The turbulence models created are also implemented in Shape Optimization (ShpO), to test whether or not they provide useful information to the optimization algorithm compared to a Frozen Eddy Viscosity (FEV) field.

- In **Chapter 2** the concept of Turbulence and Laminar-to-Turbulent Transition is described, while the Chapter also presents the RANS equations for incompressible flow, the Spalart-Allmaras Turbulence Model and the $\gamma - Re_\theta$ Transition Model. A literature survey of algebraic turbulence and transition models is conducted and the emergence of Data-Driven modeling is described.

- In **Chapter 3** the process of Optimization especially with Evolutionary Algorithms (EAs) and the Continuous Adjoint Optimization Method is explained and the concept of Shape Optimization (ShpO) is presented.

- In **Chapter 4** Symbolic Regression (SR) as part of Interpretable Machine Learning (ML) is introduced. The most widely used SR packages are presented and finally, the PySR algorithm used for SR purposes in this thesis is also explained in detail. A simple example of use of the PySR algorithm is presented in order to render the reader more familiar with the Regressor.

- In **Chapter 5** the algebraic surrogates to the transition model built through SR for the NLF(1)-0416 and ONERA OAT15A airfoils are presented. Their accuracy is tested in a number of conditions, in terms of transition point and lift and drag coefficients prediction.
- In **Chapter 6** the algebraic surrogates to the turbulence model built through SR for the NACA 0012 airfoil and the Ahmed body are presented and the results of their integration in simple RANS simulations, as well as in a ShpO loop are evaluated.
- In **Chapter 7** a summary of the findings of the previous chapters is presented, alongside the essential conclusions and proposals for future scientific research based on the methodology developed in this thesis.

Chapter 2

Turbulence

Turbulent flow in fluid dynamics refers to chaotic changes in velocity and pressure, as opposed to laminar flow, where the fluid follows discrete parallel layers. The most prominent characteristic of turbulent flow is the formation of vortices (eddies), with varying sizes and vorticities, due to the spatial and temporal fluctuations in the fundamental flow quantities. Due to its stochastic nature, it is not studied deterministically, but through statistical methods and models. It is by nature a three-dimensional and time-dependent phenomenon [42].

2.1 Turbulence Modeling

Everything stated above leads to the conclusion that a full resolution of turbulent flows is computationally prohibitive for complex geometries, leaving only approximation strategies and modeling, in order to obtain sufficiently accurate results to use in engineering applications. Turbulence modeling has been one of the most challenging problems in CFD over the past decades. While the Navier-Stokes equations accurately govern these flows, the computational cost of resolving the Kolmogorov scale—where kinetic energy dissipates into heat—remains prohibitive for most engineering applications.

The computational methods used for turbulence modeling in descending order of accuracy are arranged as follows. The Direct Numerical Simulation (DNS) [8] remains the most accurate and the computational overhead to the problem, as it solves the Navier-Stokes equations in every scale, ranging from the largest eddies, which contribute the most to the kinetic energy to the smallest eddies, termed the Kolmogorov Scale [20]. Performing DNS in industrial applications requires massive computational means (as the cost increases with the cube of the Reynolds number), exceeding the capabilities of common computers and clusters and limiting the implementation only to supercomputers. The Large Eddy Simulation (LES) [16] methodology is a compromise made in view of reducing the computational cost to a certain degree, as an exact calculation of the Navier Stokes equations is calculated for the large scale eddies, but the smaller eddies, whose energy contribution is significantly lower, are modeled. The LES method remains, however, too computationally demanding to implement routinely in industrial simulations. The Reynolds - Averaged Navier - Stokes (RANS) equations [38] are the most computationally efficient approach, as they solve for the time-averaged flow field. By decomposing the instantaneous velocity into mean and fluctuating components, the effect of turbulence is captured through turbulence models which close the system of equations. While RANS sacrifices the resolution of unsteady structures, it provides the necessary balance of accuracy and speed required for engineering design.

2.1.1 The RANS Equations

The Reynolds-Averaged Navier-Stokes (RANS) equations are based on the assumption that any flow quantity ϕ in turbulent flows can be decomposed into a mean term $\bar{\phi}$ and a fluctuation term ϕ' . This is known as Reynolds decomposition.

For the sake of brevity the resulting RANS equations are presented here for the case of an incompressible fluid:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{2.1}$$

$$\rho \bar{v}_j \frac{\partial \bar{v}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial \bar{v}_j}{\partial x_i} \right) - \rho \overline{v_i' v_j'} \right] \tag{2.2}$$

In these equations:

- $\bar{v}_i$ is the mean velocity component in the i-direction.

- $\bar{p}$ is the mean pressure.
- ρ and μ represent the fluid density and dynamic viscosity, respectively.
- $\tau_{ij} = -\rho\overline{v_i'v_j'}$ is the Reynolds stress tensor.

To close the system of equations, the Boussinesq hypothesis is employed, which relates the Reynolds stresses to the mean velocity gradients through the eddy viscosity ν_t:

$$-\overline{v_i'v_j'} = \nu_t\left(\frac{\partial\bar{v}_i}{\partial x_j} + \frac{\partial\bar{v}_j}{\partial x_i}\right) - \frac{2}{3}k\delta_{ij} \tag{2.3}$$

where k is the turbulent kinetic energy of the fluid, defined as follows:

$$k = \frac{1}{2}\overline{v_i'v_j'} \tag{2.4}$$

While cases involving higher Mach numbers require the Favre-averaging formulation to account for density fluctuations, the incompressible form clearly highlights the emergence of the Reynolds stress tensor, which necessitates the turbulence closures discussed in the following sections.

2.1.2 Algebraic Turbulence Models

The development of algebraic turbulence models stems from the ”mixing length” hypothesis proposed by Prandtl (1925) [37]. The most significant milestones in this category include the Cebeci-Smith Model (1967) [7], treating the boundary layer as two layers and constucting a separate eddy viscosity equation for each one, and the Baldwin-Lomax Model (1978) [4], which eliminates the need to define the boundary layer edge, leading to more robust modeling for separated flows.
Algebraic transition models do not solve a transport equation for the intermittency factor, but rather ”switch” the turbulence model on based on local flow conditions. The most noteworthy work in this area includes Michel’s Criterion (1954) [31], containing a classical empirical correlation relating the local Reynolds number based on momentum thickness (θ) to the Reynolds number based on the distance from the leading edge (x), the Abu-Ghannam and Shaw (1980) [1], which accounts for the effects of free-stream turbulence intensity and pressure gradients and the more

recent Algebraic γ Model, developed by Kubacki and Dick (2016) [23].

2.1.3 The Spalart-Allmaras Turbulence Model

The Spalart-Allmaras model is a one-equation model that solves a transport equation for a modified turbulent viscosity $\tilde{\nu}$. The evolution of this variable is governed by the following partial differential equation:

$$\begin{aligned}\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = \gamma \cdot \left[C_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu}\right] - \left[C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2}\right] \left(\frac{\tilde{\nu}}{d}\right)^2 \\ + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right) + C_{b2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}\right]\end{aligned} \tag{2.5}$$

In Eq. 2.5, the production term is multiplied by the intermittency factor γ to account for the laminar-to-turbulent transition process. A detailed derivation of the transport equation for γ will be presented in 2.2.1.

The kinematic eddy viscosity ν_t is then obtained from the working variable using a damping function f_{v1}:

$$\nu_t = \tilde{\nu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu} \tag{2.6}$$

The modified vorticity $\tilde{S}$, which prevents the production term from vanishing at the wall, is defined as:

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{2.7}$$

where S is the magnitude of the vorticity and d is the distance to the closest wall. The destruction of turbulent viscosity near the wall is controlled by the f_w function:

$$f_w = g\left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6}\right]^{1/6}, \quad g = r + C_{w2}(r^6 - r), \quad r = \min\left[\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10\right] \tag{2.8}$$

To handle the transition from laminar to turbulent flow, the model includes the f_{t1} and f_{t2} trip functions:

$$f_{t1} = C_{t1} g_t \exp\left(-C_{t2} \frac{\omega_t^2}{\Delta U^2} [d^2 + g_t^2 d_t^2]\right) \tag{2.9}$$

$$f_{t2} = C_{t3} \exp(-C_{t4} \chi^2) \tag{2.10}$$

The standard closure constants for the model are:

- $C_{b1} = 0.1355, \quad C_{b2} = 0.622, \quad \sigma = 2/3, \quad \kappa = 0.41$
- $C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1+C_{b2}}{\sigma}, \quad C_{w2} = 0.3, \quad C_{w3} = 2.0, \quad C_{v1} = 7.1$
- $C_{t1} = 1.0, \quad C_{t2} = 2.0, \quad C_{t3} = 1.2, \quad C_{t4} = 0.5$

2.2 Laminar-to-Turbulent Transition

2.2.1 The $\gamma - Re_\theta$ Transition Model

The transition model employed in this thesis is based on the local correlation-based transition $\gamma - Re_\theta$ framework originally proposed by Langtry and Menter (2009) [29]. Specifically, a modified, numerically smooth one-equation version developed by Piotrowski and Zingg (2021) [36] is used to be differentiable for the continuous adjoint formulation. This model introduces a transport equation for the intermittency factor, γ, which controls the production term of the SA model. The transport equation for γ is given by:

$$\bar{v}_j \frac{\partial \gamma}{\partial x_j} = P_\gamma - E_\gamma + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\gamma}\right)\frac{\partial \gamma}{\partial x_j}\right] \tag{2.11}$$

Where:

- P_γ is the transition source (production) term, which triggers the growth of intermittency based on local flow correlations.

- E_γ is the destruction (relaminarization) term.

- σ_γ is the effective Schmidt number for the diffusion of intermittency.

The coupling with the Spalart-Allmaras model is achieved by multiplying the original production term of the $\tilde{\nu}$ equation by the intermittency factor γ. In the transitional regions, γ varies from 0 (laminar) to 1 (fully turbulent), effectively "turning on" the turbulence production as the flow transitions. Unlike earlier transition models, this formulation ensures numerical smoothness, making it particularly suitable for gradient-based aerodynamic shape optimization using adjoint methods.

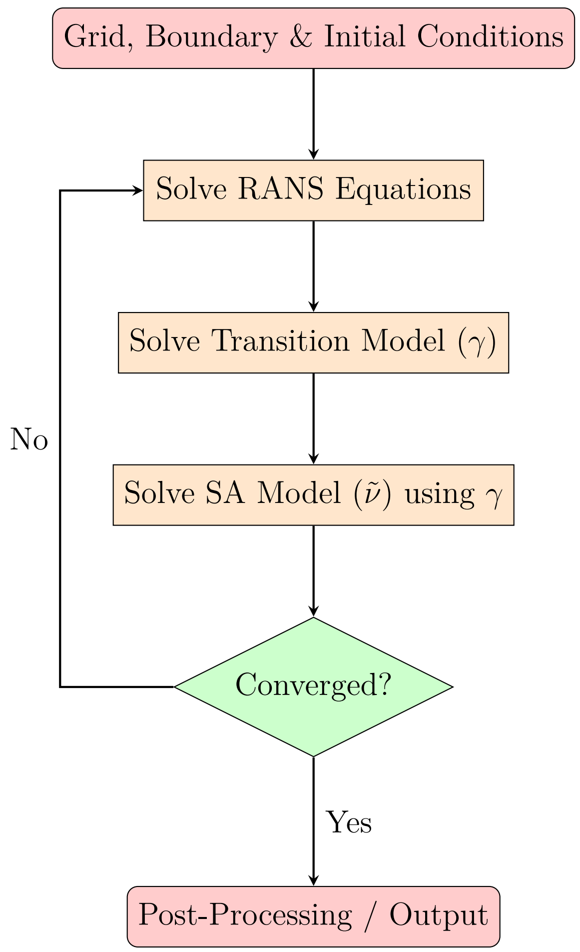


Figure 2.1: *Flowchart of the coupled RANS-Transition Model-SA solver iteration process.*

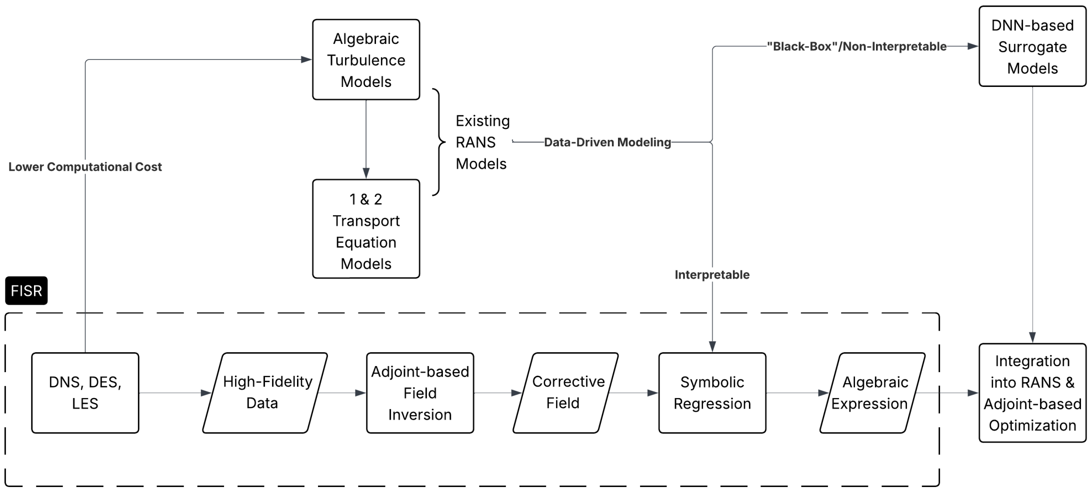


Figure 2.2: *Evolution of Turbulence Modeling and description of the FISR Methodology*

2.3 Data-Driven Turbulence & Transition Modeling via Symbolic Regression

The limitations of existing algebraic models in combination with the increased computational cost of one or two transport equation models have led, in recent years, to the emergence of data-driven turbulence and transition modeling as a robust alternative [12].

Surrogate models based on DNNs have demonstrated high accuracy in reproducing high-fidelity CFD data [21, 25]. However, their "black-box" nature remains a significant barrier for engineering applications, as they offer no physical insight and can exhibit numerical instability when integrated into RANS solvers. The extrapolation of DNNs into flow conditions or geometries outside of the training dataset often leads to predictions with no physical meaning. Furthermore, from an optimization perspective, the complex and non-linear architecture of a DNN complicates the calculation of sensitivities. While automatic differentiation can be employed to obtain derivatives for gradient-based methods, the process often introduces a significant computational and memory overhead.

To bridge this gap, SR-based approaches have been introduced in recent years, in order to provide interpretable transition and turbulence models, using local flow quantities [52, 5]. A pioneering application in this domain was the use of the Gene

Expression Programming (GEP) framework to identify improved Explicit Algebraic Reynolds Stress Models (EARSM) [52]. This approach has been subsequently extended to transition modeling through the Eddy Viscosity Adaptation and Symbolic Regression (EVA-SR). The first stage of the EVA-SR methodology is the Eddy Viscosity Adaptation process, which aims to solve an optimization problem, often using the adjoint method, to obtain a spatially varying corrective field high-fidelity data, such as DNS or LES. The produced field represents the descrepancy between reference high-fidelity data and the RANS baseline. In the second stage, Symbolic Regression is used as the tool to map this field to a set of flow and geometrical variables. Unlike neural networks, the resulting SR-derived models are closed-form algebraic expressions. This advantage is of uttermost importance in ensuring physical consistency (e.g. ensuring no negative values of eddy viscosity are produced) and in integrating the surrogate model within the continuous adjoint-based optimization framework, as the expression is analytically differentiable [55, 19]. SR-based transition modeling is a more recent approach [56] and preliminary studies show that the reproduction of high-fidelity data in terms of reconstructing transition mechanisms is possible through SR. However, the full integration of such surrogate models within adjoint-based design optimization cycles remains an open area of research. Based on the above, this thesis aims to evaluate the accuracy of SR-based algebraic surrogate turbulence and transition models and subsequently their performance in a shape optimization framework.

2.4 CFD Simulations

Some of the simulations performed in this thesis use the in-house CFD GPU-enabled solver PUMA, developed by the Parallel CFD & Optimization (PCOpt) Unit of the National Technical University of Athens (NTUA) [57, 3, 48, 49, 32]. The rest use the open-source OpenFOAM (Open Field Operation and Manipulation) software [26]. While both solvers are based on the Finite Volume (FV) method, they differ in their spatial discretization approach, as PUMA utilizes a vertex-centered formulation, whereas OpenFOAM follows a cell-centered convention. PUMA is built in a way that allows for massive parallelization on GPUs using the CUDA/C/C++ framework. This allows significant acceleration of the RANS and adjoint computations, compared to traditional CPU-based execution. OpenFOAM uses a general purpose C++ framework, which enables the user to integrate new models without changing the underlying numerical engine. Its extensive library of built-in tools renders it ideal for testing a variety of cases. Lastly, PUMA simulates compressible flows, whereas OpenFOAM is restricted to incompressible cases.

Chapter 3

Optimization

An optimization problem refers to the search for the optimal solution $\vec{X}_{opt}$ to a specified problem, through minimizing or maximizing a given objective function $F(\vec{X})$. The objective function is essentially the target quantity of the problem, which the process seeks to optimize. Optimization problems can be classified depending on the number of objectives into two categories: Single-Objective Optimization (SOO), in which there is only one objective and Multi-Objective Optimization (MOO), where there is a greater number of conflicting objectives. In a Multi-Objective Optimization (MOO) problem, multiple solutions with equivalent fitness—in the sense that no single objective can be improved without degrading another—are acquired and presented as a Pareto Front. Each point on this front represents a different trade-off, where the relative importance of one objective is prioritized over the others.

3.1 Shape Optimization

Shape Optimization (ShpO) is the process of finding the optimal shape of a given geometry, in order to minimize or maximize an objective function. Considering the fact that the methodology aims at finding an optimal shape, the design variables $\vec{b}$ determine the geometry and the algorithm relies on the computation of sensitivity derivatives (SDs) to iteratively update the given shape. A classic example of aerodynamic ShpO is the search for the optimal shape of an airfoil which maximizes its lift and minimizes or retains its drag coefficient. The two objectives are conflicting, rendering this specific example a MOO problem.

In the context of this thesis, the SR-based turbulence surrogate models constructed through SR in the present thesis are used for ShpO purposes and the results are compared with the Frozen Eddy Viscosity (FEV) Method [50], which implements a frozen ν_t corrective field in the primal solver and assumes laminar flow in the adjoint solver. The implementation of the surrogate model in the primal solver of the ShpO process holds a significant advantage against the FEV approach, as the eddy viscosity field is dynamically updating depending on geometrical variables within the algebraic expression. This can provide useful information, which would otherwise be missing.

3.2 Optimization Methods

Optimization methods are split into Gradient-Free or Stochastic and Gradient-Based or Deterministic, depending on whether they utilize or not the derivatives of the objective function w.r.t the design variables in order to find an optimum. Stochastic methods rely on probabilistic algorithms and a ”survival of the fittest” methodology, whereas deterministic optimization determines the direction of the next iteration based on local derivative values.

3.2.1 Gradient-Free Methods

Evolutionary Algorithms (EAs) [17] refer to a gradient-free stochastic optimization method, based on Darwin's Theory of Evolution [10]. They are population-based and rely on processes like the application of the crossover and mutation operations on a set of selected individuals in each generation, in order to create new individuals, whose fitness is once again evaluated. Their stochastic nature lies within the probability-based parents' selection and crossover and mutation application. The implementation of methods such as elitism can significantly speed-up the algorithm's convergence, while population distribution makes parallelization possible, which is of uttermost importance in industial applications. The advantages of EAs lie within the fact that, by implementing certain strategies, they are less likely to be trapped in a local optimum. They maintain high diversity in the populations' individuals, due to the employment of evolution operators, which allow them to navigate the entirety of the search space. A flow-chart of a basic EA optimization-loop is presented in Fig. 3.1. In this thesis, EAs are implemented for Symbolic Regression within the PySR library.

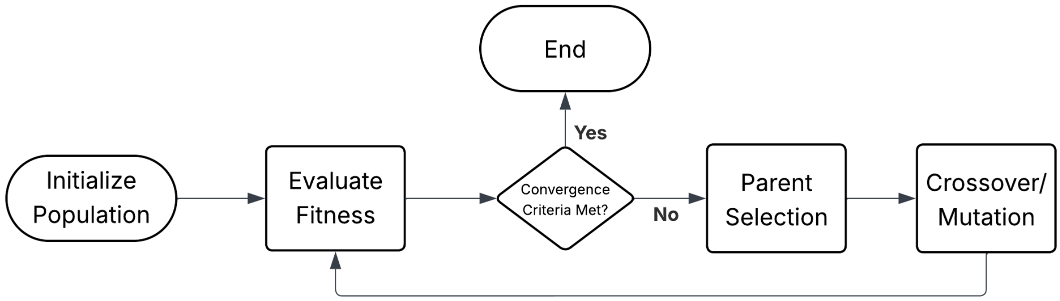


Figure 3.1: *Basic EA Optimization Loop*

3.2.2 Gradient-Based Methods

Gradient-based optimization methods [17] rely on the calculation of the objective function's J derivative, in order to provide a solution which serves as an optimum to the problem. Their deterministic nature means zero sensitivity to the design variables' initialization, which ensures total reproducibility and achievement of an optimum. While they are very computationally efficient and converge rapidly, they depend on the local slope of the loss function and are, thus, inherently "local" in nature. This means that, while they always find an optimum, it is often a local one. The Continuous Adjoint Method solves a set of dual equations — the Field Adjoint Equations (FAE) — which provide the Sensitivity Derivatives (SDs) of the objective function to all design variables simultaneously with a cost roughly equivalent to a single additional flow solution [18]. This means that the computational cost remains independent of the number of design variables, which is the biggest advantage of the adjoint optimization method. The core idea lies in the introduction of a set of adjoint multipliers (Lagrange multipliers), Ψ, because the primal equations' residuals are zero. In the continuous adjoint approach, the adjoint equations (FAE) and adjoint boundary conditions (ABC) are derived analytically from the governing equations (Navier-Stokes) before being spatially discretized. By solving the resulting linear system for Ψ, the gradient can be obtained via a single additional simulation. This renders the adjoint method particularly efficient for large-scale shape optimization problems. This approach offers the advantage of physical interpretability of the adjoint fields and lower memory requirements compared to discrete adjoint methods.

This thesis uses the *adjointOptimisationFoam* toolbox within the OpenFOAM environment to implement the Continuous Adjoint Method for Shape Optimization (ShpO) purposes, employing the SR-based surrogate turbulence models developed. The library was developed in the PCOpt of the NTUA [33].

Chapter 4

Symbolic Regression

Symbolic Regression (SR) refers to a supervised Machine Learning (ML) method and Genetic Programming application [22], which consists of the search for an analytical expression for a certain output Y given a set of inputs $\vec{X}$. The fact that it produces human-interpretable expressions leads to its description as Interpretable Machine Learning. The goal of the method is twofold, as it lies within minimizing both the prediction error as well as the derived equation's complexity. During training, the optimal combination of inputs and algebraic operators is chosen. Its practicality in science and especially in CFD-Based Optimization can be easily imagined, considering the fact that a large number of related physical problems are not governed by clear and well-constructed mathematical expressions, while depending on multiple quantities and conditions. This results in uncertainty concerning the choice of design variables of an optimization problem. In these cases, SR holds the potential to become a very useful tool to make such dependencies interpretable, contributing in the much easier handling of the problem.

The most commonly used tools in SR are Evolutionary Algorithms (EA) and Deep Learning (DL) [34]. This diploma thesis mainly uses PySR, an SR tool that uses an Evolutionary Algorithm to extract the optimal algebraic expression. It is used extensively in order to model both the transition intermittency γ and the turbulent eddy viscosity ν_t, providing analytical expressions and replacing the transition and turbulence models respectively.

4.1 The PySR Algorithm

PySR [9] is an open-source Python and Julia Scientific Symbolic Regression library. It uses multiple independent populations, meaning easy distribution and parallelization across multiple cores, and an *evolve-simplify-optimize* loop, which modifies the typical EA to handle mathematical expressions. The algorithm uses tournament selection for the individual selection and the mutation and crossover operators for new individual creation.
During the search, an "expression-tree", like the one in Fig. 4.1, is constructed constrained by the maximum complexity chosen. This tree contains nodes, which hold either a mathematical operator or a constant. These are the design variables. The complexity of a given expression is equal to the number of nodes in an expression-tree.

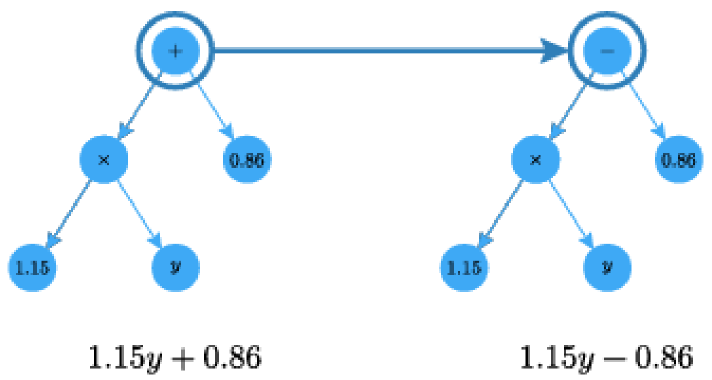


Figure 4.1: *Mutation applied to an expression tree in the top node. The addition "+" operator is mutated into the subtraction "-" operator [9].*

4.1.1 Algorithm Description

The core-algorithm is a typical EA, however some notable modifications are implemented. Firstly, as mentioned before, the search is distributed across multiple "islands" of individuals, which "migrate" asynchronously, in view of the algorithm's parallelization, as illustrated in Fig. 4.3. This means that several separate smaller search spaces with independent populations are created and the fittest individuals from each population are exchanged among them, resulting in much faster conver-

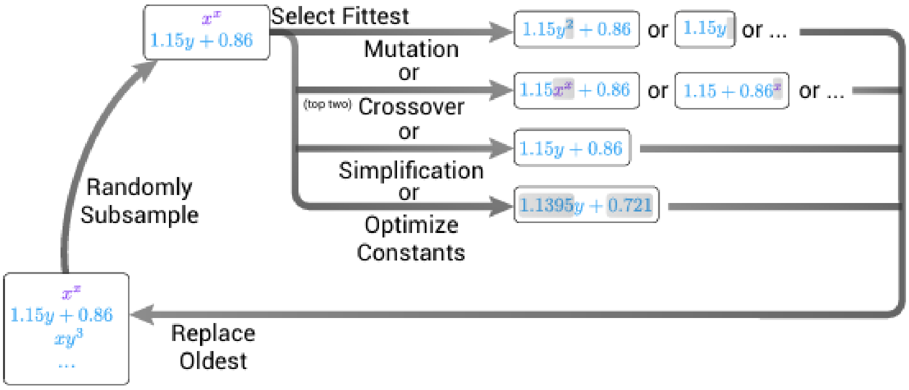


Figure 4.2: *PySR Inner Optimization Loop [9].*

gence. Secondly the algorithm implements "age-regularized" evolution, as it replaces in each generation the eldest individual, rather than the one with the lowest fitness score, in order to prevent early convergence due to finding a local minimum. Thirdly, a simulated annealing-based scheme is used in the individual selection, in view of a faster search procedure. The temperature $T \in [0,1]$ determines the rejection probability of a mutation, which relates to the change in fitness after the possible mutation. The equation that describes this probability is the following:

$$p = exp(\frac{L_F - L_E}{aT}) \tag{4.1}$$

where L_F and L_T the fitness of the mutated and original individuals respectively and a a hyperparameter. When the algorithm is in a high temperature phase, the denominator within the exponential increases, leading the mutation probability p to approach 1 and thus, diversity is encouraged. The hyperparameter a controls the importance of the temperature in the search process. If $a \to 0$, all mutations that result in lower fitness are rejected. Thirdly, the *evolve-simplify-optimize* loop is used. "Evolve" refers to the entire evolution of a population. The "Simplify" step is the creation of an equivalent but simpler form of the equation. The "Optimize" step refers to a few optimization cycles only for the constants using a classical algorithm (BFGS by default).

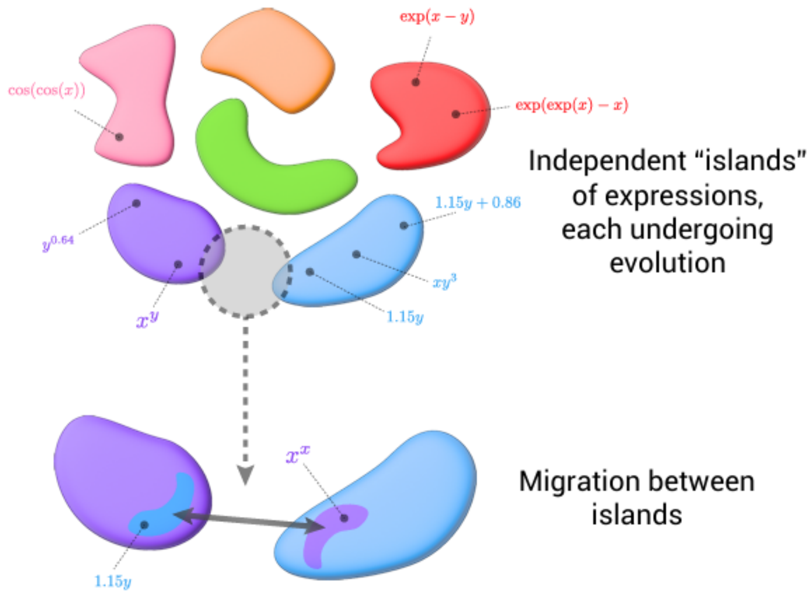


Figure 4.3: *PySR Outer Loop: Migration Between Populations. In each "island" the inner loop is iterated and the populations migrate between islands regularly [9].*

4.1.2 Loss Function

The prediction loss function is the Mean Squared Error (MSE) between the true and predicted output values by default, as described in Eq. 4.2. However, the user also has the option to define either a weighted MSE, or a completely different custom-made loss function.

$$\ell_{pred}(E) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.2}$$

with n the output data size.

An adaptive parsimony metric is used in the final loss function, which is described by Eq. 4.3, in order to account for the need to minimize the expression's complexity apart from maximizing its accuracy.

$$\ell(E) = \ell_{pred}(E) + (parsimony)C(E) \tag{4.3}$$

with $C(E)$ the complexity of expression E and ℓ_{pred} the predictive loss. PySR uses an adaptive penalty to ensure that the search does not become biased toward a specific complexity level and that each complexity corresponds to approximately the same number of expressions. The final PySR loss function is the following:

$$\ell(E) = \ell_{pred}(E) \cdot exp(frecency[C(E)]) \tag{4.4}$$

with ”frecency” being a combined measure of the frequency and recency of appearance.
Finally, a variety of additional features are available on PySR, such as tools for denoising data, adding weights to certain inputs, introducing a custom loss function or even custom operators.

4.2 Workflow of the Regression

The implementation follows a structured process to derive the symbolic expression:

1. **Data Acquisition and Variable Selection:** The code reads the to-be-fitted data and the user defines the dependent (output) and independent (input) variables.

2. **Data Normalization:** In most cases it is considered best practice to normalize the variables, in order to ensure that the order of magnitude does not give an advantage to any variable and that the algebraic expression can sufficiently be generalized.

3. **EA Parameter Definition:** The user establishes the EA parameters (number of populations, population size, selection model).

4. **Model Configuration:** The regression model is built by defining the maximum depth and complexity of the derived expression, the loss function which the algorithm seeks to minimize, the allowed mathematical operators for the expression tree and the number of iterations.

4.3 A simple PySR example

The process of uncovering an algebraic expression is presented through a simple example, in order to render the algorithm and its interface more familiar. In this context, the target function is described in Eq. 4.5 and the corresponding expression tree that the algorithm seeks to uncover is illustrated in Fig. 4.4.

$$y = 3\sin(x_3)e^{x_1} + x_2^2 - 0.5 + \cos(x_0) \tag{4.5}$$

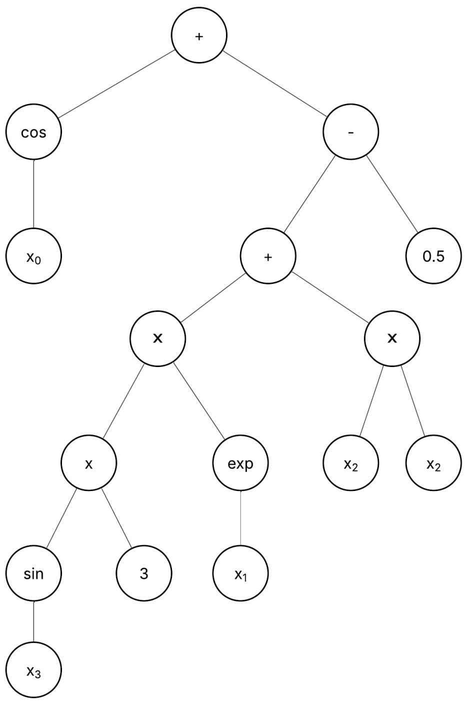


Figure 4.4: *Expression tree corresponding to Eq. 4.5. It contains 16 nodes, which define the equation's complexity.*

The input data matrix $X \in \mathbb{R}^{100\times 5}$ is generated so that it follows a normal distribution, through the *numpy* library in Python. The output data is generated by using Eq. 4.5 on the input data. Once the input and output data is generated, the EA and regressor model parameters are defined, as illustrated in Table 4.1. The population size is set to the moderate value of 60, while 30 separate populations are used, which quickly evolve to local solutions and periodically migrate to lead to the optimal expression. The selection process follows the parsimony principle, aiming to minimizing both the loss function and the expression's complexity. To ensure the resulting expressions remain easily interpretable, the maximum complexity was set to 30 and the maximum depth to 5, in order to prevent deep nesting within the function. The selected loss function is the Mean Squared Error between the predicted value $\hat{y}$ and the true value y, as described in Eq. 4.2, with $n = 100$ the input data size. The algorithm is set to run for 150 optimization cycles, a number which allows it to reach a stable convergence without unnecessarily increased computational cost.

EA Parameter	Value
Population Size	60
Populations	30
Selection Criteria	Lowest MSE & Complexity
Maximum Complexity	30
Maximum Depth	5
Binary Operators	$+, -, *, pow$
Unary Operators	exp, sin, cos, log

Table 4.1: *PySR Example: Evolutionary Algorithm Parameters*

Complexity	**Loss**	**Score**	**Equation**
1	2.796e+01	1.594e+01	$y = x_3$
2	2.523e+01	1.030e-01	$y = \exp(x_3)$
3	2.478e+01	1.771e-02	$y = 3.358^{x_3}$
4	1.492e+01	5.074e-01	$y = x_3 \cdot \exp(x_1)$
5	5.842e+00	9.377e-01	$y = (4.1272^{x_1}) \cdot x_3$
6	5.662e+00	3.135e-02	$y = (x_3 \cdot 2.473) \cdot \exp(x_1)$
7	4.279e+00	2.800e-01	$y = \exp(x_1 + 1.1194) \cdot \sin(x_3)$
8	3.797e+00	1.194e-01	$y = (x_3 \cdot \exp(x_1 + 0.90568)) + 1.3654$
9	1.924e+00	6.799e-01	$y = x_2 \cdot x_2 + (4.078^{x_1} \cdot x_3)$
10	1.162e+00	5.040e-01	$y = x_2 \cdot x_2 + (x_3 \cdot \exp(x_1 + 0.89728))$
11	1.983e-01	1.768e+00	$y = x_2 \cdot x_2 + (\exp(x_1 + 1.1039) \cdot \sin(x_3))$
13	1.751e-01	6.230e-02	$y = (\sin(x_3) \cdot \exp(x_1 + 1.1035)) - (-0.15243 - x_2 \cdot x_2)$
15	1.724e-01	7.623e-03	$y = ((-0.0080588 - \sin(x_3)) \cdot (\exp(x_1) \cdot -3.0016)) - (-0.11157 - x_2 \cdot x_2)$
16	**1.306e-13**	**1.594e+01**	$\boldsymbol{y = (\exp(x_1) \cdot 3 \sin(x_3)) - (0.5 - (\cos(x_0) + x_2 \cdot x_2))}$
18	9.857e-14	1.406e-01	$y = ((\exp(x_1) \cdot (-3)) \cdot \sin(1.509\text{e-}08 - x_3)) - (0.5 - (\cos(x_0) + x_2 \cdot x_2))$
20	7.485e-14	1.376e-01	$y = ((\exp(x_1) \cdot (-3)) \cdot ((x_1 \cdot 1.331\text{e-}08) - \sin(x_3))) - (0.5 - (\cos(x_0) + x_2 \cdot x_2))$

Table 4.2: *PySR output: Complexity, Loss, Score, and Symbolic Equations*

The algorithm was able to retrieve the exact equation within the 150 iterations, as illustrated in Table 4.2. This is the output table, which PySR provides at the end of every search. It contains a number of proposed equations with increasing complexity, along with their corresponding loss and score. A high score indicates a significant reduction in error for a marginal increase in complexity, as described in Eq. 4.6.

$$S(c_i) = -\frac{\ln(L_i) - \ln(L_{i-1})}{c_i - c_{i-1}} \tag{4.6}$$

where:

- L_i and c_i represent the loss (MSE) and the complexity of the current candidate model, respectively.
- L_{i-1} and c_{i-1} represent the loss and complexity of the preceding model on the Pareto-optimal front.

Chapter 5

Applications of SR in Transition Modeling

The transition intermittency factor γ of the $\gamma - Re_\theta$ model determines the laminar-to-turbulent flow transition, as stated before. The creation of an algebraic expression to describe a PDE-based model, if its prediction accuracy is satisfactory, leads to a reduction in the computational cost. In order to derive this algebraic expression, a CFD computation of the flow is performed using PUMA and the flow data (nodal coordinates, velocity, pressure, etc.) become available. It is then used offline as input to the SR regressor, from which the final expression for the intermittency factor is derived.

The drag and lift coefficients C_D, C_L of the airfoil are used as evaluation criteria. The accuracy of the transition point prediction is tested through the visualisation of the intermittency fields, as well as the skin friction coefficient C_f distribtion on each case. The skin friction coefficient presents a ”spike” at the transition point, indicating that the flow has become fully turbulent, due to the increased momentum exchange between the outer flow and the wall.

The cases presented in the next sections are the following:

- **Case A - NLF(1)-0416 Airfoil** designed by NASA for natural laminar flow. It was chosen as the first incompressible case for this thesis because it is one of the most widely used cases for studying transition.

- **Case B - ONERA OAT15A Airfoil** designed for transonic flow. The observation of a transonic buffet phenomenon, associated with the shock-boundary layer interaction makes it a useful candidate for the testing of the algebraic surrogate to the transition model.

- **Case C - NLF(1)-0416 & OAT15A Airfoils** used together as input data for the derivation of a single, more generalized algebraic expression.

5.1 Case A: NLF(1)-0416 Airfoil

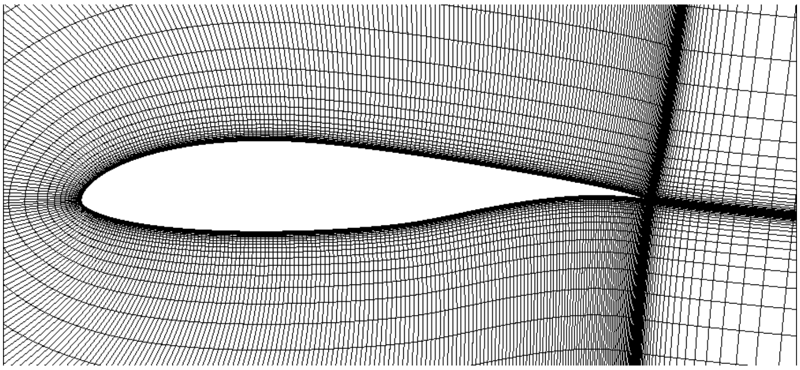
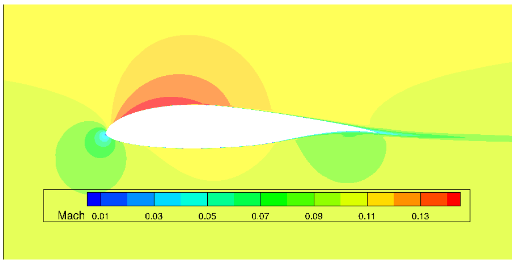
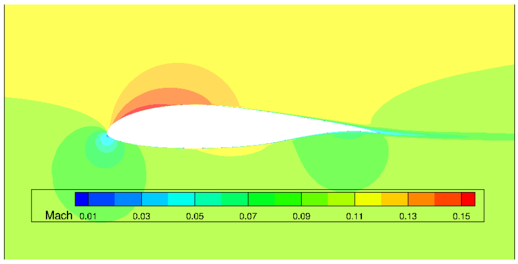


Figure 5.1: *C-Type Grid for the NLF(1)-0416 Airfoil consisting of 77086 nodes.*

The NLF(1)-0416 Airfoil [44] is a low-speed Natural Laminar Flow Airfoil, developed by NASA, designed specifically to maintain a laminar boundary layer over a significant portion of both the upper and lower surfaces (approximately 40% of the chord length) without the need for active suction systems. Its geometric characteristics are described in Table 5.1. The PUMA simulations conducted in this thesis are at $M = 0.1$ and different angles of attack (AoA) (0°, 2.03°, 3°, 3.5°), as shown in Fig. 5.3, and will be used to assess the accuracy of the surrogate models on various conditions. The C-type structured grid -which was, though, treated by the PUMA code, as unstructured- consists of 77086 nodes (Fig. 5.1). The drag and lift coefficients from the CFD-simulation, as well as the experimental results [44] are summarized in Table 5.2. Naturally, they are both increasing as the AoA increases. In the RANS simulations, the turbulence model used is SA and the transition model is the Piotrowski-Zingg variation of the $\gamma - Re_\theta$, which were both described in Chapter 2.



(a) Mach Field at AoA = 2.03°.



(b) Mach Field at AoA = 3.50°.

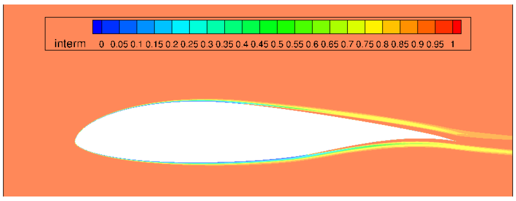
Figure 5.2: *Flow around the NLF(1)-0416 Airfoil at AoA* = 2.03° *&* 3.50°.

Geometric Characteristic	Symbol	Value
Chord Length	c	0.60902 m
Thickness to Chord Ratio	t/c	16%

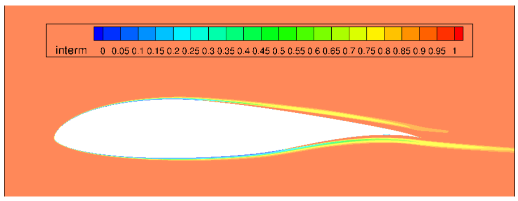
Table 5.1: *NLF(1)-0416: Geometric Characteristics.*

AoA(°) Data	$C_D(-)$ Experimental	$C_D(-)$ PUMA	$C_L(-)$ Experimental	$C_L(-)$ PUMA
0.00	0.0061	0.0057	0.4637	0.4835
2.03	0.0063	0.0063	0.7038	0.7200
3.00	0.0065	0.0068	0.8108	0.8308
3.50	0.0068	0.0070	0.8661	0.8878

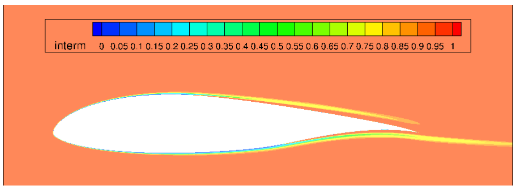
Table 5.2: *NLF(1)-0416 Airfoil: Reference Drag (C_D) and Lift (C_L) coefficients in different AoA. The experimental results are by [44]. The PUMA solver employs SA and Piotrowski-Zingg for turbulence and transition modeling respectively.*



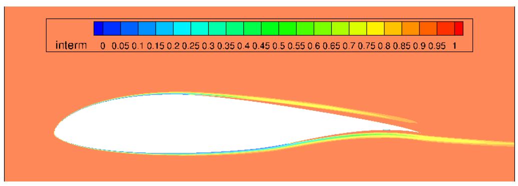
(a) Intermittency Field at AoA = 0°.



(b) Intermittency Field at AoA = 2.03°.



(c) Intermittency Field at AoA = 3°.



(d) Intermittency Field at AoA = 3.50°.

Figure 5.3: *Intermittency fields from PUMA using the Piotrowski-Zingg transition model at different AoA.*

5.1.1 SR-based Algebraic Transition Model

The data derived from the simulation -only for a specific AoA each time- is normalized post-process and fed into the SR as input, as presented in Table 5.4. The output values are not normalized, as the intermittency factor is inherently non-dimensional and constrained within the range [0,1]. The SR output is always an algebraic expression for the transition intermittency γ. The Mean Squared Error (MSE) of the intermittency is chosen as the SR Loss Function, as follows:

$$F_{loss} = MSE = \frac{1}{n}\sum_{i=1}^{n}(\gamma_i - \hat{\gamma}_i)^2 \tag{5.1}$$

where $n = 77086$ is the number of grid nodes.

The chosen EA parameters for the Regressor are summarized in Table 5.3. The population size and the number of populations are set to 50 and 30 respectively, values which proved to achieve a fast convergence. The maximum depth and complexity are 6 and 30 respectively, allowing the algorithm to yield a relatively complex expression in order to best fit the intermittency field to local flow and geometric quantities. Finally, the allowed mathematical operators contain trigonometric, exponential, logarithmic, root and power functions, so that the Regressor's search space is not too restricted for such a demanding task.

EA Parameter	Value
Population Size	50
Populations	30
Selection Criteria	Lowest MSE & Complexity
Maximum Complexity	30
Maximum Depth	6
Binary Operators	$+, -, *, /, pow$
Unary Operators	$exp, sin, cos, tan, log, sqrt$

Table 5.3: *SR Training: Evolutionary Algorithm Parameters.*

The intermittency expressions derived from SR are summarized in Table 5.5, along with the variables they contain, the corresponding training loss, and their complexity.

The most important finding is that the vorticity and the turbulent eddy viscosity

SR Input	Normalized Quantity	Expression
c_0	Coordinate x	x/c
c_1	Wall Distance	d_{wall}/c
c_2	Density	ρ/ρ_∞
c_3	Velocity x	V_x/V_∞
c_4	Velocity y	V_y/V_∞
c_5	Pressure	p/p_∞
c_6	Vorticity	$\Omega c/V_\infty$
c_7	Turbulent Eddy Viscosity	$\nu_t/V_\infty c$
c_8	Local Mach Number	M
c_9	Coordinate y	y/c

Table 5.4: *SR Training: The 10 Normalized Input Variables.*

Expression	Training AoA(°)	Variables	Complexity	Loss
A	2.03	$c_0, c_3, c_4, c_5, c_6, c_7$	30	0.001599
B	3.50	c_1, c_6, c_7, c_8	21	0.001516
C	3.50	c_0, c_3, c_6, c_7	28	0.002655

Table 5.5: *SR Output: Intermittency Expressions, described by the input data AoA, the independent variables contained in the final expression, the complexity and the loss of each expression.*

are always part of the output equation, as they are quantities inherently related to the boundary layer and, thus, provide valuable information concerning the flow's transition. The normalized horizontal position and velocities also appear in the equations, signifying the impact of the geometry and the inflow conditions on transition modeling.

Of the three outputs, the one with the lowest training loss and simultaneously the lowest complexity is Expression B, which is the following:

$$\gamma = 0.8074^{\phi \cdot \tau^{(10.3013 \cdot \kappa \cdot \lambda)}} \tag{5.2}$$

With $\phi = c_6^{0.6444}$, $\lambda = c_7^{0.3227}$, $\kappa = c_6^{0.1337}$, $\tau = c_8^2 \cdot c_1$ and as stated before $c_1 = d_{wall}/c$, $c_6 = \Omega c/V_\infty$, $c_7 = \nu_t/V_\infty c$, $c_8 = M$.

The expression is integrated in the PUMA solver and replaces the Transition Model in order to compare the flowfield with the reference data and evaluate its intermittency prediction accuracy, as well as the way that it interferes with the rest of the field quantities. It is tested in all four angles of attack, which were mentioned before.

Tested AoA(°)	C_D value (% error)	C_L value (% error)
0.00	0.0038 (-33.23)	0.5094 (5.37)
2.03	0.0042 (-33.30)	0.7543 (4.77)
3.00	0.0067 (-0.50)	0.8315 (0.08)
3.50	0.0069 (-0.70)	0.8888 (0.11)

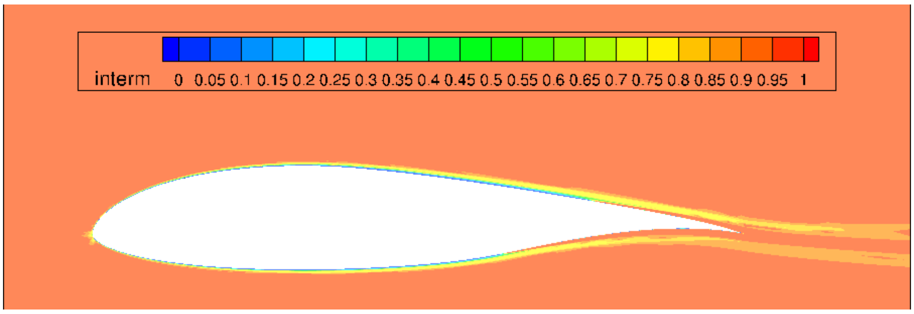
Table 5.6: *Equation 5.2 - Trained in AoA = 3.50°: Drag and Lift Coefficients Relative Error in different AoA.*

As shown by the results in Table 5.6, Eq. 5.2 provides a promising prediction for the lift and drag coefficients for AoA close to the training AoA (3.5°). However, while the AoA decreases, the relative error increases significantly and no satisfactory prediction is acquired. The fact that the algebraic model always underestimates the drag coefficient and overestimates the lift coefficient is also noteworthy.
The accuracy of the transition point's prediction is evaluated through the skin friction coefficient's C_f distribution. Fig. 5.5 shows that the algebraic model achieves a very good prediction of the transition point for the AoA close to the training angle, results that align with the drag and lift coefficients prediction.

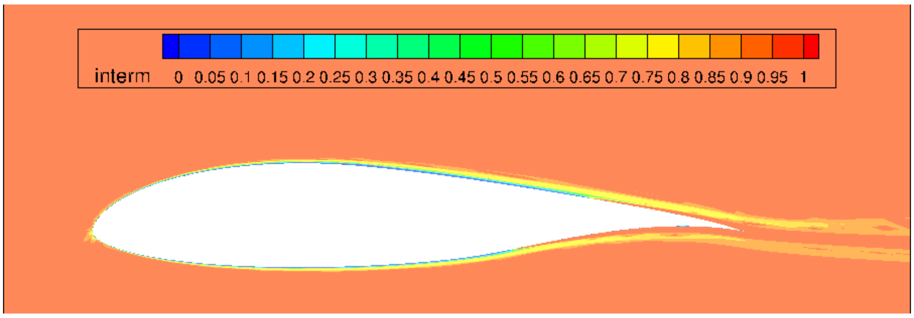
For the smaller angles of attack, where the relative drag and lift coefficient error is significantly higher, a correction loop is constructed. The reference (Transition Model) intermittency field and the intermittency field constructed by the algebraic model are subtracted node-wise, resulting in a new field that describes the intermittency difference in each node. This field is then fed as input to a new round of SR training, from which an expression for this error is obtained. This expression is then superimposed over Eq. 5.2 and the simulation is once again executed using PUMA with the corrected algebraic transition model. This correction can be performed for multiple rounds until convergence. The results of the correction are presented in Table 5.7 and visual representation of the field at AoA = 2.03° is shown in Fig. 5.6.

Correction Round	C_D value (% error)	C_L value (% error)
Round 0	0.0042 (-33.30)	0.7543 (4.77)
Round 1	0.0066 (5.23)	0.7167 (-0.45)
Round 2	0.0060 (-5.47)	0.7220 (0.28)

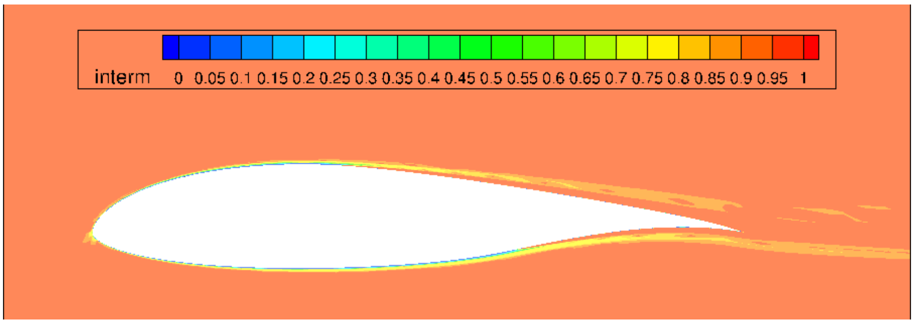
Table 5.7: *Eq. 5.2: Drag and Lift Coefficients Values Relative Error to RANS Simulations after Proposed Correction Loop at AoA=2.03°.*



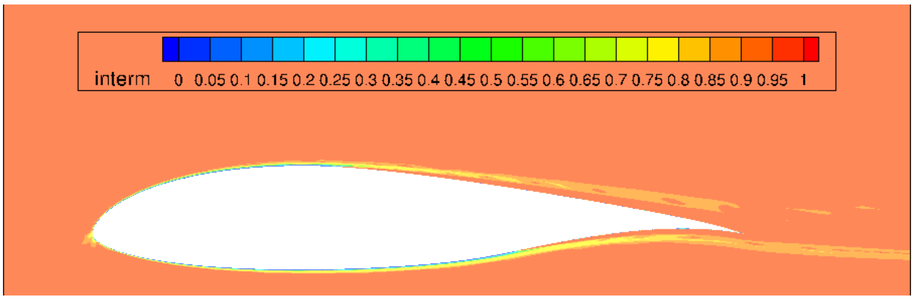
(a) Intermittency Field at AoA = 0°.



(b) Intermittency Field at AoA = 2.03°.

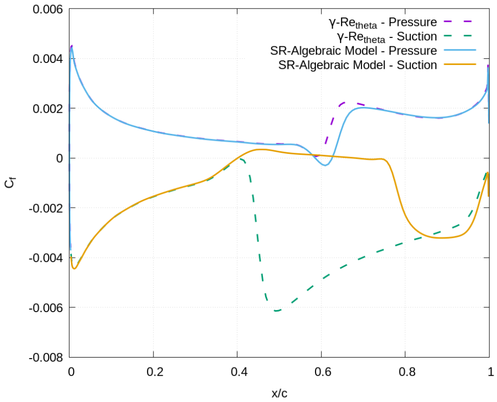


(c) Intermittency Field at AoA = 3°.

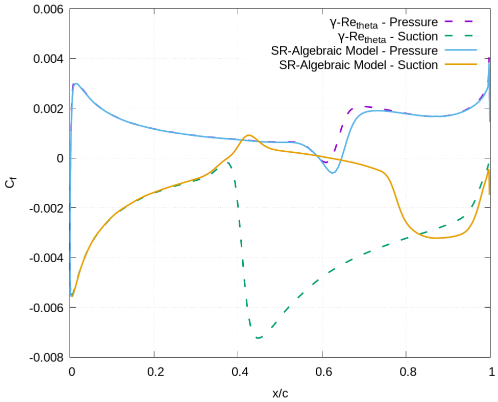


(d) Intermittency Field at AoA = 3.50°.

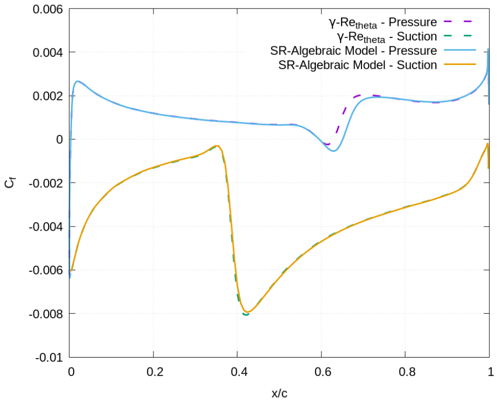
Figure 5.4: *Intermittency fields using the SR-based Eq.5.2 as a surrogate to the transition model at different AoA.*



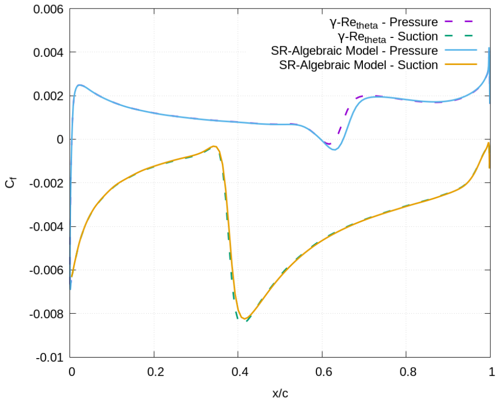
(a) AoA = 0.00°.



(b) AoA = 2.03°.



(c) AoA = 3.00°.



(d) AoA = 3.50°.

Figure 5.5: *NLF(1)-0416: Skin friction coefficient (C_f) distributions for the NLF(1)-0416 airfoil using Eq.5.2.*

The testing process of Equation 5.2, which was the output of a SR process with input data from a certain AoA, provided an important finding. While this algebraic surrogate is able to predict the lift and drag coefficients with sufficient accuracy in angles of attack relatively close to the training AoA, it fails to predict a field with inflow conditions further away from the training sample. This outcome, while expected, as the algebraic expression contains field quantities which change drastically as the flow around the airfoil changes, leads to the conclusion that such a model cannot be used in any case other than in a similar geometry with very similar inflow conditions. Hence, the following training uses fields from two different inflow angles of attack $(2.03°, 3.5°)$ as input, to examine the ability of the new algebraic model to accurately predict fields from a wider range of inflow angles of attack.

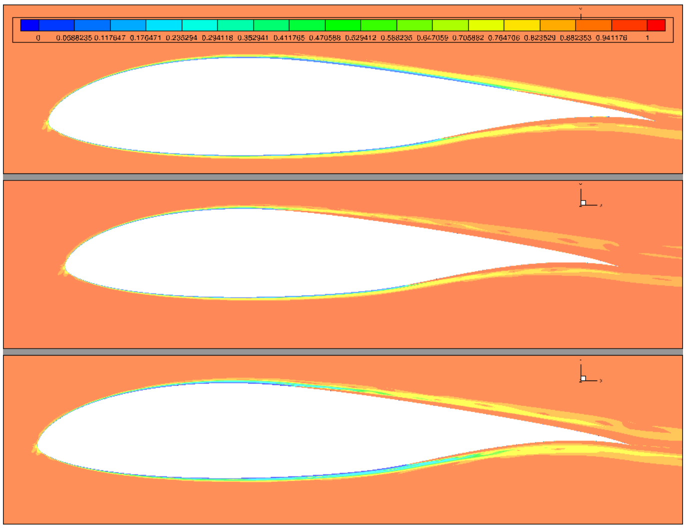


Figure 5.6: *Intermittency Field: Two Rounds of the Correction Loop for Eq. 5.2 at AoA=2.03°. Top Field: Produced with Eq. 5.2, Middle Field: Produced after the first correction round, Bottom Field: Produced after the second correction round.*

$$\gamma = \left(c_7^{c_2-\Phi}\right)^{\Psi} \tag{5.3}$$

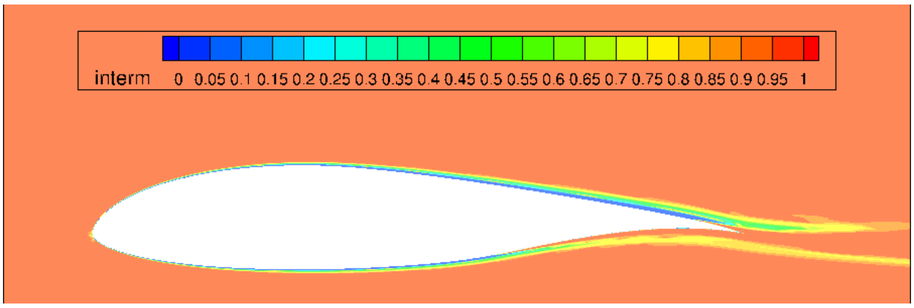
with $\Phi = (0.9416^{c_6})^{(0.1721+c_0)}$, $\Psi = \left[\left(\frac{c_7}{c_6}\right)^{c_7^{0.2657}}\right]^{\exp(\cos c_4+\cos c_8)}$ and $c_0 = x/c$, $c_2 = \rho/\rho_\infty$, $c_4 = V_y/V_\infty$, $c_6 = \Omega c/V_\infty$, $c_7 = \nu_t/V_\infty c$, $c_8 = M$.

The resulting Equation 5.3 is summarized in Table 5.8.

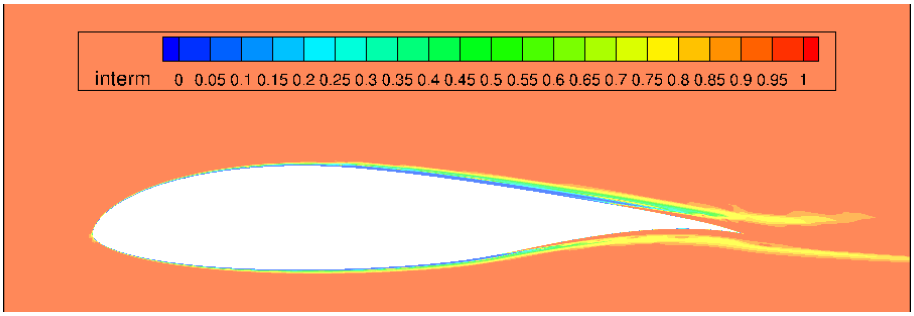
Equation	Training AoA(°)	Variables	Complexity	Loss
5.3	2.03 & 3.50	$c_0, c_2, c_4, c_6, c_7, c_8$	26	0.00261

Table 5.8: *SR Output: Intermittency Equation from two Fields*

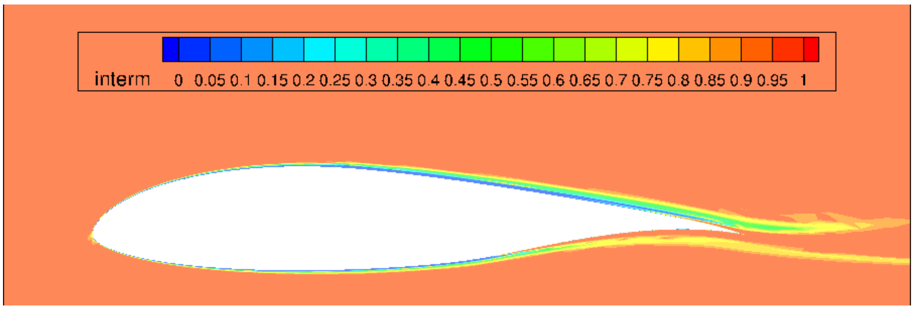
The loss (MSE) of Equation 5.3 is comparable to that of Equation C 5.5, although its complexity is lower. It is noteworthy that this equation, while still containing the normalized x coordinate, the vorticity and the turbulent eddy viscosity, also contains the local Mach number and the vertical component of the velocity. This serves the purpose of determining the change in the flowfield due to the change in the angle of attack. The new equation is tested on all four angles of attack that were simulated earlier and the results are shown in Table 5.9.



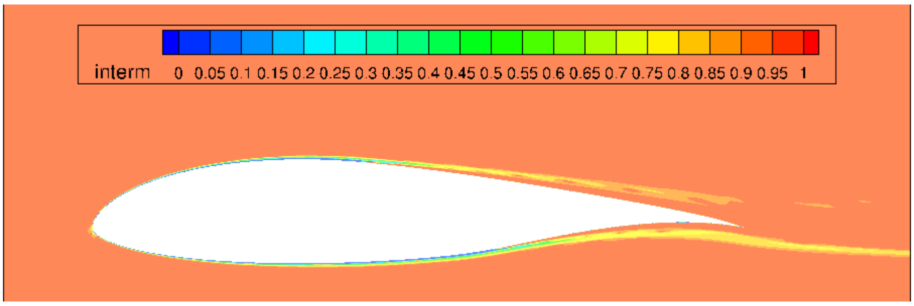
(a) Intermittency Field at AoA = 0°.



(b) Intermittency Field at AoA = 2.03°.



(c) Intermittency Field at AoA = 3°.



(d) Intermittency Field at AoA = 3.50°.

Figure 5.7: *Intermittency fields using the SR-based Eq.5.3 as a surrogate to the transition model at different AoA.*

Tested AoA(°)	C_D value (% error)	C_L value (% error)
0.00	0.0037 (-35.03)	0.4944 (2.27)
2.03	0.0049 (-22.75)	0.7063 (-1.90)
3.00	0.0043 (-37.00)	0.8703 (4.96)
3.50	0.0063 (-9.31)	0.8912 (0.38)

Table 5.9: *Equation 5.3: Drag and Lift Coefficients Relative Error in different AoA.*

From Tables 5.6 and 5.9 it becomes obvious that Eq. 5.2 is far more successful in predicting the C_L and C_D coefficients when the inflow AoA remains close to 3.5° but loses accuracy quickly when used for lower angles of attack. On the contrary, Equation 5.3 provides a more consistent but certainly lower accuracy in a wider range of AoA. This is a compromise made in view of the model's generalization in somewhat different inflow conditions.

5.2 Case B: ONERA OAT15A Airfoil

The procedure that was previously followed is now tested on the ONERA OAT15A airfoil. It is chosen as the next test case, because of the buffet phenomenon that occurs under transonic conditions. This refers to the interaction of the turbulent boundary layer and the occurring shock wave, which results in a self-sustained, low-frequency oscillation, causing instability. The buffeting is observed in the pressure field around the airfoil in Fig. 5.9. ONERA has performed experiments in their S3 wind tunnel (0.8×0.76 m^2).

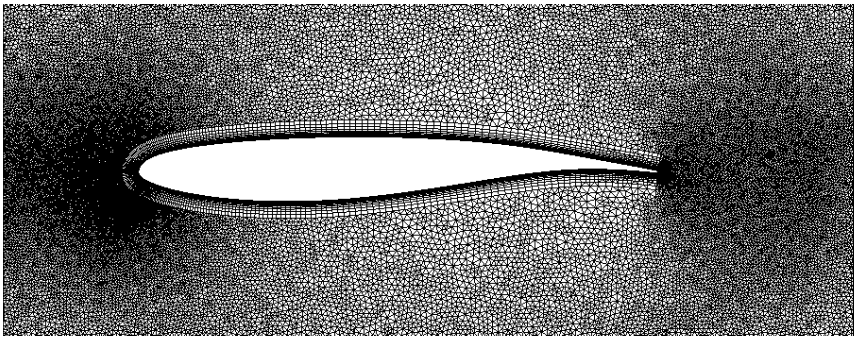


Figure 5.8: *O-Type Hybrid Grid for the OAT15A Airfoil.*

The experimentally tested airfoil has the following characteristics:

The experimental and computational results show that when exposed to an inflow

Geometric Characteristic	Symbol	Value
Chord Length	c	0.230 m
Thickness to Chord Ratio	t/c	12.5%
TE Thickness	t_{TE}	$0.5\%c$

Table 5.10: *ONERA OAT15A: Geometric Characteristics.*

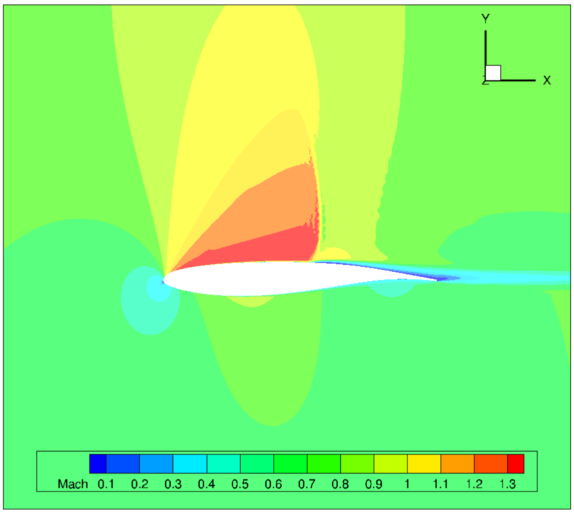
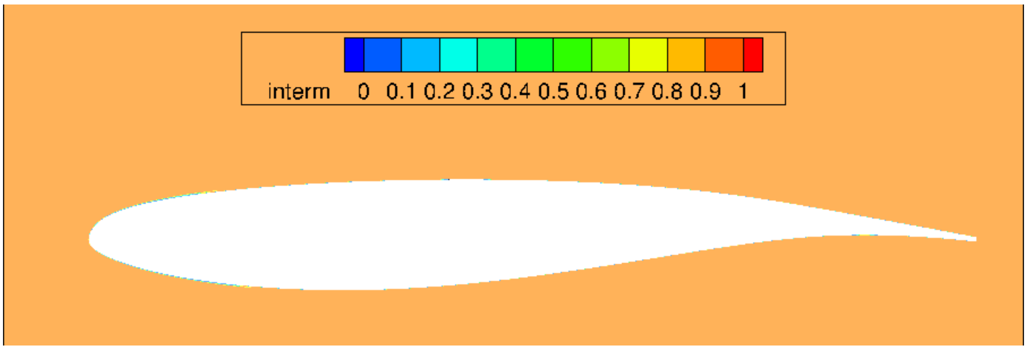


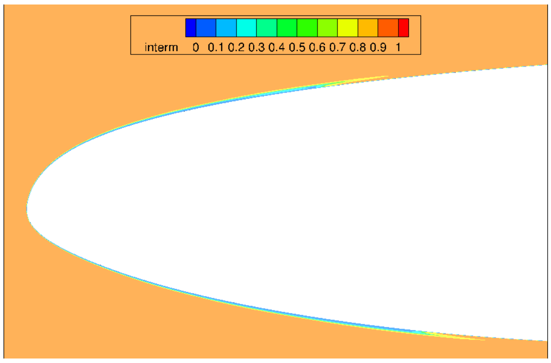
Figure 5.9: *ONERA OAT15A Airfoil: Mach number around the airfoil at AoA =* $3.5°$.

with M=0.73, $P_{stat} = 0.78\ kPa$, $T = 271\ K$ and an AoA up to almost 3.5°, the flow can be considered steady. However at 3.5° unsteadiness occurs. The interaction of the airfoil's boundary layer with the occurring shock wave is observed for the first time when the inflow AoA is 3.25°.
The O-Type hybrid triangular-element grid with 254148 nodes used for the simulations conducted in this Thesis has been generated by [3] and is illustrated in 5.8.

The flow around the OAT15A airfoil was once again simulated using PUMA, under inflow conditions of M=0.73, $P_{stat} = 0.78\ kPa$, $T = 271\ K$ and $AoA = 2.00°$, in order to create the input field for the SR training. The reference intermittency field for these inflow conditions can be seen in Fig. 5.10(a). In Fig.5.10(b), the detail of the Leading Edge area and the transitional region is shown.



(a) Intermittency Field at AoA = 2.00°



(b) Detail: Leading Edge Area

Figure 5.10: *ONERA15A Airfoil: Reference Intermittency Field and Leading Edge Detail, using the Piotrowski-Zingg transition model within PUMA.*

5.2.1 SR-based Algebraic Transition Model

The flow data acquired from the simulation are used for training in SR, with the input quantities being the ones described in Table 5.4 and the EA settings the ones in Table 5.3. The algorithm produces the following expression for the intermittency γ:

$$\gamma = \sqrt{\theta} \tag{5.4}$$

with $\kappa = \frac{2.8952558}{c_6+2.4448037}$, $\psi = c_7^{c_7}$, $\phi = \psi^{(c_5 \cdot 305315.44)}$, $\delta = \kappa^{\phi}$, $\lambda = \frac{c_2^{1.3415653}}{e^{c_0}}$, $\zeta = (c_0 \cdot c_3) \cdot 3.7659185$, $\eta = \lambda^{\zeta}$, $\theta = \delta^{\eta}$ and $c_0 = x/c$, $c_3 = V_x/V_\infty$, $c_5 = p/p_\infty$, $c_6 = \Omega c/V_\infty$, $c_7 = \nu_t/V_\infty c$.

Quantities appearing in Eq. 5.4 are once again the turbulent eddy viscosity ν_t and the vorticity Ω, as well as the relative horizontal position along with two flow

Equation	Training AoA(°)	Variables	Complexity	Loss
5.4	2.00	c_0, c_3, c_5, c_6, c_7	27	0.00046

Table 5.11: *SR Output: Intermittency Equation from OAT15A Field.*

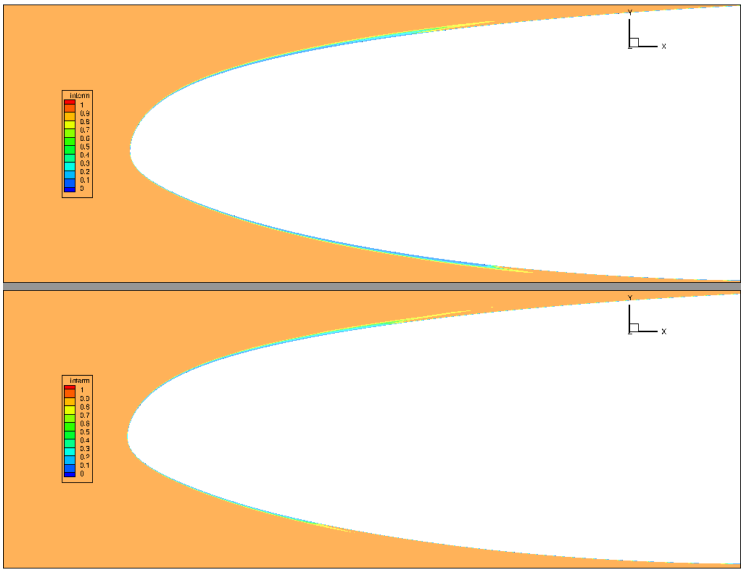
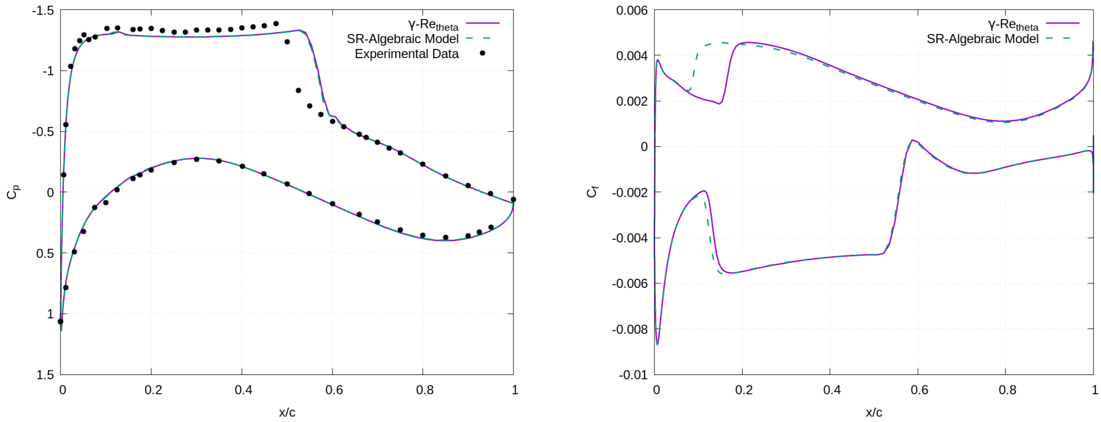


Figure 5.11: *LE Area detail of Intermittency Field at AoA = 2.50°. Top Field: Produced with the Piotrowski-Zingg transition model, Bottom Field: Produced using Eq. 5.4 as a surrogate to the transition model within PUMA.*

quantities, the local pressure and the horizontal velocity.

The new Eq. 5.4 is subsequently integrated into the PUMA solver in order to produce a new field. It is tested with AoA ranging from 1 to 3.50°. The resulting friction coefficient C_f distribution in comparison with the reference distribution can be seen in Fig. 5.12(b) and the corresponding C_D and C_L in Fig. 5.13. The new algebraic model seems to perform sufficiently in a range of AoA, with the maximum C_D relative error being 1.01%, at AoA=1°. The C_p distribution in every AoA is almost perfectly aligned with the $\gamma - Re_\theta$ distribution, as well as sufficiently aligned with the experimental data, as shown in Fig. 5.12(a) . However, the skin friction coefficient C_f distributions show that the algebraic model consistently underestimates the transition point. This leads to the conclusion, that the small error in C_D and C_L derives from the mismatch in the C_f distributions and the viscous component of the drag force. The underestimation of the transition point can also be observed in Fig. 5.11.



(a) Pressure Coefficient (C_p) (b) Skin friction coefficient (C_f)

Figure 5.12: *Aerodynamic distributions for the ONERA15A airfoil at AoA = 2.50° with Equation 5.4.*

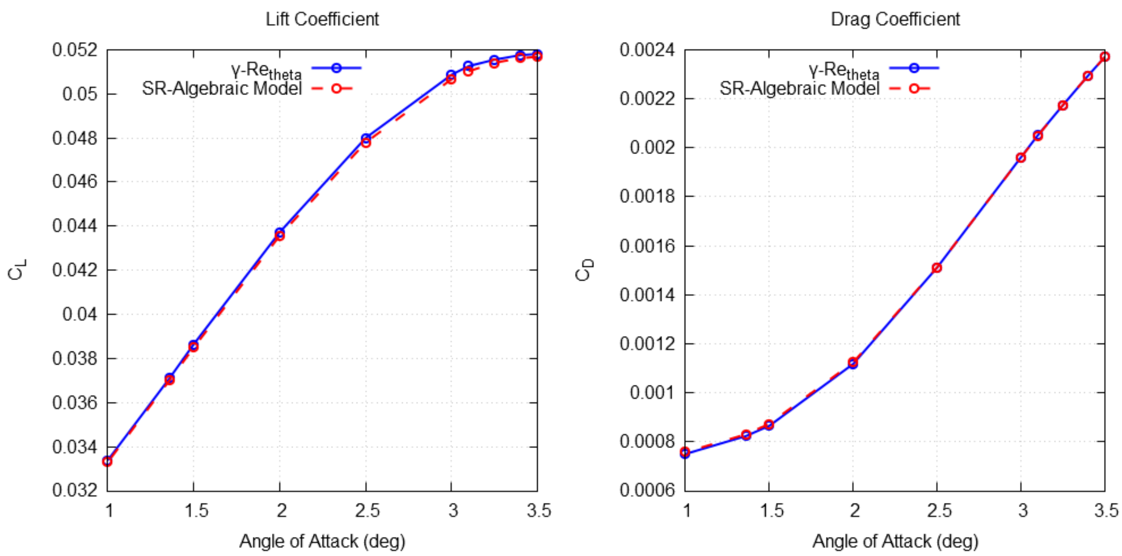


Figure 5.13: *OAT15A Airfoil: Drag and Lift Coefficients in different AoA with Equation 5.4.*

5.3 Case C: Combination of NLF(1)-0416 and ONERA OAT15A Airfoils

The two previous cases provided useful insights to the process of the development of an algebraic surrogate transition model using Symbolic Regression. However, the fact that the input data for the Regressor consisted only of quantities corresponding to the flow around a specific geometry, leads to the conclusion that the produced equations are case-dependent and, thus, cannot be trusted for the modeling of fields with largely varying conditions, around a variety of different geometries. The need for a more generalized model led to a new idea. The SR Regressor was used to yield an expression using data from two fields, one around the NLF(1)-0416 and one around the OAT15A airfoil. The described approach aims at generalizing the transition model by training it with data stemming from two very different fields. The first airfoil is designed for natural laminar flow and the second one is designed for transonic conditions and can exhibit unsteady phenomena, such as buffeting. These differences are crucial and an expression capable of modeling such big transition discrepancies can lead to a very optimistic view of data-driven modeling.

5.3.1 SR-based Algebraic Transition Model

The regression process is the typical one previously described. The SR variables and the EA settings remain the same as in Tables 5.4 and 5.3 accordingly. The only difference is that in this case, the input variables of each airfoil are normalized with the corresponding geometric and far-field data of each separate field. Data from the NLF(1)-0416 and OAT15A airfoils exposed to inflow with AoA = 2° and M=0.1 and M=0.73 accordingly are fed into the SR algorithm and the produced equation is the following:

$$\gamma = \phi^{\psi} \tag{5.5}$$

with $\kappa = c_7^{0.033316202}$, $\phi = \sin(\sin(\kappa)) + 0.41349724$, $\theta = \frac{c_6}{0.22231957}$, $\zeta = 0.45793113 - c_7^{0.06932941}$, $\lambda = \theta^{\zeta}$, $\tau = \frac{c_3}{\cos(c_2)} + 1.8690305$, $\psi = \lambda^{\tau}$ and $c_2 = \rho/\rho_{\infty}$, $c_3 = V_x/V_{\infty}$, $c_6 = \Omega c/V_{\infty}$, $c_7 = \nu_t/V_{\infty}c$.

Equation	Training AoA(°)	Variables	Complexity	Loss
5.5	2.00	c_2, c_3, c_6, c_7	24	0.001225

Table 5.12: *SR Output: Intermittency Equation from NLF(1)-0416 and OAT15A.*

AoA(°)	NLF(1)-0416 C_D (% error)	NLF(1)-0416 C_L (% error)	OAT15A C_D (% error)	OAT15A C_L (% error)
2.00	0.006388 (1.41)	0.715803 (-0.09)	0.001128 (0.89)	0.045651 (4.42)
2.50	0.006965 (7.46)	0.763131 (-1.44)	0.001584 (4.79)	0.050529 (5.29)
3.00	0.006882 (1.72)	0.828059 (-0.33)	0.002117 (7.93)	0.054181 (6.53)

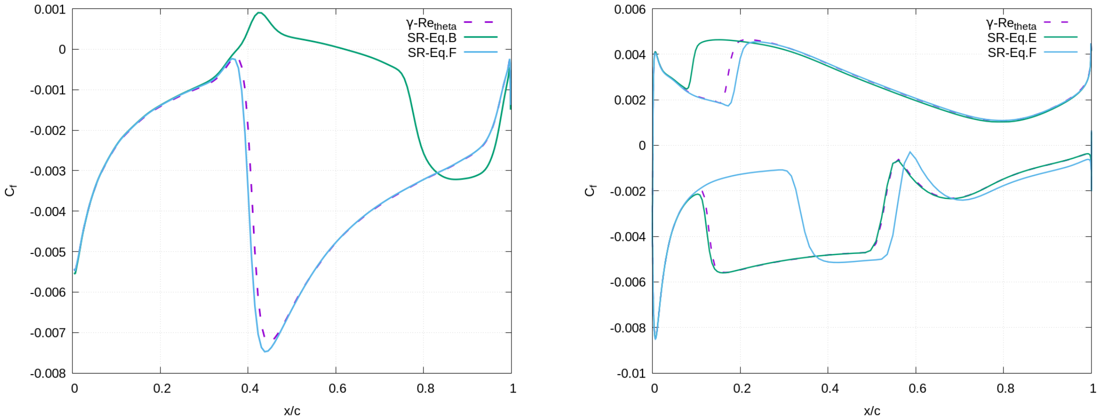
Table 5.13: *Equation 5.5: Drag and Lift Coefficients Relative Error in different AoA.*

The variables contained in Eq. 5.5 are once again the vorticity and the turbulent eddy viscosity, along with the local density and horizontal velocity. This combination leads to the conclusion that the Regressor found of uttermost importance two quantities (vorticity and turbulent eddy viscosity) closely related with turbulence and two quantities able to determine the change in the geometry of the initial conditions (velocity and density).

The new algebraic surrogate model (Eq. 5.5) is subsequently integrated into the PUMA solver to use on the two airfoils, in order to be evaluated in three AoA. The C_D and C_L coefficients results are summarized in Table 5.13. The lift and drag coefficients relative errors are very promising, as the predictions are slightly less accurate in comparison to the ones from Eqs. 5.2 (5.6) and 5.4 (5.13), but considerably better generalized in terms of geometry and inflow conditions. The trade-off of sacrificing the accuracy in some degree in order to produce a model that could be used in a wider range of applications seems fair.

The fields produced by the PUMA solver with Eq. 5.5 as the algebraic surrogate to the transition model are presented in Figs. 5.15 and 5.16 in comparison with the original ones, produced by the $\gamma - Re_\theta$ model. The impact of the intermittency field of the NLF(1)-0416 airfoil in the yielding of the expression is greater than the one of the OAT15A, because of the more significant and extended changes of the targeted quantity around the airfoil. This explains why the surrogate model captures the reference NLF(1)-0416 field significantly better than the reference OAT15A field.

The skin friction coefficients C_f distributions are presented in Fig. 5.14 for both

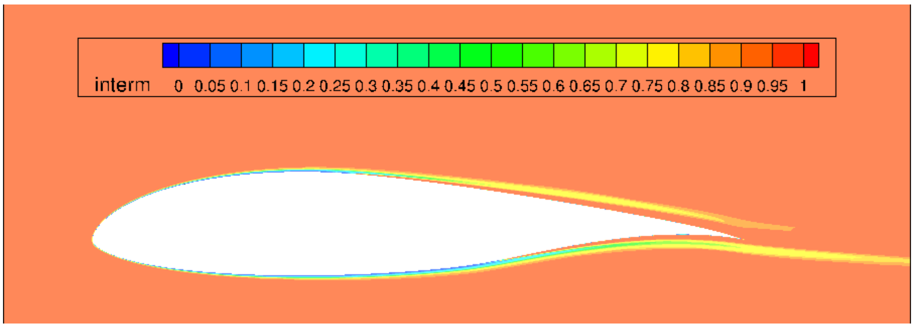


(a) NLF(1)-0416 airfoil - Suction Side

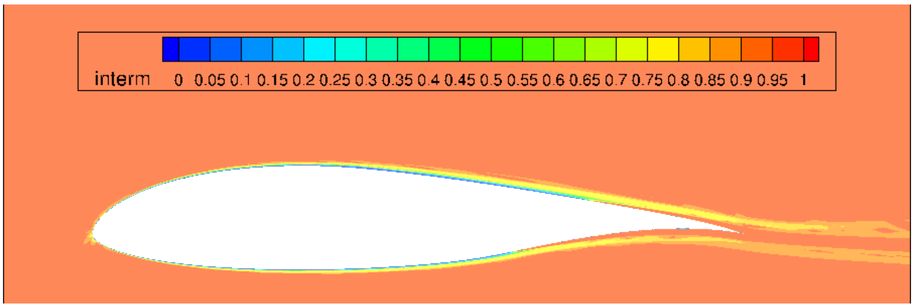
(b) OAT15A airfoil

Figure 5.14: *Skin friction coefficient C_f distribution comparison using Eqs. 5.2, 5.4, 5.5 at AoA=2 °. Eq.B is Eq. 5.2 -yielded with NLF(1)-0416 data-, Eq.E is Eq. 5.4 -yielded with OAT15A data- and Eq. F is Eq. 5.5 -yielded with data from both airfoils.*

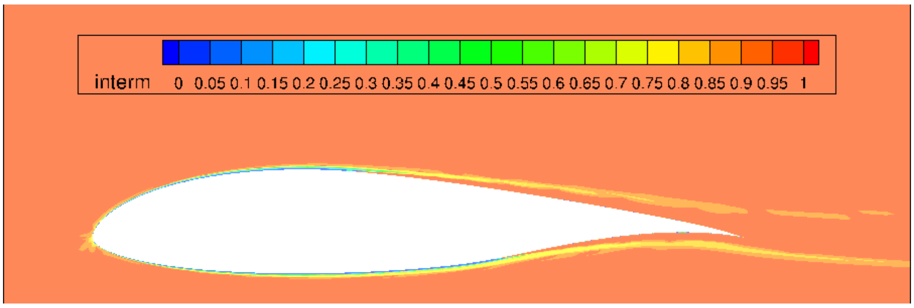
airfoils, and the transition point prediction using both the expressions trained from each airfoil separately and the expression trained from both is compared. Once again the prediction on the NLF(1)-0416 airfoil is sufficient due to its large impact on the training, while the prediction on the OAT15A airfoil is significantly worse.



(a) Intermittency Field using $\gamma - Re_\theta$

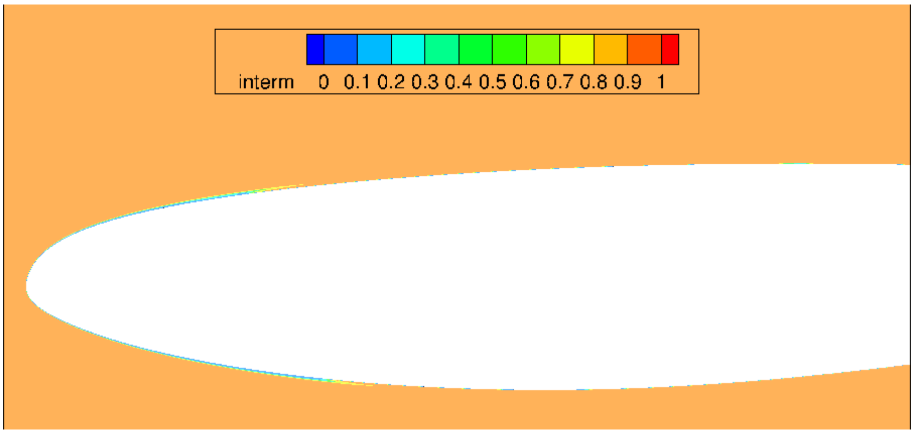


(b) Intermittency Field using Eq. 5.2, yielded from NLF(1)-0416 data.

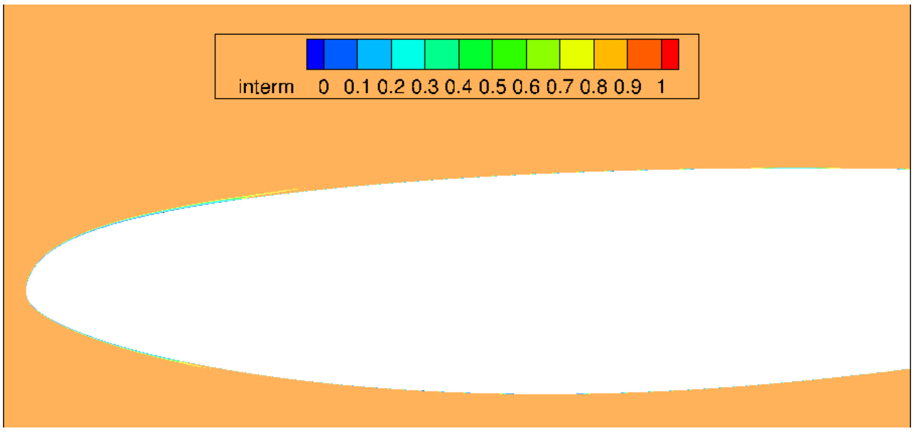


(c) Intermittency Field using Eq. 5.5, yielded from NLF(1)-0416 and OAT15A data.

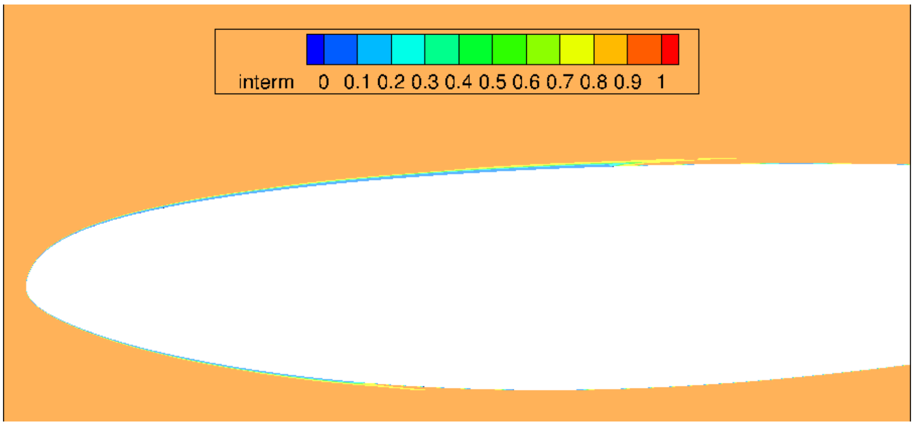
Figure 5.15: *NLF(1)-0416: Intermittency Field comparison at AoA=2°.*



(a) Intermittency Field using $\gamma - Re_\theta$



(b) Intermittency Field using Eq. 5.4, yielded from OAT15A data.



(c) Intermittency Field using Eq. 5.5, yielded from NLF(1)-0416 and OAT15A data.

Figure 5.16: *ONERA OAT15A: Intermittency Field comparison at AoA=2 °.*

Chapter 6

Applications of SR in Turbulence Modeling

This chapter focuses on the integration of the SR algorithm in Eddy Viscosity Adaptation - Symbolic Regression framework. The **Eddy Viscosity Adaptation (EVA)** methodology was recently developed by [50]. The aim of this particular methodology is to yield an algebraic surrogate to the SA turbulence model. This surrogate model, i.e. the algebraic expression for the eddy viscosity, will be based on the EVA corrective field, in order to minimize the discrepancy between the RANS-SA based velocity field and a high-fidelity field. Subsequently, it will be introduced in a ShpO loop for both test cases and its results will be compared to those of a frozen ν_t field approach.

Field Inversion refers to the process of backing-out data, such as DES or LES simulations, to solve a minimization problem for the difference between this high-fidelity data and the results of simple RANS solutions. A corrective field for the quantity of interest (e.g. the eddy viscosity) is yielded as a result of the optimization process. **Symbolic Regression** is subsequently used to create an algebraic surrogate to the turbulence model using the produced field as input.

The EVA methodology aims at minimizing the discrepancy between a high-fidelity reference velocity field and a RANS velocity field. In order to achieve this, an adjoint-based code within the OpenFOAM environment was created by [50], which optimizes the eddy viscosity ν_t in every cell of the computational mesh. The objective function of the optimization problem is the following:

$$J = \frac{1}{2}\int_{\Omega}(v_{RANS} - v_{HiFi})^2 d\Omega \tag{6.1}$$

where v_{HiFi} is the high-fidelity reference velocity field, considered the ground truth. The result of the process described above is a corrective field, which resembles the reference field more closely than the simple SA evaluation.

In this context, the EVA field is provided post-process to the SR algorithm and an algebraic expression for the eddy viscosity, containing local flow and geometric quantities, is obtained. This expression is integrated within the RANS simulation as a surrogate to the SA turbulence model, while the inverted eddy viscosity field and the accuracy of the drag coefficient C_D are evaluated. The final step is its integration within the ShpO framework, to test whether or not the dynamic updating of the ν_t field in each cycle provides useful information to the optimization algorithm and improves the final value of the objective function, i.e. the drag coefficient, compared to the Frozen Eddy Viscosity (FEV) optimization process.

The aforementioned methodology is tested on two cases within OpenFOAM:

- **Case D - NACA 0012 Airfoil**: 2D case on which the EVA methodology was primarily tested by [50].
- **Case E - Ahmed Body** with 35° slant: 3D case with prominent flow separation.

6.1 Case D: NACA 0012 Airfoil

The NACA 0012 Airfoil is a commonly used test case for validating the accuracy of turbulence models and surrogates. The computational mesh is a C-type 2D grid containing 37800 cells, as illustrated in Fig. 6.1. All simulations are performed for $Re = 10^6$ and $AoA = 12°$.

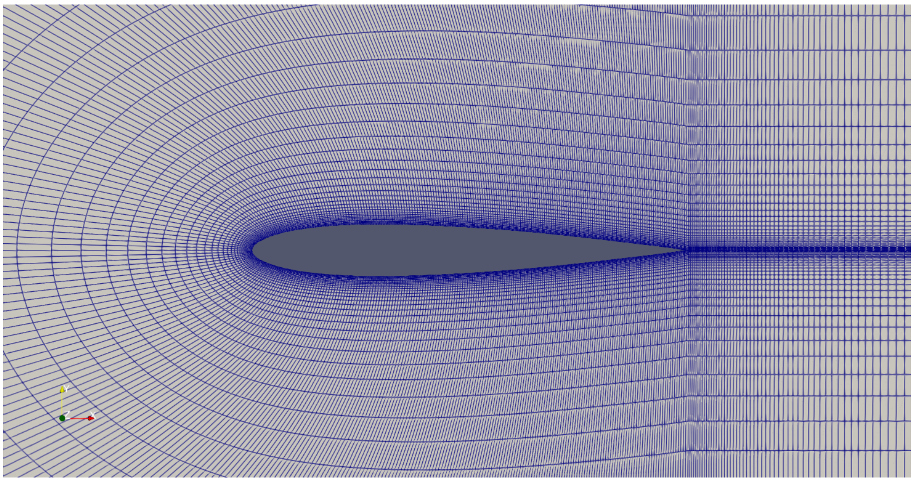


Figure 6.1: *NACA 0012 Airfoil: The C-type 2D grid used for the OpenFOAM simulations.*

6.1.1 EVA - SR

The EVA field, which was constructed by [35], is used as the input data for the SR Regressor. The input variables are described in Table 6.1 and the algorithm's settings in Table 6.2. The strain rate is introduced as an input variable, because of its inherent relation to turbulence, while the density and local Mach number are not fed into the algorithm. The EA population number and size are once again 30 and 50 accordingly. The maximum depth is restricted to 5 in order to produce a less nested expression for the turbulence surrogate model. The allowed unary operators are the same as the ones in the previous cases except for the power operator, which has been removed alongside the trigonometric binary operators, for simplicity and numerical stability purposes. Also, a custom loss function was implemented in order to prevent the yielding of an expression which would result in instability when integrated into the RANS solver. Specifically, the loss function is the MSE of the input EVA ν_t and the predicted $\hat{\nu}_t$ values , as described in Eq. 6.2, but a "death-penalty" factor of 10^{10} is added if the wall distance is placed in the denominator of the expression. This is achieved by searching the expression tree and locating the division nodes. If the right child, which corresponds to the denominator of the fraction, is the wall distance, the death-penalty is added. In this way the unwanted expression has a very low fitness score and is thus eliminated from the selection process. The ν_t values that are fed into the Regressor are also normalized within the range of [0,1],

SR Input	Normalized Quantity	Expression
c_0	Coordinate x	x/c
c_1	Coordinate y	y/c
c_2	Pressure	$p/(0.5 \cdot V_\infty^2)$
c_3	Strain Rate	$S \cdot c/V_\infty$
c_4	Velocity x	V_x/V_∞
c_5	Velocity y	V_y/V_∞
c_6	Vorticity	$\Omega \cdot c/V_\infty$
c_7	Wall Distance	d_{wall}/c

Table 6.1: *SR Training: The 8 Normalized Input Variables.*

EA Parameter	Value
Population Size	50
Populations	30
Selection Criteria	Lowest MSE & Complexity
Maximum Complexity	30
Maximum Depth	5
Binary Operators	$+, -, *, /$
Unary Operators	$exp, log, sqrt$

Table 6.2: *SR Training: Evolutionary Algorithm Parameters.*

in order to achieve non-dimensionality and better generalization.

$$F_{loss} = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(\nu_{t,i} - \hat{\nu}_{t,i})^2 & \text{if } c_7 \text{ not in denominator} \\ \frac{1}{n}\sum_{i=1}^{n}(\nu_{t,i} - \hat{\nu}_{t,i})^2 + 10^{10} & \text{otherwise} \end{cases} \tag{6.2}$$

with $n = 37800$ the number of cells of the grid.

$$\nu_t = 0.013 \cdot [\frac{c_0 + c_7}{c_4\left(\frac{11.202037}{c_3}\right) + 1.7163204} + 0.032510918] \tag{6.3}$$

The algorithm produces a number of expressions. Some of them are integrated within the OpenFOAM solver in order to produce a C_d prediction and fields to be evaluated, in the context of a parametric analysis. The drag coefficient results are summarized in Table 6.3 and the corresponding eddy viscosity fields in Fig. 6.2. Fig. 6.2(f) illustrates the inability of expressions of higher complexity to converge, due to numerical instabilities which stem from the final expressions form. This is also validated by the corresponding velocity field, shown in Fig. 6.3.

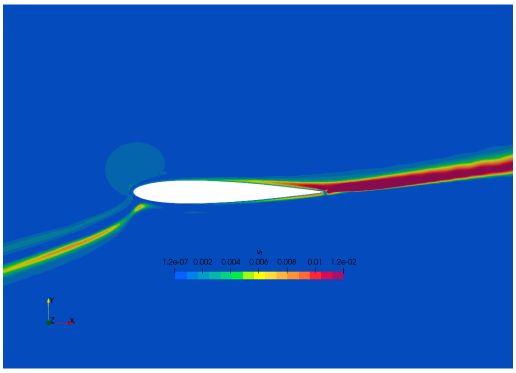
Complexity	SR MSE	Variables	C_d
9	0.0093	c_0, c_4	0.02786
11	0.0087	c_0, c_4, c_5	0.02793
13	0.0082	c_0, c_4, c_5, c_8	0.02793
14	0.0079	c_0, c_4, c_5, c_8	0.02836
29	0.0052	$c_0, c_2, c_3, c_4, c_5, c_8$	0.08445
DES Value			0.02390

Table 6.3: *SR Parametric Analysis: Resulting expressions of increasing complexity and corresponding drag coefficient results.*

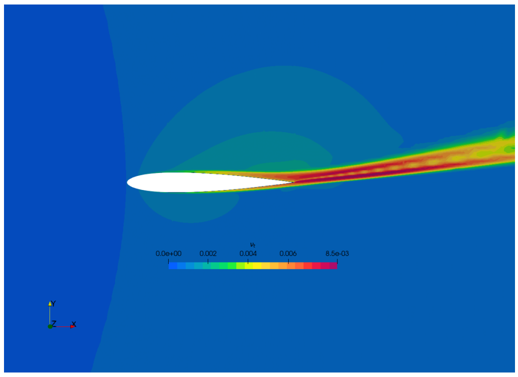
The expression described in Eq. 6.3 with a complexity of 13 and MSE = 0.00816 is the one chosen as the surrogate turbulence model, as lower complexity equations lack accuracy in terms of ν_t field prediction, whereas the higher complexity equations that were tested resulted in numerical instability. The resulting equation contains the x position, the strain rate, the horizontal velocity and the wall distance. As expected, both geometric and flow quantities are taken into account in order to have an adequate eddy viscosity prediction.

Eq. 6.3 is integrated within the OpenFOAM code as a surrogate, replacing the SA turbulence model. A RANS simulation is then performed and the results are subsequently evaluated. The EVA (input) and SR (output) eddy viscosity fields are illustrated in Fig. 6.4. The SR output field presents a sufficient resemblance to the EVA (input) field, while the flowline up to the leading edge of the airfoil which the EVA methodology produced but had no physical meaning is removed. This is explained by the fact that the new surrogate model performs algebraic calculations with only local quantities, which are integrated in a continuous manner within the model, leaving no room for numerical instability and non-physical errors.

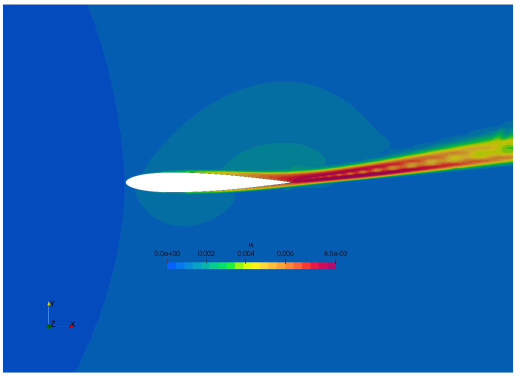
Fig. 6.5 illustrates the four velocity fields produced by the DES, RANS, EVA and SR simulations. The DES field is considered the ground truth to which the rest of the results are compared. The RANS field is produced by implementing the original SA turbulence model. It is apparent that the RANS simulation fails to perfectly model the separation region close to the trailing edge and the wake. This is the main weakness that the EVA method improves, as shown in the corresponding figure. The EVA corrected field is the input for the SR algorithm and the last field is produced after using the output expression for ν_t as the surrogate turbulence model



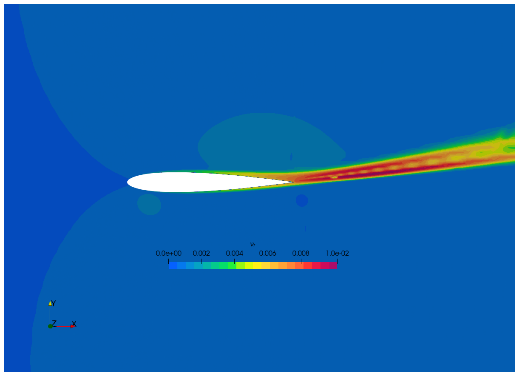
(a) Eddy Viscosity Field produced with EVA.



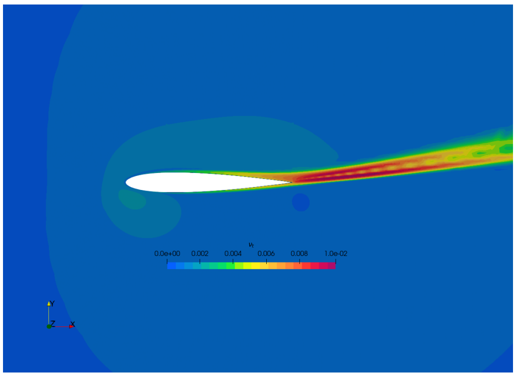
(b) Eddy Viscosity Field produced with SR Expression with a complexity of 9.



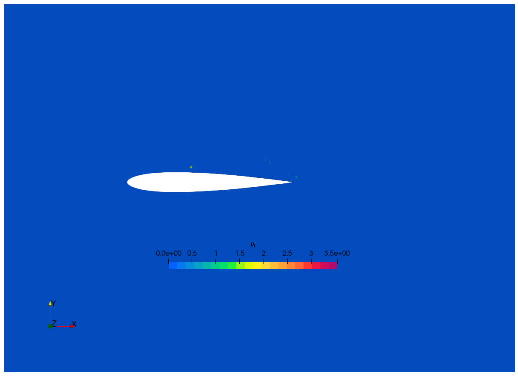
(c) Eddy Viscosity Field produced with SR Expression with a complexity of 11.



(d) Eddy Viscosity Field produced with SR Expression with a complexity of 13



(e) Eddy Viscosity Field produced with SR Expression with a complexity of 14.



(f) Eddy Viscosity Field produced with SR Expression with a complexity of 29.

Figure 6.2: *SR Parametric Analysis: Eddy Viscosity fields using the EVA method and SR surrogates of varying complexities.*

in the RANS simulation. It is clear that the field inversion process is successful with the use of the surrogate model, as the SR field, while providing a less accurate evaluation of the eddy viscosity field than the EVA method, improves the modeling

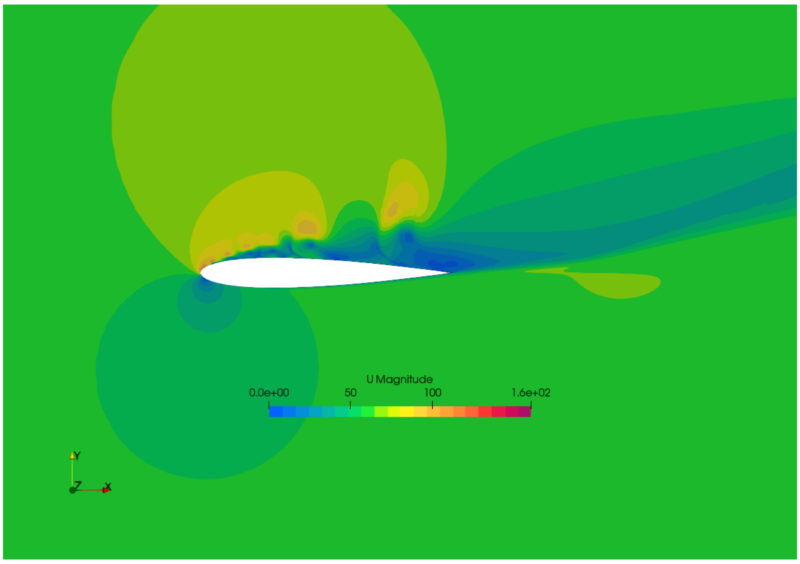
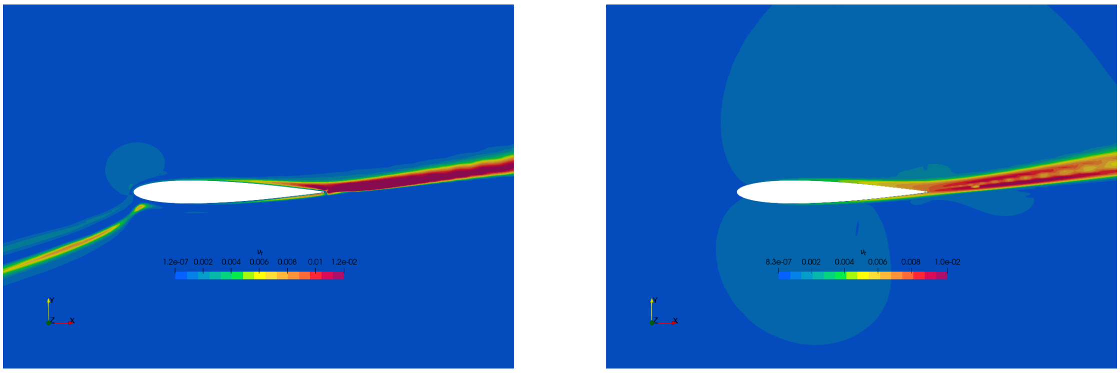


Figure 6.3: *NACA 0012: Velocity Magnitude field using the RANS-SR method, implementing the expression with complexity of 29. The solver is unable to converge, due to numerical instabilities stemming from the expression's form.*

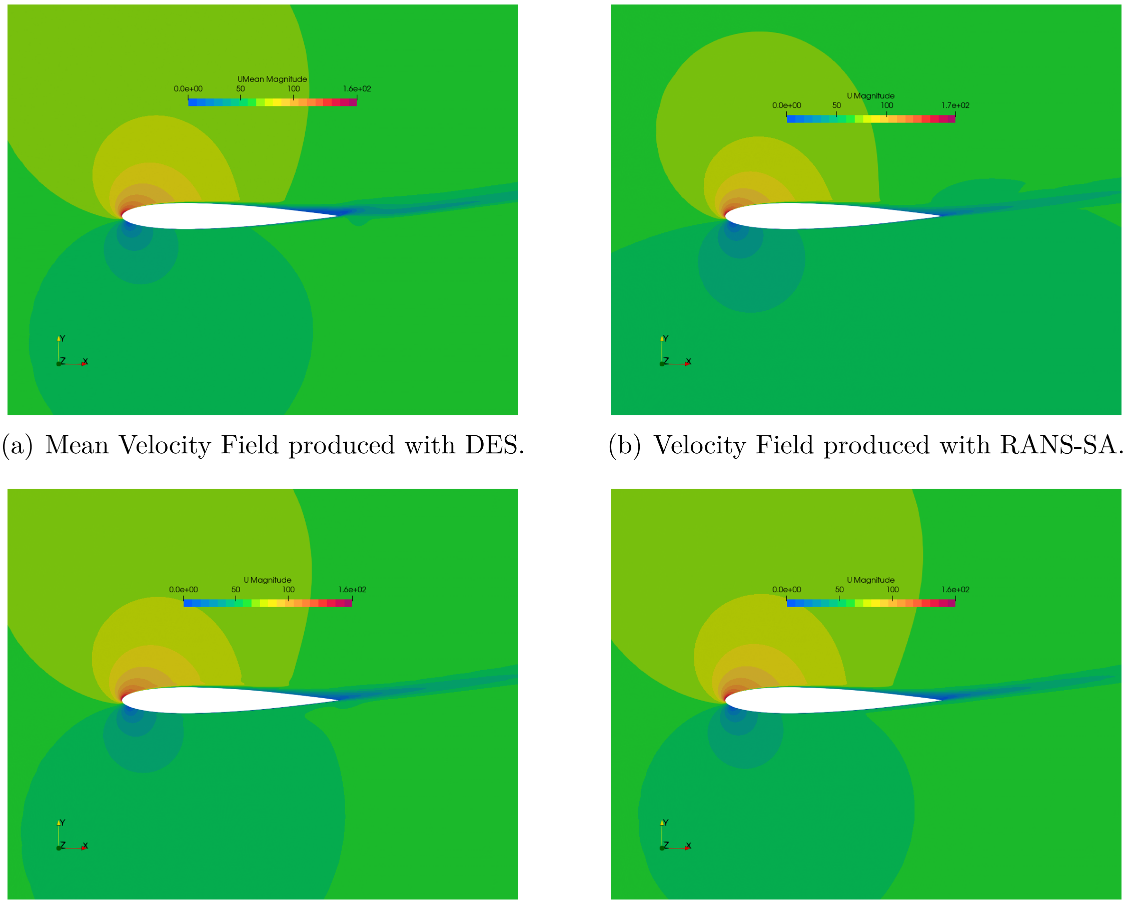


(a) Eddy Viscosity Field produced with EVA.

(b) Eddy Viscosity Field produced with Final SR Surrogate.

Figure 6.4: *Eddy Viscosity fields using the EVA and RANS-SR methods.*

of the separation zone compared to the RANS-SA evaluation. This aligns with what was expected of this method, as the Regressor was fed with data from the corrected EVA field, thus being taught to predict the separation and recirculation region more sufficiently. The lesser accuracy compared to the EVA method is also completely logical, as the SR process can never result in a loss function (MSE between input and output) of exactly zero. It would therefore never be expected that the field produced by SR would be as accurate or even better than the EVA field.



(a) Mean Velocity Field produced with DES.

(b) Velocity Field produced with RANS-SA.

(c) Velocity Field produced with EVA.

(d) Velocity Field produced with RANS-SR.

Figure 6.5: *Velocity fields using DES, RANS-SA, EVA, RANS-SR methods.*

Subsequently, the four values of the drag coefficient are compared in Table 6.4. The RANS simulation provides the most accurate C_D prediction, followed by the SR-surrogate, while the EVA method holds the last place. This does not, however, signify that the SR-surrogate results in a more accurate flow modeling compared to the EVA method, as the drag coefficient was the target quantity neither for the EVA method nor for the SR. The improvement of the C_D value is considered coincidental, as it is simply a metric taken into account and not the quantity of interest for the field inversion process.

6.1.2 ShpO with the SR-based Turbulence Model

The effectiveness of Eq. 6.3 as a surrogate turbulence model is further tested with its implementation in a ShpO loop for the NACA 0012 Airfoil. Every optimization

Method	C_D
DES	0.0240
RANS-SA	0.0242
EVA	0.0292
SR	0.0279

Table 6.4: *NACA 0012 Airfoil: Drag Coefficient calculated with DES, RANS with SA, the EVA method and the SR surrogate turbulence model.*

process described from this point on refers to flow conditions of $Re = 10^6$ and $AoA = 12°$. The parametrization of the airfoil is done using volumetric B-Splines, as in [50], where the geometry changes through the displacement of the 4 free control points. The objective for the ShpO problem is the drag coefficient C_D minimization.

It has already been deduced by [50] that the FEV optimization method eliminates the computational overhead of performing unsteady adjoint evaluations for ShpO, provided that the high-fidelity data of the DES simulation are given and there is no need for performing such costly evaluations beforehand. In this ShpO loop, the EVA eddy viscosity field was frozen throughout the optimization process, thus eliminating the computational cost related to the turbulence model adjoint equations.

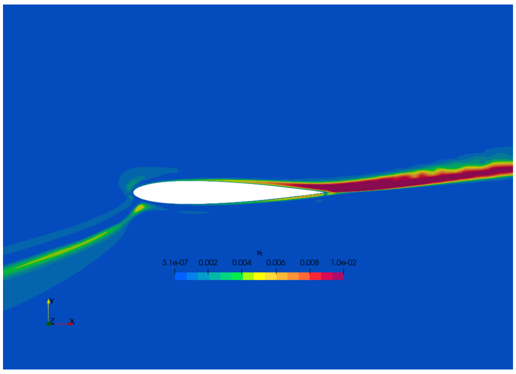
Based on this approach, a ShpO loop for the airfoil using not the EVA field but the algebraic expression for ν_t (6.3) is proposed. In this way, the eddy viscosity field is not frozen, but rather dynamically updated in each optimization cycle, using the geometric and flow data calculated in the previous cycle. The algebraic expression remains fixed and is not differentiated to provide added terms for the FAEs.

The ν_t field around the initial and the optimized geometry with the EVA and SR fields is illustrated in Fig. 6.6. The proposed method holds a significant advantage compared to the FEV process. The change in the airfoil's geometry is taken into account through the algebraic expression for ν_t, providing a more accurate evaluation in each optimization cycle, as opposed to the FEV method, where after a number of optimization cycles the frozen field is no longer corresponding with the final geometry. This results in a lower final objective function value compared to the previous method. The C_D of the optimized geometries are reevaluated with DES, as summarized in Table 6.5.

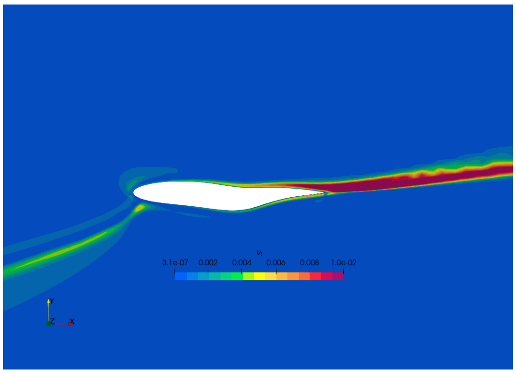
The results of this process show that SR-based modeling can be significantly advan-

Method	C_D
EVA	0.0239
SR	0.0233

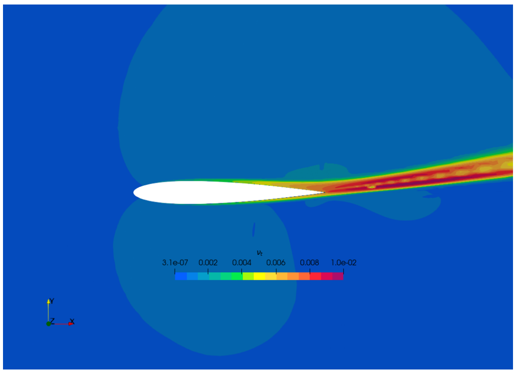
Table 6.5: *NACA 0012 Airfoil ShpO: Drag Coefficient of final geometries -reevaluated with DES- optimized with the EVA method and the fixed SR surrogate expression.*



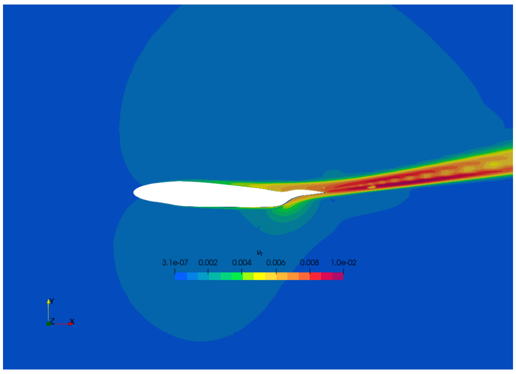
(a) Initial Geometry with the EVA field.



(b) Final Geometry with the EVA field.



(c) Initial Geometry with the SR field.



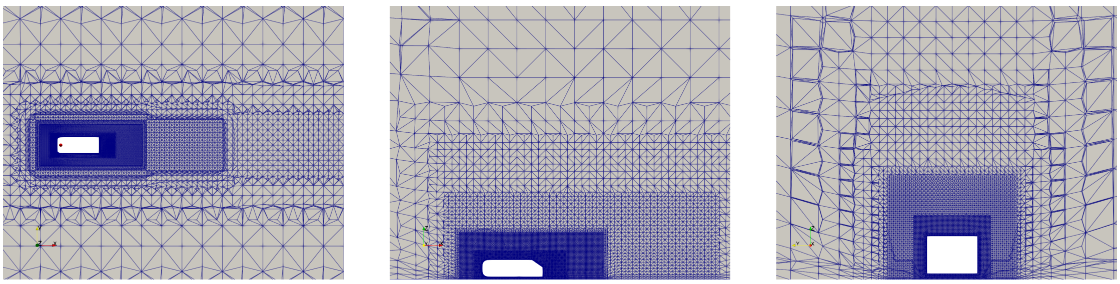
(d) Final Geometry with the SR field.

Figure 6.6: *ShpO of the NACA 0012 Airfoil: Initial and final eddy viscosity fields using the EVA and RANS-SR methods.*

tageous in terms of ShpO. The cost of the original turbulence-model's differentiation and integration into the FAEs is eliminated, like in the FEV approach, because the calculation of a simple algebraic expression in each optimization cycle adds no computational overhead, while the added benefit of the dynamically updated field is of uttermost importance. The time needed for the yielding of an algebraic expression through the SR algorithm must of course be taken into account to make the comparison fair. This step adds some cost to the proposed process, although if a case-specific or even more generalized algebraic model has already been created, its use in the methodology described above adds no computational overhead.

6.2 Case E: Ahmed Body

The second case on which SR-based turbulence modeling is tested is the Ahmed Body [2] with 35° slant, serving the purpose of testing the aforementioned methodology on a 3D geometry. This specific geometry was chosen because it results in a large separation and recirculation region, which is what the EVA process seeks to better model. The computational grid, consisting of 2925403 cells is illustrated in Fig. 6.7. The simulations conducted in this thesis are for freestream velocity $V_\infty = 40 m/s$ and zero angles of attack.



(a) XY Grid. (b) XZ Grid. (c) YZ Grid.

Figure 6.7: *Ahmed Body: 3D Computational grid.*

6.2.1 EVA - SR

The process followed is the same as in Case D. High-fidelity DES simulation data is backed-out and the EVA corrected field is created. It is subsequently fed into the Regressor in order to produce an analytical expression for ν_t with the input variables

SR Input	Normalized Quantity	Expression
c_0	Coordinate x	x/c
c_1	Coordinate y	y/c
c_2	Coordinate z	z/c
c_3	Pressure	$p/(0.5 \cdot V_\infty^2)$
c_4	Strain Rate	$S \cdot c/V_\infty$
c_5	Velocity x	V_x/V_∞
c_6	Velocity y	V_y/V_∞
c_7	Velocity z	V_z/V_∞
c_8	Vorticity Magnitude	$\Omega \cdot c/V_\infty$
c_9	Wall Distance	d_{wall}/c

Table 6.6: *SR Training: The 10 Normalized Input Variables. The reference length is $c = 1044$.*

EA Parameter	Value
Population Size	50
Populations	30
Selection Criteria	Lowest MSE & Complexity
Maximum Complexity	30
Maximum Depth	5
Binary Operators	$+, -, *, /$
Unary Operators	$log, sqrt$

Table 6.7: *SR Training: Evolutionary Algorithm Parameters.*

being the ones in Table 6.6 and the SR parameters the ones in Table 6.7. The ν_t values that are fed into the Regressor are once again normalized within the range of [0,1]. The loss function is also the same as in the previous case, as described in Eq. 6.2, with $n = 2925403$. Because of the large size of the 3D input data (2925403 cells $\times$ 10 input variables), the regression is set to run in parallel on 16 processors, as the PySR library allows for parallelization in order to reduce the search time.

The Regressor converged to a set of analytical expressions and the one chosen to serve as the new algebraic surrogate turbulence model is the following.

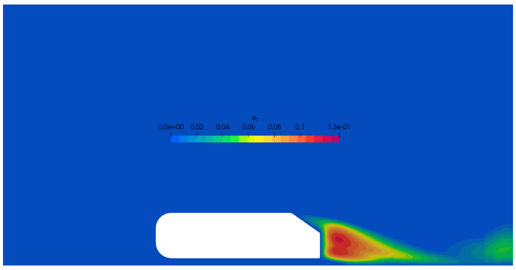
$$\nu_t = 0.13 \cdot (c_5 - 1.0012314) \cdot (c_9 - 5234.8203) \tag{6.4}$$

where $c_5 = V_x/V_\infty$ and $c_9 = d_{wall}/c$.

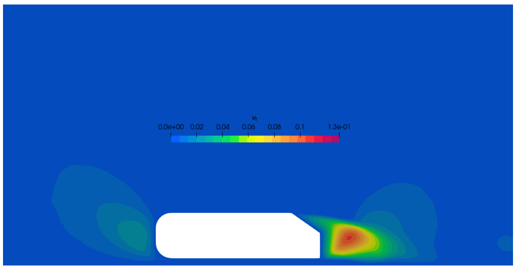
The EVA (input) field and the SR (output) ν_t field are illustrated in Fig. 6.8. The

Method	C_D
DES	0.306
RANS-SA	0.329
EVA	0.313
RANS-SR	0.310

Table 6.8: *Ahmed Body: Drag Coefficient calculated with DES, RANS with SA, the EVA method and the SR surrogate turbulence model.*



(a) Eddy Viscosity Field produced with EVA - Side View.



(b) Eddy Viscosity Field produced with SR - Side View.

Figure 6.8: *Eddy Viscosity fields using the EVA and RANS-SR methods for the Ahmed Body.*

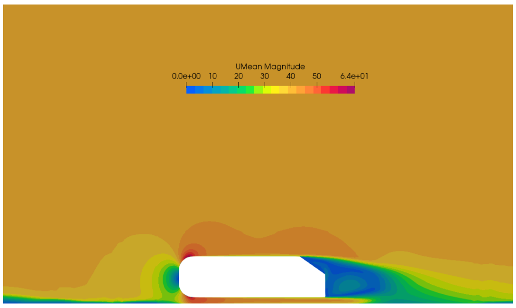
output field closely resembles the input field, while presenting some discrepancies that are awaited due to the fact that the SR algorithm's loss is not zero.

In terms of separation modeling, the velocity fields produced by the DES, RANS-SA, EVA and RANS-SR simulations are illustrated in Fig. 6.9. Similarly to the NACA 0012 Airfoil case, the RANS-SR field successfully models the separation region in the rear of the body, providing a closer result to the DES reference field, compared to the RANS-SA field.

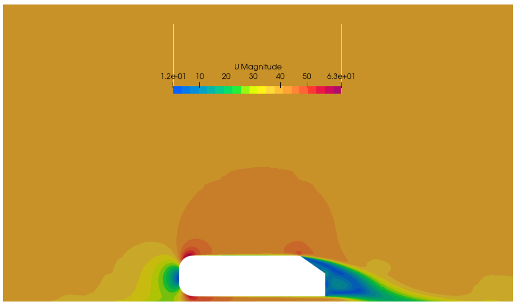
The corresponding drag coefficient results are presented in Table 6.8. The DES value is the ground-truth, closer to which is the RANS-SR value. This is once again coincidental, as the C_D minimization is not the target of the EVA and SR methodologies.

6.2.2 ShpO with the SR-based Turbulence Model

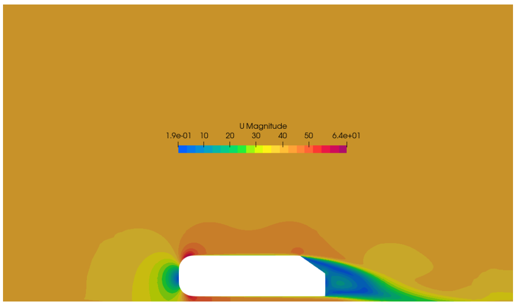
Similarly to the previous case, the Ahmed body is subjected into a ShpO loop, using the SR-based surrogate for the calculation of the ν_t field, in order to take advantage of the physical quantities contained, which will dynamically update the field in each



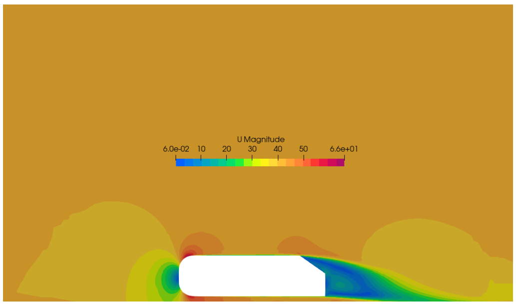
(a) Mean Velocity Field produced with DES.



(b) Velocity Field produced with RANS-SA.



(c) Velocity Field produced with EVA.



(d) Velocity Field produced with RANS-SR.

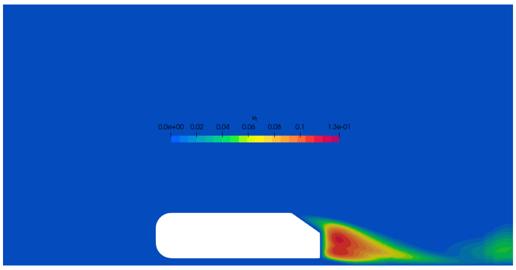
Figure 6.9: *Velocity fields using DES, RANS-SA, EVA, RANS-SR methods for the Ahmed Body.*

optimization cycle. The results are compared with the EVA-based ShpO performed by [35].

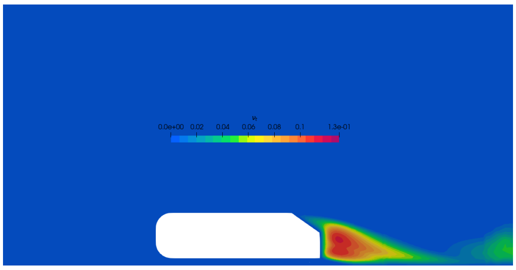
The SR-based ShpO is performed using two strategies, i.e. the use of the fixed algebraic expression throughout the optimization process, exactly as in Case D, and the full differentiation of the surrogate, in view of its complete integration within the continuous adjoint formulation.

For the second approach, Eq. 6.4 is differentiated and provides added terms to the FAEs. More specifically, the algebraic surrogate depends on the streamwise velocity component $c_5 = V_x/V_\infty$ and the wall distance $c_9 = d_{wall}/c$, the differentiation of which results in added terms to the adjoint momentum and eikonal equations respectively. This process is completed by [35] and a final ShpO loop is performed, using the fully differentiated closed form surrogate.

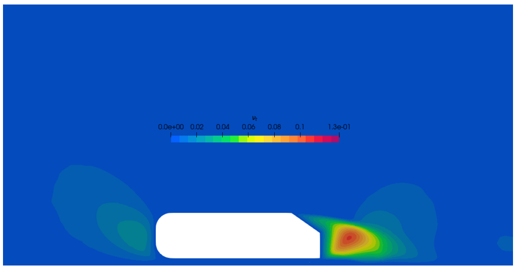
The final results are summarized in Table 6.9, whereas the final eddy viscosity fields and the final geometries are illustrated in Figs. 6.10 and 6.11 respectively.



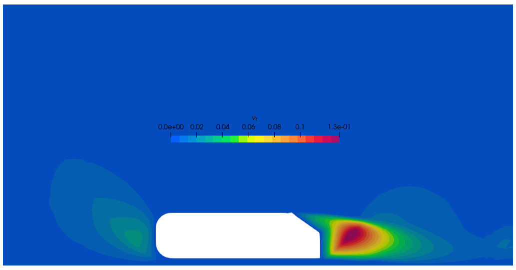
(a) Initial Geometry with the EVA field.



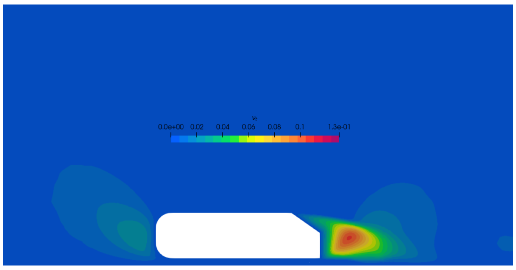
(b) Final Geometry with the EVA field.



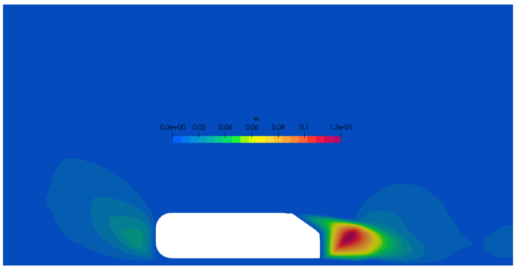
(c) Initial Geometry with the SR field.



(d) Final Geometry with the SR field - Fixed algebraic expression without differentiation.



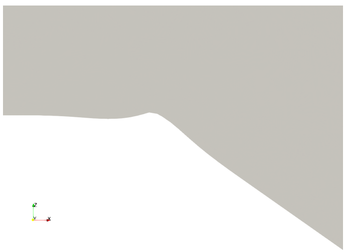
(e) Initial Geometry with the SR field.



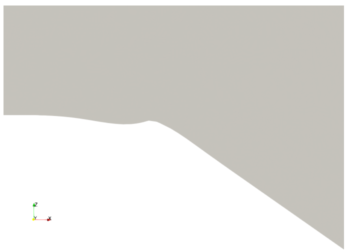
(f) Final Geometry with the SR field - Fully Differentiated.

Figure 6.10: *ShpO of the Ahmed Body: Initial and final eddy viscosity fields using the EVA and RANS-SR methods.*

(a) Final Geometry produced with the EVA method.



(b) Final Geometry produced with the RANS-SR method - Fixed algebraic expression without differentiation.



(c) Final Geometry produced with the RANS-SR method - Fully Differentiated.

Figure 6.11: *ShpO of the Ahmed Body: Final Geometry produced with the EVA and RANS-SR methods.*

Method	C_D Optimization	C_D DES Reevaluation
EVA	0.304	0.273
SR-Fixed	0.305	0.269
SR-Diff	0.299	0.289

Table 6.9: *Ahmed Body ShpO: Drag Coefficient of final geometries optimized with the EVA method, the fixed and the differentiated SR surrogate expression. Column 2: Evaluation with the same turbulence model that was used in the optimization process. Column 3: Reevaluation with DES.*

Similarly to Case D, the dynamic updating of the eddy viscosity field through the fixed SR-based surrogate yields slightly better results, regarding the objective function, compared to the FEV approach. This is attributed to the ability of the algebraic surrogate model to account for the variations of the geometry in each optimization cycle. Furthermore, the optimization algorithm failed after only one cycle in the FEV approach, as the frozen field could not result in a convergence of the primal problem after even a slight change in the geometry. However, the RANS-Fixed SR optimization process withstood four optimization cycles, resulting in a bigger change in the geometry.

Regarding the full differentiation approach, where the surrogate model is fully integrated within the adjoint framework, the best objective function value, when evaluated using the same model employed during the optimization process, is achieved by the SR-Differentiated method. However, the drag coefficient reduction in the final geometry is smaller compared to the other two methods, when reevaluated with DES. This can be attributed to the fact that as the geometry undergoes significant changes during the ShpO process, the surrogate model suffers from a degradation in predictive fidelity. Because the algebraic expression was derived exclusively from the flow field around the baseline geometry, its generalization capability decreases as the shape evolves, preventing it from accurately reconstructing the ν_t field around the highly modified final configurations. Since the gradient information relies on an algebraic surrogate that has drifted outside its training data, the calculated derivatives lack accuracy, leading the ShpO process to converge toward a non-optimal geometric configuration.

In each case the optimization algorithm converges in the creation of a formation resembling a spoiler, which is awaited for a basic car model optimization, as illustrated in Fig. 6.11.

Chapter 7

Conclusions & Future Research

This thesis focuses on the building of algebraic surrogates for transition and turbulence modeling in CFD, using Symbolic Regression (SR). This concept is based on the interpretability advantage of the SR-based surrogate models, compared to the "black-box" nature of a DNN-based surrogate. Considering also the fact that the integration of such an algebraic surrogate in the CFD solver results in lower cost computations, due to the elimination of the turbulence or transition transport PDEs by an algebraic expression, this approach is very promising for the development of computationally efficient and physically consistent CFD closures.

The programmed SR code employs the PySR library. The yielding process of the algebraic surrogate is structured as follows. A CFD simulation, which provides the reference flow data is performed using the classic closure models. Flow and geometric quantities are obtained and normalized, in order to be fed into the regressor. The normalization's aim is twofold, i.e. maintaining unit consistency and achieving better generalization in a wider range of cases. If necessary, the output target values (either the eddy viscosity ν_t for turbulence or the intermittency factor γ for transition) are also normalized for better generalization. The regressor's parameters, namely the evolutionary and allowed mathematical operators, are subsequently defined, and the final regression model is built by minimizing the loss function, with predetermined maximum complexity and depth allowed in the resulting expressions. The algorithm converges to a set of algebraic expressions with increasing complexity and simultaneously decreasing loss value. Some of these expressions are chosen to serve as surrogate algebraic turbulence or transition models within the CFD solver.

For transition modeling, the first test case was the NLF(1)-0416 Airfoil, which is designed to have a transition point at approximately 60% of the chord length. The above process was implemented and the derived surrogate model resulted in a sufficient prediction of the aerodynamic coefficients (limiting the C_D & C_L relative error to -0.7% & 0.11% respectively) when tested close to the training Angle of Attack (AoA), but the error increased when the testing AoA was further away. This inspired the construction of a correction loop, which seems to provide better results after two rounds (reducing the error from -33.30% and 4.77% to -5.47% and 0.28% for C_D and C_L respectively). A more generalized algebraic surrogate model, trained from fields of two different angles of attack, was created and although its accuracy was lower (with relative errors reaching up to -37.00% for C_D and 4.96% for C_L), it resulted in extended applicability within a range of AoA. Similar to the C_D and C_L values, the transition point prediction is accurate in AoA close to the training one, but loses accuracy otherwise.

The second case was the OAT15A transonic Airfoil. The surrogate transition model produced from this case was very successful in terms of C_D and C_L error, in a wide range of AoA (with a maximum C_D relative error of just 1.01% at $AoA = 1°$). However, it results in the underestimation of the location of the transition point.

The third case was the yielding of a surrogate model from two fields: one from the NLF(1)-0416 and one from the OAT15A Airfoil in the same AoA. The goal here was to create a more generalized model, which would provide more consistent predictions for both low-speed and high-speed/transonic flows. The resulting surrogate model captures the NLF(1)-0416 field better than the OAT15A field, which is logical, as the intermittency field of the NLF(1)-0416 airfoil has a greater extent and thus a higher weight in the training process. However, the aerodynamic coefficients' results are relatively accurate in both airfoils (with C_D relative errors ranging from 0.89% to 7.93% and C_L errors from -0.09% to 6.53% across the tested angles of attack for both geometries).

In terms of turbulence modeling, SR was employed within the Eddy Viscosity Adaptation (EVA) framework. More specifically, the EVA methodology was implemented in order to create the corrective field which was served as the input for the SR algorithm. The regressor's output was an analytical expression for the eddy viscosity, which served as the turbulence closure, replacing the standard turbulence model in the CFD calculation. The surrogate model was also employed within a Shape Optimization (ShpO) loop, and the results were compared to those of the Frozen Eddy Viscosity (FEV) approach.

This methodology was first tested on the NACA 0012 Airfoil, on which the modeling of the separation zone (i.e., the area of interest, which the RANS-SA simulation failed to model accurately and which the EVA process seeks to improve) with the SR-based surrogate was very sufficient. The ShpO loop which implemented the surrogate model resulted in a better final value of the objective function ($C_D = 0.0233$), compared to the final result of the FEV ShpO loop ($C_D = 0.0239$).

The EVA-SR process was finally employed on the Ahmed Body with 35° slant. The SR-based surrogate yielded sufficient results concerning the separation and recirculation region modeling and C_D results ($C_D = 0.310$) comparable to the high-fidelity DES data ($C_D = 0.306$) and the EVA method ($C_D = 0.313$), but significantly better than the simple RANS-SA value ($C_D = 0.329$). As for the ShpO, the final drag coefficient (0.269) is slightly improved compared to the final value of the ShpO with the FEV process (0.273), as the process benefits from the closed-form expression for the eddy viscosity in the primal problem calculation. Finally, the algebraic surrogate based on the Ahmed body was differentiated, providing full closure of the continuous adjoint formulation. However, the results of the fully differentiated version of the adjoint-based ShpO were not very sufficient (0.289), as the fidelity of the original model -which was trained only on the baseline geometry- is severely degraded after significant changes in the shape of the geometry.

The development of SR-based transition and turbulence models constitutes a highly promising approach in the field of data-driven modeling, primarily due to the interpretability of the resulting surrogate models and the significant reduction in computational overhead. While the current results demonstrate that the accuracy of the surrogates is sensitive to the extent and diversity of the training datasets, the methodology has proven its ability to capture the underlying physics of complex phenomena like boundary layer transition and shock-induced separation.

The creation of an extensive field database with a variety of geometries exposed in flows with ranging conditions, can lead to the yielding of a robust and generalized SR-based model. This could lead to a less case-sensitive model which would be more trusted in various calculations.

A final but critical issue that must be addressed is the numerical robustness of the SR-derived expressions when integrated into the RANS solver. Currently, a significant portion of the candidate models — despite showing high predictive accuracy during the training phase — fail to maintain numerical stability during the iterative CFD process. This necessitates a ”trial and error” approach, where numerous equations from the Pareto front must be manually assesed within the solver.

To overcome this bottleneck, future research should focus on the implementation of physical and numerical constraints directly into the SR loss function. By penalizing expressions that exhibit non-physical behavior (such as negative eddy viscosity) or extreme gradients, the regressor can be guided toward solutions that are inherently stable.

Bibliography

[1] B. J. Abu-Ghannam and R. Shaw. Natural transition of boundary layers — The effects of turbulence, pressure gradient, and flow history. *Journal of Mechanical Engineering Science*, 22(5):213–228, 1980.

[2] S. R. Ahmed, G. Ramm, and G. Faltin. Some salient features of the time-averaged ground vehicle wake. *SAE Transactions*, 93:473–503, 1984.

[3] V. G. Asouti. *Aerodynamic Analysis and Design Methods at High and Low Speed Flows, on Multiprocessor Platforms (in Greek)*. PhD thesis, National Technical University of Athens, 2009.

[4] B. Baldwin and H. Lomax. Thin layer approximation and algebraic model for separated turbulent flows. In *16th Aerospace Sciences Meeting*, page 257, 1978.

[5] S. Brunton, B. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *arXiv*, 2019.

[6] P. V. Bulat, N. V. Prodan, and A. A. Kurnukhin. On the influence of the laminar-turbulent transition in the numerical modeling of the wing airfoil. *Russian Aeronautics*, 64(3):455–465, July 2021.

[7] T. Cebeci and A. M. O. Smith. Analysis of turbulent boundary layers. Technical report, Douglas Aircraft Co Inc Long Beach CA, 1967.

[8] G. N. Coleman and R. Sandberg. A primer on direct numerical simulation of turbulence – methods, procedures and guidelines. Mar. 2010.

[9] M. Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl, 2023.

[10] C. Darwin. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray, London, 1859.

[11] Y. de Valk. Optimization of airfoil shape and its impact on the laminar-to-turbulent transition. Master's thesis, University of Twente, Enschede, Netherlands, July 2022.

[12] K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51(1):357–377, Jan. 2019.

[13] K. Duraisamy, Z. J. Zhang, and A. P. Singh. New approaches in turbulence and transition modeling using data-driven techniques. In *53rd AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, Jan. 2015.

[14] Y. D. Dwivedi, A. Wahab, A. D. Pallay, et al. Effects of surface roughness on separated and transitional flows over a wing. *AIAA Journal*, 50(3):593–609, Mar. 2012.

[15] H. W. Emmons. The laminar-turbulent transition in a boundary layer — Part I. *Journal of the Aeronautical Sciences*, 18(7):490–498, July 1951.

[16] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765, July 1991.

[17] K. C. Giannakoglou. Optimization methods in aerodynamics (in Greek). Lecture Notes (4th edition, with exercises), National Technical University of Athens (NTUA), School of Mechanical Engineering, 2006.

[18] M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65(3–4):393–415, Dec. 2000.

[19] Z. He, C. Wu, and Y. Zhang. A field inversion and symbolic regression enhanced Spalart–Allmaras model for airfoil stall prediction. *Physics of Fluids*, 36(6), June 2024.

[20] A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Proceedings: Mathematical and Physical Sciences*, 434(1890):9–13, 1991.

[21] M. G. Kontou. *The Continuous Adjoint Method with Consistent Discretization Schemes for Transitional Flows and the Use of Deep Neural Networks in Shape Optimization in Fluid Mechanics*. PhD thesis, National Technical University of Athens, 2023.

[22] J. R. Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4(2), June 1994.

[23] S. Kubacki and E. Dick. An algebraic intermittency model for prediction of transitional boundary layers. *International Journal of Heat and Fluid Flow*, 60:161–175, 2016.

[24] B. E. Launder and D. B. Spalding. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3(2):269–289,

Mar. 1974.

[25] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, Oct. 2016.

[26] O. Ltd. *OpenFOAM – The Open Source CFD Toolbox, User Guide*. ESI Group, Bracknell, UK, 2024.

[27] E. Malkiel and R. E. Mayle. Transition in a separation bubble. *Journal of Turbomachinery*, 118(4):752–759, Oct. 1996.

[28] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, Aug. 1994.

[29] F. R. Menter, R. B. Langtry, S. R. Likki, Y. B. Suzen, P. G. Huang, and S. Völker. A correlation-based transition model using local variables — Part I: Model formulation. *Journal of Turbomachinery*, 128(3):413–422, Mar. 2004.

[30] F. R. Menter, P. E. Smirnov, T. Liu, and R. Avancha. A one-equation local correlation-based transition model. *Flow, Turbulence and Combustion*, 95(4):583–619, July 2015.

[31] R. Michel. Etude de la transition dans la couche limite: Détermination du point de transition et calcul de la trainée des profils d'ailes en incompressible. ONERA report, ONERA, 1954.

[32] M. Monfaredi. *Continuous Adjoint-Based Aeroacoustic Shape Optimization by Coupling the Ffowcs-Williams Hawkings Analogy with URANS Equations*. PhD thesis, National Technical University of Athens, 2021.

[33] E. M. Papoutsis-Kiachagias. *Adjoint Methods for Turbulent Flows, Applied to Shape or Topology Optimization and Robust Design*. PhD thesis, National Technical University of Athens, Athens, Greece, 2013.

[34] B. K. Petersen, M. Landajuela, T. N. Mundhenk, C. P. Santiago, S. K. Kim, and J. T. Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients, 2019.

[35] A. A. Petrocheilos. Data-Driven Eddy Viscosity Adaptation via the Continuous Adjoint Method. Applications in Aerodynamic Shape Optimization. Diploma thesis, National Technical University of Athens, July 2026.

[36] M. G. H. Piotrowski and D. W. Zingg. Smooth local correlation-based transition model for the Spalart–Allmaras turbulence model. *AIAA Journal*, 59(2):474–492, Feb. 2021.

[37] L. Prandtl. 7. Bericht über Untersuchungen zur ausgebildeten Turbulenz.

ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik, 5(2):136–139, Jan. 1925.

[38] O. Reynolds. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society of London. (A.)*, 186:123–164, 1895.

[39] H. Schlichting and K. Gersten. *Boundary-Layer Theory*. Springer Berlin Heidelberg, 2017.

[40] P. J. Schmid and D. S. Henningson. *Stability and Transition in Shear Flows*. Applied Mathematical Sciences. Springer New York, 2001.

[41] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[42] M. T. Schobeiri. *Advanced Fluid Mechanics and Heat Transfer for Engineers and Scientists*. Springer International Publishing, 2022.

[43] J. Smagorinsky. General circulation experiments with the primitive equations: I. The basic experiment. *Monthly Weather Review*, 91(3):99–164, Mar. 1963.

[44] D. M. Somers. Design and experimental results for a natural-laminar-flow airfoil for general aviation applications. NASA Technical Paper 1861, NASA Langley Research Center, Hampton, Virginia, June 1981.

[45] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. AIAA Paper 1992-439, 30th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, USA, Jan. 1992.

[46] P. Spalart, W.-H. Jou, M. Strelets, and S. Allmaras. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. Jan. 1997.

[47] T. Stephens. gplearn: Genetic Programming in Python, with a scikit-learn inspired API. `https://github.com/trevorstephens/gplearn`, 2015. Accessed: 2024-05-22.

[48] X. S. Trompoukis. *Numerical Solution of Aerodynamic-Aeroelastic Problems on Graphics Processing Units (in Greek)*. PhD thesis, National Technical University of Athens, 2012.

[49] K. Tsiakas. *Development of Shape Parameterization Techniques, a Flow Solver and Its Adjoint, for Optimization on GPUs. Turbomachinery and External Aerodynamics Applications*. PhD thesis, National Technical University of Athens, 2019.

[50] C. Tsolakos. Eddy viscosity adaptation using the Continuous Adjoint Method with application in Aerodynamic Shape Optimization. Diploma thesis, National

Technical University of Athens, Feb. 2026.

[51] J. van Ingen. The e^N method for transition prediction. Historical review of work at TU Delft. In *38th Fluid Dynamics Conference and Exhibit*. American Institute of Aeronautics and Astronautics, June 2008.

[52] J. Weatheritt and R. Sandberg. A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. *Journal of Computational Physics*, 325:22–37, Nov. 2016.

[53] D. C. Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal*, 26(11):1299–1310, Nov. 1988.

[54] C. Wu, S. Zhang, and Y. Zhang. Development of a generalizable data-driven turbulence model: Conditioned field inversion and symbolic regression. *AIAA Journal*, 63(2):687–706, Feb. 2025.

[55] C. Wu and Y. Zhang. Data-driven turbulence modeling using field inversion and deep symbolic regression. *Proceedings of the Aerospace Europe Conference – EUCASS – CEAS – 2023*, 2023.

[56] L. Wu and Z. Xiao. Overshoot-resolved transition modeling based on field inversion and symbolic regression, 2025.

[57] T. Zervogiannis. *Optimization Methods in Aerodynamics and Turbomachinery Based on the Adjoint Technique, Hybrid Grids and the Exact Hessian Matrix (in Greek)*. PhD thesis, National Technical University of Athens, 2011.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής &
Βελτιστοποίησης

Συμβολική Παλινδρόμηση στη Μοντελοποίηση της Τύρβης και της Μετάβασης στην Αεροδυναμική Βελτιστοποίηση Μορφής

Διπλωματική Εργασία

Καλλιόπη Μπατσή

Επιβλέποντες: Κυριάκος Χ. Γιαννακόγλου, Καθηγητής ΕΜΠ
Δρ. Β. Ασούτη, Εντεταλμένη Διδάσκουσα ΕΜΠ

Αθήνα, 2026

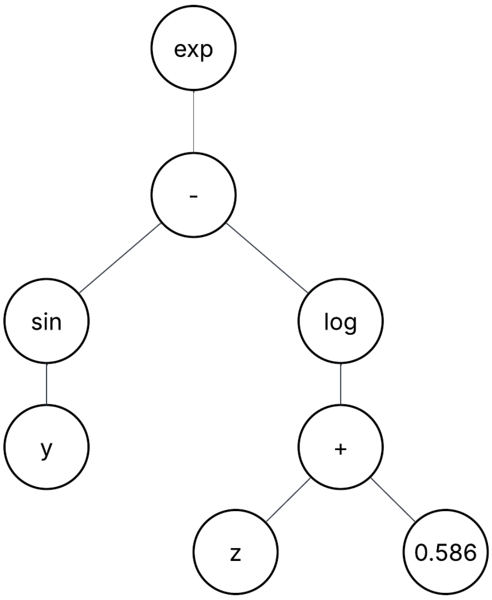
Εισαγωγή

Η μελέτη της τύρβης και της μετάβασης από τη στρωτή στην τυρβώδη ροή με προσομοιώσεις υψηλής πιστότητας συνοδεύεται από υψηλό υπολογιστικό κόστος σε βιομηχανικές εφαρμογές. Παράλληλα, τα παραδοσιακά αλγεβρικά μοντέλα τύρβης δεν παρέχουν ικανοποιητική ακρίβεια σε περίπλοκες ροές, ενώ τα περισσότερο ακριβή μοντέλα μίας ή δύο εξισώσεων (Spalart-Allmaras, $\gamma - Re_\theta$) εισάγουν επιπλέον μερικές διαφορικές εξισώσεις στη διαδικασία επίλυσης, αυξάνοντας το υπολογιστικό κόστος. Τα υποκατάστατα μοντέλων που βασίζονται σε δεδομένα αξιοποιούν έως τώρα κύριως τα Βαθιά Νευρωνικά Δίκτυα, τα οποία επιδεικνύουν μεγάλη ακρίβεια, ωστόσο ταυτόχρονα παραμένουν ευαίσθητα στην υπερπροσαρμογή και στην αδυναμία γενίκευσης. Σημαντικότερη παράμετρο αποτελεί το γεγονός ότι λειτουργούν ως "μαύρο κουτί", μη αφήνοντας περιθώριο για αναλυτική ερμηνεία. Αντικείμενο της διπλωματικής αυτής εργασίας αποτελεί η κατασκευή ερμηνεύσιμων αλγεβρικών υποκατάστατων μοντέλων τύρβης και μετάβασης, μέσω Συμβολικής Παλινδρόμησης. Για τον σκοπό αυτό, αξιοποιείται ο εξελικτικός αλγόριθμος Συμβολικής Παλινδρόμησης PySR, ο οποίος τροφοδοτείται με δεδομένα προσομοιώσεων Υπολογιστικής Ρευστοδυναμικής και καταλήγει στην αναλυτική συσχέτιση της εξόδου - η οποία είναι είτε η τυρβώδης συνεκτικότητα ν_t είτε ο παράγοντας διάλειψης γ για την τύρβη και τη μετάβαση αντίστοιχα - με τις εισόδους, οι οποίες είναι ροϊκές και γεωμετρικές ποσότητες. Τα υποκατάστατα μοντέλα μετάβασης αξιολογούνται ως προς την πρόβλεψη του σημείου μετάβασης και των αεροδυναμικών συντελεστών, ενώ τα υποκατάστατα μοντέλα τύρβης χρησιμοποιούνται για τη βελτίωση της μοντελοποίησης περιοχών ανακυκλοφορίας, και τέλος, εντάσσονται σε βρόχο Βελτιστοποίησης Σχήματος.

Συμβολική Παλινδρόμηση

Η Συμβολική Παλινδρόμηση (Symbolic Regression - SR) εντάσσεται στην Ερμηνεύσιμη Μηχανική Μάθηση, καθώς πρόκειται για αλγορίθμους που αναζητούν αναλυτική συσχέτιση μεταξύ μιας εξόδου Y για δεδομένες εισόδους $\vec(X)$. Κάθε τέτοιος αλγόριθμος βασίζεται στα λεγόμενα "δέντρα εξισώσεων", όπως αυτό του Σχ. 1 τα οποία συνδέουν μεταξύ τους κόμβους που περιέχουν είτε εισόδους, είτε μαθηματικούς τελεστές είτε αριθμητικούς συντελεστές. Η πολυπλοκότητα μιας έκφρασης ορίζεται ως ο αριθμός των κόμβων στο δέντρο.

Για την αναζήτηση χρησιμοποιούνται είτε Εξελικτικοί Αλγόριθμοι (ΕΑ) είτε Βαθιά Νευρωνικά Δίκτυα (ΒΝΔ), οι οποίοι στοχεύουν συγχρόνως στην ελαχιστοποίηση του σφάλματος πρόβλεψης και της πολυπλοκότητας, με αποτέλεσμα να παράγουν ένα μέτωπο Pareto βέλτιστων αλγεβρικών εκφράσεων. Στην διπλωματική εργασία αξιοποιείται η βιβλιοθήκη PySR σε γλώσσα Python, η οποία αποτελεί έναν τροποποιημένο ΕΑ για τη διαδικασία της βελτιστοποίησης, με μεταβλητές σχεδιασμού τις ποσότητες εντός των κόμβων του δέντρου των εκφράσεων.



Σχήμα 1: Παράδειγμα δέντρου εξίσωσης. Οι κόμβοι περιέχουν τις μεταβλητές σχεδιασμού του αλγορίθμου Συμβολικής Παλινδρόμησης. Η πολυπλοκότητα ορίζεται ως ο αριθμός των κόμβων.

Εφαρμογές Συμβολικής Παλινδρόμησης στη Μοντελοποίηση της Μετάβασης

Για τη μοντελοποίηση της μετάβασης η διαδικασία που ακολουθείται είναι η εξής:

- Προσομοίωση της ροής με λογισμικό Υπολογιστικής Ρευστοδυναμικής (εδώ ο οικείος κώδικας PUMA) με μοντέλα Spalart-Allmaras & $\gamma - Re_\theta$ για την παραγωγή των δεδομένων που θα τροφοδοτηθούν στον αλγόριθμο Συμβολικής Παλινδρόμησης.
- Χρήση αλγορίθμου SR για συσχέτιση της εξόδου γ με τις εισόδους (ροϊκές και γεωμετρικές ποσότητες). Ο αλγόριθμος παράγει ένα σύνολο βέλτιστων αλγεβρικών εκφράσεων για τον παράγοντα γ συναρτήσει των εισόδων. Από το σύνολο αυτό των παραγόμενων τύπων, η τελική έκφραση επιλέγεται επιδιώκοντας την πιο αποδοτική ισορροπία μεταξύ της ελαχιστοποίησης της απόκλισης και της διατήρησης χαμηλής πολυπλοκότητας, ελέγχοντας παράλληλα την αριθμητική της ευστάθεια.
- Ενσωμάτωση του αλγεβρικού υποκατάστατου μοντέλου μετάβασης στον επιλύτη RANS, αντί για το μοντέλο $\gamma - Re_\theta$ και αξιολόγηση αποτελεσμάτων.

Εφαρμογή Α: Αεροτομή NLF(1)-0416

Ο αλγόριθμος SR τροφοδοτείται με το πεδίο γ που προκύπτει γύρω από την αεροτομή NLF(1)-0416 σε συνθήκες $M = 0.1, AoA = 3.50^\circ$ και παράγει την αναλυτική σχέση 1 για την έξοδο:

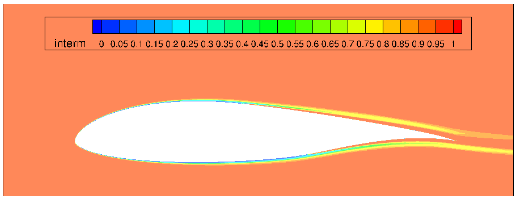
$$\gamma = 0.8074^{\phi \cdot \tau^{(10.3013 \cdot \kappa \cdot \lambda)}} \tag{1}$$

όπου $\phi = c_6^{0.6444}$, $\lambda = c_7^{0.3227}$, $\kappa = c_6^{0.1337}$, $\tau = c_8^2 \cdot c_1$ και $c_1 = d_{wall}/c$, $c_6 = \Omega c/V_\infty$, $c_7 = \nu_t/V_\infty c$, $c_8 = M$.

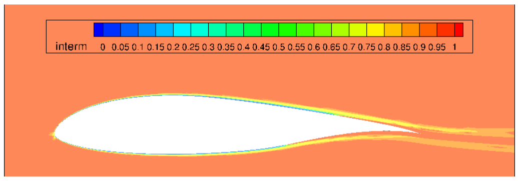
Το υποκατάστατο ενσωματώνεται στον επιλύτη PUMA και προκύπτουν οι τιμές ποσοστιαίων αποκλίσεων για τους αεροδυναμικούς συντελεστές, που απεικονίζονται στον Πίνακα 1. Τα αντίστοιχα πεδία απεικονίζονται στο Σχ. 2, ενώ οι κατανομές του συντελεστή τριβής στο Σχ. 3. Το υποκατάστατο παρουσιάζει μεγαλύτερη ακρίβεια στην πρόβλεψη τόσο των αεροδυναμικών συντελεστών όσο και του σημείου μετάβασης, όσο δοκιμάζεται σε γωνίες προσβολής κοντά στη γωνία εκπαίδευσης, ενώ η πιστότητά του ελαττώνεται όσο ελαττώνεται η γωνία δοκιμής.

Γωνία	C_D		C_L	
Προσβολής (°)	$\gamma - Re_\theta$	SR (Απόκλ. %)	$\gamma - Re_\theta$	SR (Απόκλ. %)
0.00	0.0057	0.0038 (-33.23)	0.4835	0.5094 (5.37)
2.03	0.0063	0.0042 (-33.30)	0.7200	0.7543 (4.77)
3.00	0.0068	0.0067 (-0.50)	0.8308	0.8315 (0.08)
3.50	0.0070	0.0069 (-0.70)	0.8878	0.8888 (0.11)

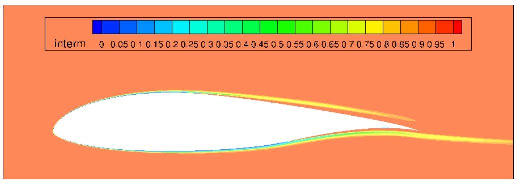
Πίνακας 1: Εξ. 1 - Εκπαίδευση σε γωνία προσβολής 3.50°: Τιμές PUMA με $\gamma - Re_\theta$ Piotrowski-Zingg, τιμές αλγεβρικού υποκατάστατου (SR) και σχετική απόκλιση αεροδυναμικών συντελεστών για διάφορες γωνίες προσβολής.



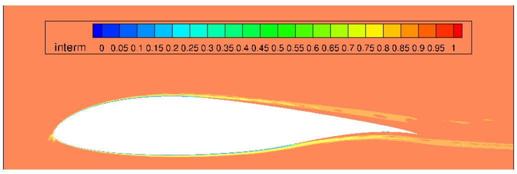
(α΄) Πεδίο $\gamma - Re_\theta$ AoA = 0°.



(β΄) Πεδίο SR AoA = 0°.



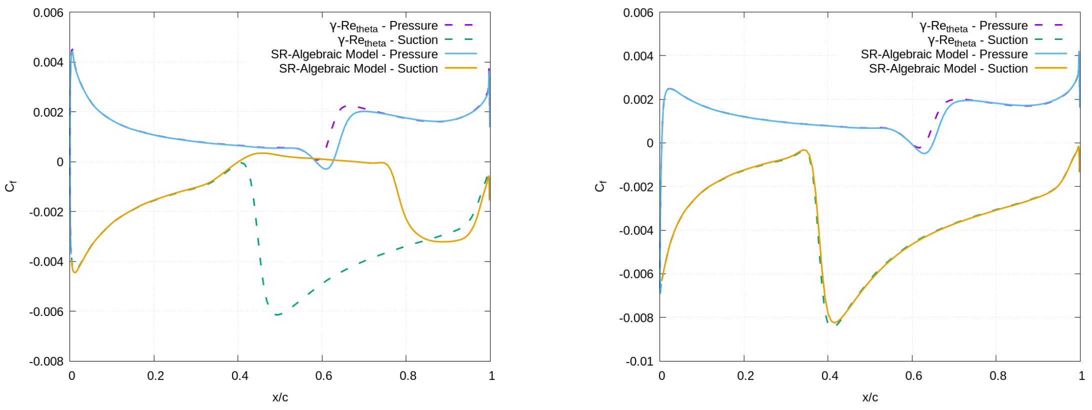
(γ΄) Πεδίο $\gamma - Re_\theta$ AoA = 3.50°.



(δ΄) Πεδίο SR AoA = 3.50°.

Σχήμα 2: Πεδία διάλειψης ροής γ με χρήση του μοντέλου $\gamma - Re_\theta$ και της Εξ.1 στην προσομοίωση RANS σε διαφορετικές γωνίες προσβολής.

Για τις μικρότερες γωνίες προσβολής, όπου η απόκλιση των αεροδυναμικών συντελεστών είναι αυξημένη, κατασκευάζεται ένας βρόχος διόρθωσης, με τον οποίο οι αποκλίσεις των συντελεστών C_D και C_L για γωνία προσβολής 2.03° μειώνονται στο -5.47% και 0.28% αντίστοιχα.



(α΄) AoA = 0.00°. (β΄) AoA = 3.50°.

Σχήμα 3: NLF(1)-0416: Κατανομή συντελεστή τριβής (C_f) με χρήση του μοντέλου $\gamma - Re_\theta$ και της Εξ.1.

Ακολουθεί η δημιουργία ενός υποκατάστατου, εκπαιδευμένου από δύο πεδία γ που προκύπτουν σε συνθήκες $M = 0.1$ και γωνίες προσβολής $2.03°$ και $3.50°$. Προκύπτει η Εξ. 2 και τα αποτελέσματα των αεροδυναμικών συντελεστών συνοψίζονται στον Πίνακα 2. Το νέο υποκατάστατο, αν και λιγότερο ακριβές, φαίνεται συνεπέστερο σε μεγαλύτερο εύρος γωνιών και είναι το εξής:

$$\gamma = \left(c_7^{c_2-\Phi}\right)^{\Psi} \tag{2}$$

όπου $\Phi = (0.9416^{c_6})^{(0.1721+c_0)}$, $\Psi = \left[\left(\frac{c_7}{c_6}\right)^{c_7^{0.2657}}\right]^{\exp(\cos c_4+\cos c_8)}$ και $c_0 = x/c$, $c_2 = \rho/\rho_\infty$, $c_4 = V_y/V_\infty$, $c_6 = \Omega c/V_\infty$, $c_7 = \nu_t/V_\infty c$, $c_8 = M$.

Γωνία Προσβολής (°)	C_D		C_L	
	$\gamma - Re_\theta$	SR (Απόκλ. %)	$\gamma - Re_\theta$	SR (Απόκλ. %)
0.00	0.0057	0.0037 (-35.03)	0.4835	0.4944 (2.27)
2.03	0.0063	0.0049 (-22.75)	0.7200	0.7063 (-1.90)
3.00	0.0068	0.0043 (-37.00)	0.8308	0.8703 (4.96)
3.50	0.0070	0.0063 (-9.31)	0.8878	0.8912 (0.38)

Πίνακας 2: Εξ. 2 - Εκπαίδευση σε γωνίες προσβολής $2.03°$ & $3.50°$: Τιμές PUMA με $\gamma - Re_\theta$ Piotrowski-Zingg, τιμές αλγεβρικού υποκατάστατου (SR) και σχετική απόκλιση αεροδυναμικών συντελεστών για διάφορες γωνίες προσβολής.

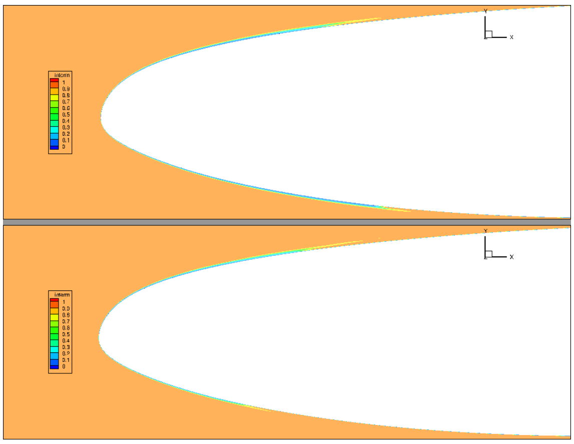
Εφαρμογή Β: Αεροτομή ONERA OAT15A

Η διαδικασία που ακολουθείται είναι όμοια, με την εκπαίδευση του νέου υποκατάστατου (Εξ. 3) να γίνεται από το πεδίο γ της αεροτομής OAT15A σε συνθήκες M=0.73, $P_{stat} = 0.78\ kPa$, $T = 271\ K$ και γωνία προσβολής $2°$. Το υποκατάστατο επιτυγχάνει ιδιαίτερα ακριβή πρόβλεψη των αεροδυναμικών συντελεστών της αεροτομής σε εύρος γωνιών προσβολής από 1° έως 3.50°, όπως φαίνεται στο Σχ. 5 και περιγράφεται από την παρακάτω εξίσωση:

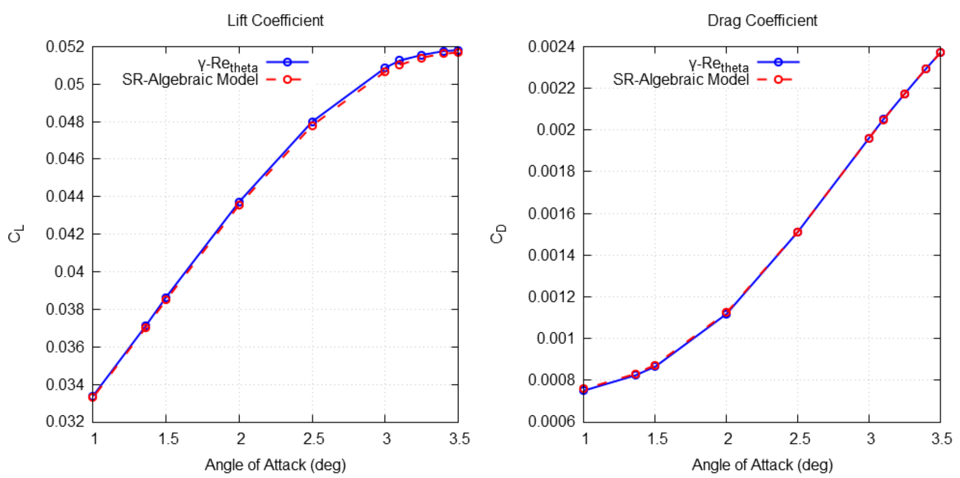
$$\gamma = \sqrt{\theta} \tag{3}$$

όπου $\kappa = \frac{2.8952558}{c_6+2.4448037}$, $\psi = c_7^{c_7}$, $\phi = \psi^{(c_5 \cdot 305315.44)}$, $\delta = \kappa^{\phi}$, $\lambda = \frac{c_2^{1.3415653}}{e^{c_0}}$, $\zeta = (c_0 \cdot c_3) \cdot 3.7659185$, $\eta = \lambda^{\zeta}$, $\theta = \delta^{\eta}$ και $c_0 = x/c$, $c_3 = V_x/V_\infty$, $c_5 = p/p_\infty$, $c_6 = \Omega c/V_\infty$, $c_7 = \nu_t/V_\infty c$.

Το Σχ. 4 καταδεικνύει ότι υποεκτιμάται το σημείο μετάβασης στην πλευρά πίεσης, κάτι που συμβαίνει συστηματικά για όλες τις γωνίες προσβολής που δοκιμάστηκαν.



Σχήμα 4: Ακμή προσβολής αεροτομής OAT15A: Πεδίο διάλειψης ροής σε γωνία προσβολής 2.50°. Πάνω πεδίο: Με χρήση του μοντέλου μετάβασης Piotrowski-Zingg, Κάτω πεδίο: Με χρήση της Εξ. 5.4 ως υποκατάστατο του μοντέλου μετάβασης στο λογισμικό PUMA.



Σχήμα 5: OAT15A Airfoil: Drag and Lift Coefficients in different AoA with Equation 5.4.

Εφαρμογή C: Αεροτομές NLF(1)-0416 & ONERA OAT15A

Για την κατασκευή ενός αλγεβρικού υποκατάστατου, το οποίο θα είναι χρήσιμο όχι μόνο σε μία γεωμετρία, ο αλγόριθμος SR τροφοδοτήθηκε με δύο πεδία του παράγοντα διάλειψης γ, ένα από την αεροτομή NLF(1)-0416 και ένα από την OAT15A, από προσομοίωση που έγινε για M=0.1 και M=0.73 αντίστοιχα και γωνία προσβολής 2° και για τις δύο αεροτομές. Οι αποκλίσεις πρόβλεψης των αεροδυναμικών συντελεστών συνοψίζονται στον Πίνακα 3 και το αλγεβρικό υποκατάστατο μοντέλο μετάβασης που προέκυψε περιγράφεται από την ακόλουθη εξίσωση:

$$\gamma = \phi^{\psi} \tag{4}$$

όπου $\kappa = c_7^{0.033316202}$, $\phi = \sin(\sin(\kappa)) + 0.41349724$, $\theta = \frac{c_6}{0.22231957}$, $\zeta = 0.45793113 - c_7^{0.06932941}$, $\lambda = \theta^{\zeta}$, $\tau = \frac{c_3}{\cos(c_2)} + 1.8690305$, $\psi = \lambda^{\tau}$ και $c_2 = \rho/\rho_\infty$, $c_3 = V_x/V_\infty$, $c_6 = \Omega c/V_\infty$, $c_7 = \nu_t/V_\infty c$.

Αεροτομή	Γωνία (°)	C_D		C_L	
		$\gamma - Re_\theta$	SR (Απόκλ. %)	$\gamma - Re_\theta$	SR (Απόκλ. %)
NLF(1)-0416	2.00	0.00631	0.00639 (1.41)	0.7164	0.7158 (-0.09)
	2.50	0.00648	0.00697 (7.46)	0.7743	0.7631 (-1.44)
	3.00	0.00677	0.00688 (1.72)	0.8308	0.8281 (-0.33)
OAT15A	2.00	0.00112	0.00113 (0.89)	0.0437	0.0457 (4.42)
	2.50	0.00151	0.00158 (4.79)	0.0480	0.0505 (5.29)
	3.00	0.00196	0.00212 (7.93)	0.0509	0.0542 (6.53)

Πίνακας 3: Εξ. 4 - Συνδυασμένο μοντέλο: Τιμές PUMA με $\gamma - Re_\theta$ Piotrowski-Zingg, τιμές αλγεβρικού υποκατάστατου (SR) και σχετική απόκλιση αεροδυναμικών συντελεστών για τις αεροτομές NLF(1)-0416 και OAT15A.

Εφαρμογές Συμβολικής Παλινδρόμησης στη Μοντελοποίηση της Τύρβης

Για τη μοντελοποίηση της τύρβης η διαδικασία που ακολουθείται στο περιβάλλον OpenFOAM διαμορφώνεται ως εξής:

- Εφαρμογή μεθοδολογίας EVA για την κατασκευή διορθωτικού πεδίου τυρβώδους συνεκτικότητας ν_t για βελτιωμένη μοντελοποίηση της περιοχής αποκόλλησης στο πεδίο ταχυτήτων.
- Χρήση αλγορίθμου SR για συσχέτιση της εξόδου ν_t -του πεδίου EVA- με τις εισόδους (ροϊκές και γεωμετρικές ποσότητες). Ο αλγόριθμος παράγει ένα σύνολο βέλτιστων αλγεβρικών εκφράσεων για την τυρβώδη συνεκτικότητα ν_t.
- Ενσωμάτωση του αλγεβρικού υποκατάστατου μοντέλου τύρβης στον επιλύτη RANS, αντί για το μοντέλο Spalart-Allmaras και αξιολόγηση αποτελεσμάτων.
- Χρήση αλγεβρικού υποκατάσταση για βελτιστοποίηση σχήματος με τη συνεχή συζυγή μέθοδο.

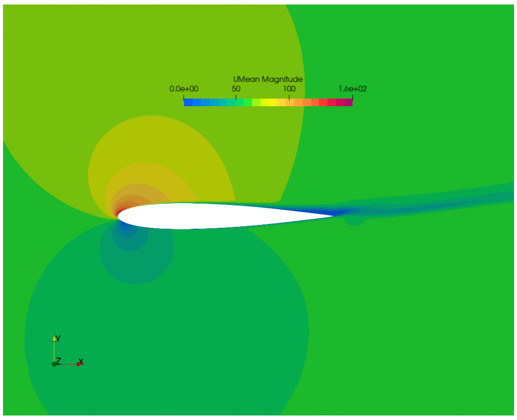
Εφαρμογή D: Αεροτομή NACA 0012

Το διορθωτικό πεδίο ν_t που προέκυψε από εφαρμογή της μεθοδολογίας EVA για συνθήκες $Re = 10^6$ και $AoA = 12^\circ$ τροφοδοτήθηκε στον αλγόριθμο SR και προέκυψε το αλγεβρικό υποκατάστατο μοντέλο μετάβασης που περιγράφεται στην ακόλουθη εξίσωση:

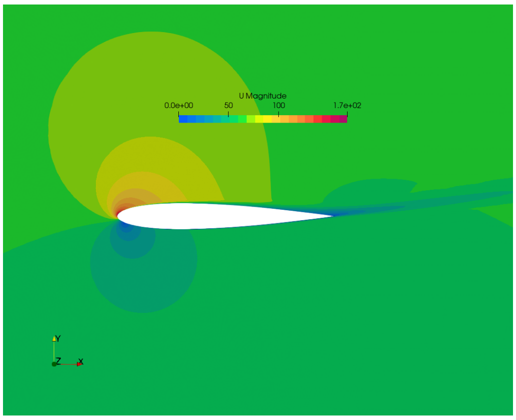
$$\nu_t = 0.013 \cdot \left[\frac{c_0 + c_7}{c_4 \left(\frac{11.202037}{c_3} \right) + 1.7163204} + 0.032510918 \right] \tag{5}$$

όπου $c_0 = x/c$, $c_3 = S \cdot /V_\infty$, $c_4 = V_x/V_\infty$, $c_7 = d_{wall}/c$.

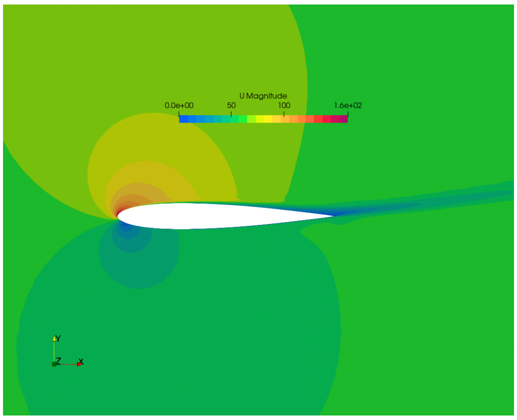
Τα πεδία ταχυτήτων παρουσιάζονται στο Σχ. 6 και το πεδίο DES θεωρείται ότι επιτυγχάνει την πιστότερη μοντελοποίηση της μετάβασης. Τα πεδία EVA και RANS-SR έχουν παρόμοια, ικανοποιητικότερα σε σχέση με το πεδίο RANS-SA αποτελέσματα στην μοντελοποίηση της αποκόλλησης.



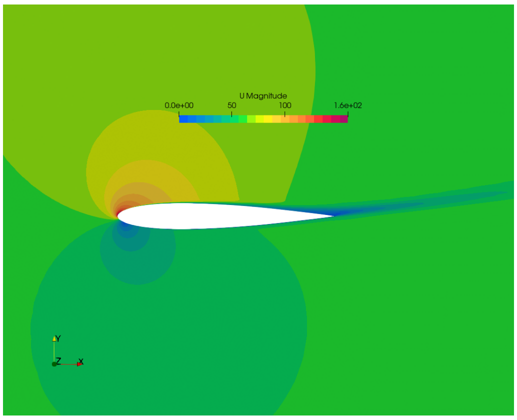
(α΄) Μέσο πεδίο ταχυτήτων DES.



(β΄) Πεδίο ταχυτήτων RANS-SA.



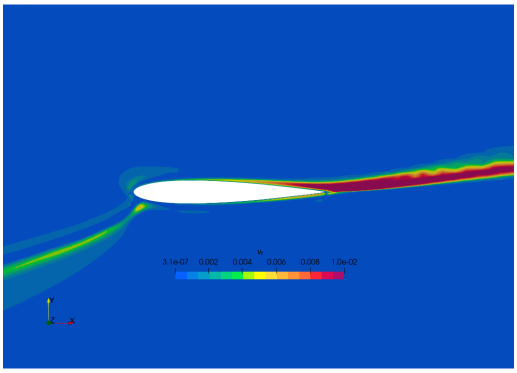
(γ΄) Πεδίο ταχυτήτων EVA.



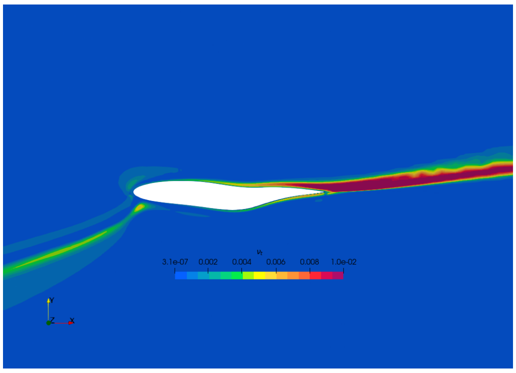
(δ΄) Πεδίο ταχυτήτων RANS-SR.

Σχήμα 6: Πεδία ταχυτήτων των μεθόδων DES, RANS-SA, EVA, RANS-SR για την αεροτομή NACA 0012.

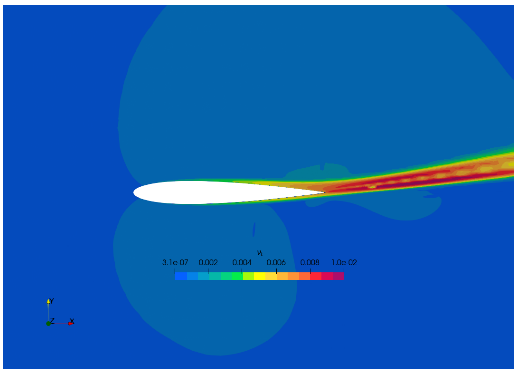
Για τη βελτιστοποίηση μορφής με στόχο την ελαχιστοποίηση του C_D συγκρίνονται η μέθοδος Frozen Eddy Viscosity (FEV) όπου το πεδίο ν_t της EVA παραμένει ”παγωμένο" κατά τη διάρκεια της βελτιστοποίησης, και η μέθοδος SR όπου το πεδίο ν_t του πρωτεύοντος προβλήματος ανανεώνεται σε κάθε κύκλο βελτιστοποίησης με βάση την Εξ. 5. Τα αποτελέσματα καταδεικνύουν ότι η διαδικασία της βελτιστοποίησης επωφελείται από τη δυναμική ανανέωση του πεδίου, λόγω των αλλαγών της γεωμετρίας, έναντι της χρήσης του "παγωμένου" πεδίου, όπως φαίνεται στον Πίνακα 4.



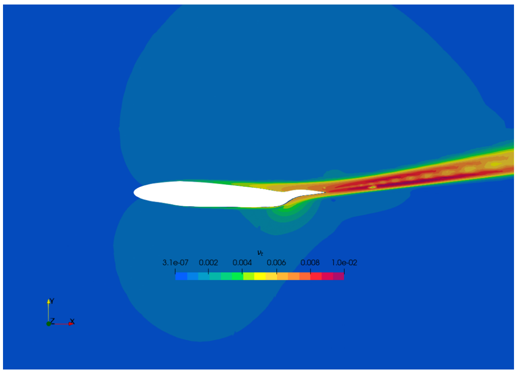
(α΄) Αρχική γεωμετρία με το πεδίο EVA.



(β΄) Τελική γεωμετρία με το πεδίο EVA.



(γ΄) Αρχική γεωμετρία με το πεδίο SR.



(δ΄) Τελική γεωμετρία με το πεδίο SR.

Σχήμα 7: Βελτιστοποίηση σχήματος αεροτομής NACA 0012: Αρχικά και τελικά πεδία τυρβώδους συνεκτικότητας με χρήση των μεθόδων EVA και RANS-SR.

Method	C_D
EVA	0.0239
SR	0.0233

Πίνακας 4: Βελτιστοποίηση σχήματος αεροτομής NACA 0012: Συντελεστής αντίστασης των τελικών γεωμετριών -όπως επαναξιολογήθηκαν με DES- βελτιστοποιημένων με τη μέθοδο EVA και τη σταθερή έκφραση του υποκατάστατου SR.

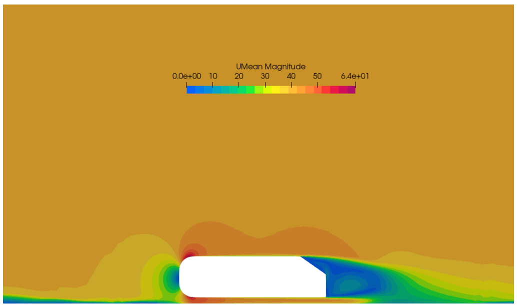
Εφαρμογή Ε: Ahmed Body

Όμοια διαδικασία ακολουθείται και για μια τριδιάστατη εφαρμογή, το Ahmed Body με γωνία 35°. Το αλγεβρικό υποκατάστατο τύρβης που προκύπτει με βάση το πεδίο EVA περιγράφεται από την ακόλουθη εξίσωση:

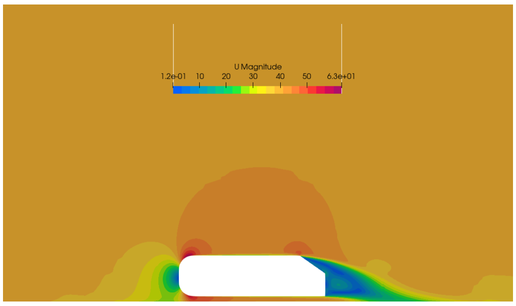
$$\nu_t = 0.13 \cdot (c_5 - 1.0012314) \cdot (c_9 - 5234.8203) \tag{6}$$

όπου $c_5 = V_x/V_\infty$ και $c_9 = d_{wall}/c$.

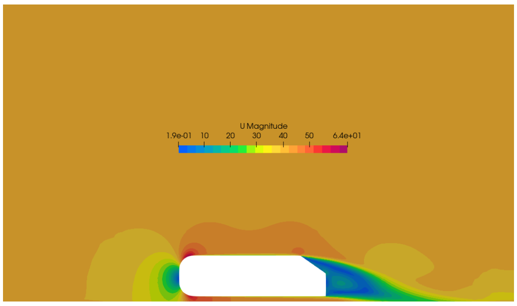
Όπως και στην προηγούμενη περίπτωση, με περισσότερο αξιόπιστη την προσομοίωση DES, τα αποτελέσματα της μοντελοποίησης της περιοχής αποκόλλησης είναι πιο ικανοποιητικά με τις μεθόδους EVA και RANS-SR, σε σύγκριση με τη μέθοδο RANS-SA. Τα παραπάνω φαίνονται στο Σχ. 8.



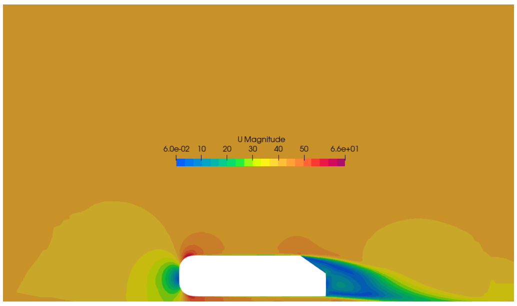
(α΄) Μέσο πεδίο ταχυτήτων DES.



(β΄) Πεδίο ταχυτήτων RANS-SA.



(γ΄) Πεδίο ταχυτήτων EVA.



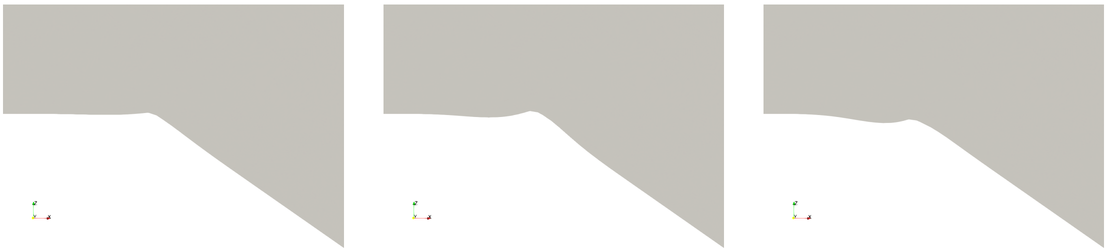
(δ΄) Πεδίο ταχυτήτων RANS-SR.

Σχήμα 8: Πεδία ταχυτήτων DES, RANS-SA, EVA, RANS-SR για το Ahmed Body.

Για τη βελτιστοποίηση μορφής συγκρίνονται οι μέθοδοι FEV, SR-Fixed, όπου το αλγεβρικό υποκατάστατο χρησιμοποιείται απλώς για ανανέωση του πεδίου ν_t στο πρωτεύον πρόβλημα, και SR-Differentiated, όπου γίνεται παραγώγιση του αλγεβρικού υποκατάστατου ώστε να εισαχθούν οι επιπλέον όροι στις συζυγείς εξισώσεις. Όπως φαίνεται στον Πίνακα 5 μέθοδος SR-Fixed παρουσιάζει τα καλύτερα αποτελέσματα τιμής της τελικής αντικειμενικής συνάρτησης σε σύγκριση με τις άλλες δύο μεθόδους, όταν γίνει επαναξιολόγηση της τελικής γεωμετρίας με DES. Ωστόσο, η καλύτερη τιμή αντικειμενικής συνάρτησης όταν αυτή αξιολογείται με το ίδιο το μοντέλο που χρησιμοποιείται στη διαδικασία βελτιστοποίησης, προκύπτει από τη μέθοδο SR-Differentiated, γεγονός απολύτως λογικό εφόσον πρόκειται για ολοκληρωμένη ένταξη του υποκατάστατου στη συζυγή μέθοδο. Ο λόγος που στην επαναξιολόγηση με DES δε φαίνεται αντίστοιχη συμπεριφορά είναι πιθανότατα ότι το υποκατάστατο, έχοντας εκπαιδευτεί από τη ροή γύρω απο την αρχική γεωμετρία, χάνει την πιστότητά του όσο η γεωμετρία αλλάζει και δεν προσφέρει ακριβή πρόβλεψη του συντελεστή αντίστασης.

Method	C_D Optimization	C_D DES Reevaluation
EVA	0.304	0.273
SR-Fixed	0.305	0.269
SR-Diff	0.299	0.289

Πίνακας 5: Βελτιστοποίηση σχήματος του Ahmed Body: Συντελεστής αντίστασης των τελικών γεωμετριών βελτιστοποιημένων με τη μέθοδο EVA και τη σταθερή και παραγωγισμένη έκφραση του υποκατάστατου SR. Στήλη 2: Τιμή που προκύπτει από αξιολόγηση με το ίδιο το μοντέλο τύρβης που αξιοποιείται στη βελτιστοποίηση. Στήλη 3: Επαναξιολόγηση τελικής γεωμετρίας με DES.



(α′) Τελική γεωμετρία με τη μέθοδο EVA.

(β′) Τελική γεωμετρία με τη μέθοδο RANS-SR-Fixed.

(γ′) Τελική γεωμετρία με τη μέθοδο RANS-SR-Differentiated.

Σχήμα 9: Βελτιστοποίηση σχήματος του Ahmed Body: Τελική γεωμετρία που παρήχθη με τις μεθόδους EVA και RANS-SR.

Συμπεράσματα και Μελλοντική Έρευνα

Η χρήση της Συμβολικής Παλινδρόμησης (SR) για τη δημιουργία ερμηνεύσιμων, αλγεβρικών υποκατάστατων μοντέλων μετάβασης και τύρβης στην Υπολογιστική Ρευστοδυναμική αποτελεί μια εξαιρετικά υποσχόμενη προσέγγιση. Τα εξαγόμενα μοντέλα επιτυγχάνουν ικανοποιητική πρόβλεψη αεροδυναμικών συντελεστών, βελτιωμένη μοντελοποίηση των περιοχών αποκόλλησης και αποτελεσματική ένταξη σε διαδικασίες βελτιστοποίησης σχήματος. Παρόλα αυτά, η ακρίβεια και η ικανότητα γενίκευσής τους εξαρτώνται άμεσα από την ποικιλομορφία των δεδομένων εκπαίδευσης, καθώς η πιστότητά τους φθίνει όταν μεταβάλλονται σημαντικά οι συνθήκες ροής ή η γεωμετρία. Σε επίπεδο μελλοντικής έρευνας, προτείνεται αφενός η δημιουργία ευρύτερων βάσεων δεδομένων ροής για την ανάπτυξη πιο γενικευμένων μοντέλων, και αφετέρου η ενσωμάτωση φυσικών και αριθμητικών περιορισμών στη συνάρτηση κόστους του αλγορίθμου. Αυτό θα διασφαλίσει εξαρχής την αριθμητική ευστάθεια των παραγόμενων εξισώσεων κατά την επίλυση, καταργώντας την ανάγκη για τη χρονοβόρα, χειροκίνητη διαδικασία επιλογής εξισώσεων μέσω δοκιμής και σφάλματος.