Apprentissage Bayesien parcimonieux et agrégation adaptative de modèles de turbulence

Soufiane Cherroud

HESAM UNIVERSITÉ

Arts et Métiers Sciences et Technologies

ÉCOLE DOCTORALE SCIENCES ET MÉTIERS DE L'INGÉNIEUR
[Laboratoire DynFluid - Campus de Paris]

THÈSE

présentée par : **Soufiane CHERROUD**
soutenue le : **1er Décembre 2023**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée à : **École Nationale Supérieure d'Arts et Métiers**
Spécialité : **Mécanique des Fluides et Énergétique**

Sparse Bayesian Learning and Adaptative Aggregation of Data-driven Turbulence Models

THÈSE dirigée par :
[Pr. Xavier GLOERFELT]

et co-encadrée par :
[Dr. Xavier MERLE]

Jury		
Pr.	Azeddine KOURTA	President
Associate Professor	Ricardo VINUESA	Reviewer
Professor	Sharath GIRIMAJI	Reviewer
Professor	Paola CINNELLA	Examinator
Professor	Azeddine KOURTA	Examinator
Professor	Xavier GLOERFELT	Director
Associate Professor	Xavier MERLE	Co-supervisor

THÈSE


وقل رب زدني علما

Acknowledgements

I would like to express my sincere gratitude to those who have contributed to the successful completion of this doctoral thesis. Firstly, I extend my deepest appreciation to my supervisors, Pr. Paola Cinnella, Mr. Xavier Merle and Pr. Xavier Gloerfelt, for their unwavering support, invaluable guidance, and continuous encouragement throughout the research process. The trajectory of this work has been shaped not only by their expertise and dedication, but also by their unending kindness and compassion. I am also grateful to the members of my doctoral committee for their insightful feedback and constructive criticism, which have significantly enriched the quality of this thesis.

I would like to express my gratitude to the research collaborators or labmates in the DynFluid laboratory at Arts & Métiers, who have provided a collaborative and stimulating research environment, with whom I have continuously enjoyed various discussions on various topics. I would also like to thank my Math and Physics teachers throughout my education, who nurtured my curiosity and love for science.

My heartfelt gratitude goes to my family for their unwavering support and understanding throughout this challenging journey. Their encouragement has been a source of strength and I am truly fortunate to have them by my side. My deepest appreciation goes to my mother, whose boundless love, encouragement, and sacrifices have been a constant source of inspiration. Her resilience and belief in my abilities have fueled my determination to pursue and complete this Ph.D. degree. I am also deeply grateful to my wife for her continued support and understand-

ing. A special thought also goes to my father, up there in the sky, who has always believed in me.

Lastly, I would like to acknowledge the countless individuals who, directly or indirectly, contributed to this journey. Your support, encouragement, and shared wisdom have been invaluable. This thesis would not have been possible without the collective efforts of all those mentioned above. Thank you for being an integral part of this academic milestone.

Abstract

This PhD thesis aims to enhance the current RANS turbulence models using Machine Learning (ML), and is organized in three main parts. First, we employ the Sparse Bayesian Learning (SBL) algorithm to derive sparse and stochastic closures of EARSM-type for the baseline $k-\omega$ SST model to address turbulent separated flows. The resulting models, denoted SBL-SpaRTA models, are interpretable, frame-invariant, and enable improved velocity and friction coefficient predictions compared to the baseline, while providing confidence intervals around the predictions. While effective on their training flow category, these models show weak generalizability. This motivates the second part of the thesis where we use the precedent framework to derive customized SBL-SpaRTA for a set of typical flow cases comprising flat plates, separated flows and jets. Then, we train a ML regressor to automatically attribute local weights to the predictions of every model, reflecting its likelihood and knowing the local underlying physics. While this "non-intrusive" approach exhibits good generalizability and substantial enhancements over the baseline model for both training and unseen test cases, its final prediction does not necessarily adhere to the conservation equations. Finally, in the third part, we address this issue by applying an intrusive methodology for model aggregating, where the customized SBL-SpaRTA are automatically blended in the CFD code using ML. This framework is compared to the non-intrusive paradigm using a systematic methodology, thus enabling to evaluate their merits and drawbacks.

Keywords: Turbulence modeling, Machine Learning, Sparse Bayesian Learning, Explicit Algebraic Reynolds Stress Models, separated flows, jet flows, boundary layers, sensitivity analysis, Mixture-of-Experts, Model Aggregation.

Résumé

L'objectif de cette thèse est d'améliorer les modèles de turbulence RANS existants au moyen de l'apprentissage automatique (ML) et se structure en trois volets principaux. D'abord, nous avons recours à l'algorithme de l'apprentissage bayésien parcimonieux (SBL) pour trouver des fermetures parcimonieuses et stochastiques de type EARSM pour le modèle $k-\omega$ SST de base, dans le but de traiter ses déficiences à bien prédire les écoulements turbulents séparés. Ces modèles, appelés SBL-SpaRTA, se caractérisent par leur interprétabilité, invariance galiléenne, et capacité à améliorer les prédictions par rapport au modèle de base, tout en fournissant des intervalles de confiance autour des prédictions. Toutefois, leur capacité de généralisation à d'autres écoulements s'avère limitée. Cette limitation motive la seconde partie de la thèse, où nous exploitons le cadre précédemment élaboré pour trouver des modèles SBL-SpaRTA personnalisés pour un ensemble de cas d'écoulement typiques, englobant des plaques planes, des écoulements séparés et des jets. Ensuite, nous procédons à l'entraînement d'un régresseur ML permettant d'attribuer automatiquement des poids locaux aux prédictions de chaque modèle, reflétant leur vraisemblance et s'appuyant sur la physique locale de l'écoulement. Alors que cette approche 'non intrusive' se distingue par sa bonne capacité de généralisation et améliore significativement le modèle de base, ses prédictions n'adhèrent pas nécessairement aux équations de conservation. Enfin, dans la troisième partie, nous remédions à cette lacune en appliquant une méthodologie intrusive pour l'agrégation des modèles, au sein de laquelle les modèles SBL-SpaRTA personnalisés sont intégrés et mélangés automatiquement dans le code CFD. Les résultats sont comparés au paradigme non intrusif, permettant ainsi d'en évaluer les points forts et limites.

Mots clés : Modélisation de turbulence, Apprentissage automatique, Apprentissage bayésien

parcimonieux, Modèles EARSM, Écoulements séparés, Écoulements de jet, Couches limites, Analyse de sensibilité, Mélange de modèles d'experts, Agrégation de modèles.

Contents

List of Tables

List of Figures

List of Acronyms

CFD Computational Fluid Dynamics

RANS Reynolds-Averaged Navier Stokes

LES Large Eddy Simulation

DNS Direct Numerical Simulation

BSL Baseline

LEVM Linear Eddy Viscosity Model

EARSM Explicit Algebraic Reynolds Stress Model

TKE Turbulent Kinetic Energy

SST Shear Stress Transport

SBL Sparse Bayesian Learning

SVM Support Vector Machine

RVM Relevance Vector Machine

QoI Quantit(y)(ies) of Interest

MAP Maximum–A–Posteriori

ZPG Zero Pressure Gradient

APG Adverse pressure gradient turbulent boundary layers

TBL Turbulent Boundary Layer

CHAN Turbulent channel flows

ASJ Axisymmetric Subsonic Jet

ANSJ Axisymmetric Near-Sonic Jet

SEP Separated flows

CBFS Curved Backward Facing Step

CD Converging-Diverging channel

PH Periodic Hills

WMH Wall-Mounted Hump

SpaRTA Sparse regression of Reynolds Tensor Anisotropy

X-MA Space-dependent Model Aggregation

UQ Uncertainty Quantification

HDI High Density Intervals

BMA Bayesian Model Averaging

BMSA Bayesian Model-Scenario Averaging

ML Machine Learning

RFR Random Forest Regressors

GPR Gaussian Process Regressor

GBR Gradient Boost Regressor

DTR Decision Tree Regressor

Chapter 1

Introduction

1.1 Bibliography

Despite the significant growth in computing power, high-fidelity turbulent flow simulations such as Direct Numerical Simulation (DNS) or Large Eddy Simulation (LES) remain prohibitively expensive for daily use in industrial applications. As a result, engineering design and optimization tasks often rely on Reynolds-Averaged Navier Stokes (RANS) equations supplemented by suitable "turbulence closure models". The averaging process in RANS equations leads to concentrate the fluctuating turbulent information into a tensorial term called Reynolds stress tensor, which will be subject to modeling by the mean of turbulence models. The latter thus play a crucial role in Computational Fluid Dynamics (CFD) solvers as they represent the influence of turbulent scales on the mean flow, and the fidelity of the CFD simulation depends on how accurate the turbulent phenomena are modeled. Many different models have been developed over the years, with varying degrees of success in predicting extended ranges of flows. The reader is referred, *e.g.*, to [1], [2], [3] and to the books of [4, 5] for an overview of RANS turbulence models. Among the widely used turbulence models, Linear Eddy Viscosity Model (LEVM) are prominent. They are based on the Boussinesq assumption, which assumes an alignment between the Reynolds stress and the mean strain rate tensors. Amid the major contributions to RANS turbulence modeling using LEVM, the $k - \epsilon$ model stands out as one of the first and most basic two-equations models. Introduced by Lumley[6] in 1967, it derives

two transport equations for the Turbulent Kinetic Energy (TKE) k and its dissipation rate ϵ. Building on the $k-\epsilon$ model, the $k-\omega$ model emerged as an extension and upgrade. The initial two-equation $k-\omega$ turbulence model introduced was Kolmogorov's $k-\omega$ model, as outlined in Kolmogorov's work in 1942 [7], with ω being the turbulence frequency. Over the years, several enhanced versions of Kolmogorov's model have emerged, proposed by various researchers, including Saiy[8](1974), Spalding[9] in 1979, Wilcox [10] in 1988, Speziale[11] in 1990, and Menter [12] in 1992. In these models, the turbulent timescale information is accounted for by means of the transport equation of the specific rate of dissipation ω instead of the dissipation rate ϵ. The $k-\omega$ Shear Stress Transport (SST) model represents another significant advancement in RANS turbulence modeling, specifically designed to provide improved predictions of turbulent shear flows. Introduced by Menter[12] in 1992, this model uses a blending of the $k-\epsilon$ and $k-\omega$ and has since become widely used in various industrial and aerospace applications. For improved predictions of Turbulent Boundary Layer (TBL), particularly near walls, the Spalart-Allmaras model proved to be highly useful. Developed by Spalart and Allmaras [13] in 1992, this one-equation turbulence model has gained popularity and finds wide application in industrial and aerospace scenarios. However, as it has been highlighted before, these models are based on the Boussinesq assumption that remains genuinely not verified even for relatively simple flows (see [5, 14]).

A large part of the turbulence modeling literature during the last three decades reports attempts to upgrade the baseline LEVM by adding nonlinear terms suited to sensitize the model to curvature effects or to improve its anisotropy. Examples are given by the SARC (Spalart–Allmaras with Rotation and Curvature, [15]), non-linear models [16], elliptic relaxation models [17], algebraic Reynolds Stress models [18] or explicit algebraic Reynolds Stress models ([19, 20, 21]), and full Reynolds Stress Models [22]. The latter require the solution of additional transport equations for the Reynolds stress components plus an equation for a quantity allowing to determine a turbulent scale. Of particular interest for the present work are so-called Explicit Algebraic Reynolds Stress Model (EARSM), originally derived by Pope [19] from the transport equations of Reynolds Stress Models (RSM) under local equilibrium

assumptions. Padé approximations were used to obtain explicit expressions, in contrast with the so-called Algebraic Stress Models (ASM) [18] that lead to implicit algebraic expressions for the Reynolds stress components. The above-mentioned EARSM models rely on purely physical arguments, along with simplifying assumptions that may limit their performance for flows significantly different from those for which they were calibrated. Since Pope's contribution, various improvements have been proposed in the literature (*e.g.*, [20, 21, 23]). Unfortunately, the balance accuracy / robustness / computational cost of such more complex models has prevented a widespread use in CFD applications. In addition, more complex models typically involve a larger number of adjustable closure coefficients. In addition, despite continuous research efforts through several decades, "no class of models has emerged as clearly superior, or clearly hopeless" [1] until now. In fact, whatever the closure assumptions, all RANS models suffer from uncertainties associated with *i)* the applicability of a RANS-type description of turbulence for a given flow; *ii)* the choice of a suitable mathematical structure for constitutive relations and auxiliary equations used to link turbulent quantities to the mean field, referred-to as structural or model-form uncertainty; Clearly, RANS models suffer from several shortcomings for complex flow configurations involving turbulence nonequilibrium, strong gradients, separations, shocks, 3D effects, etc that is related to the structure of Reynolds stress representation. *iii)* the calibration of the model closure parameters, known as parametric uncertainty. In fact, the turbulent quantities are computed via auxiliary relations (often transport equations) introducing a number of supplementary modeling hypotheses and closure coefficients, and the latter are calibrated against experimental or numerical data for so-called "canonical flows", *i.e.* simple turbulent flows representative of some elementary turbulent dynamics. However, 1) it is not always possible to determine closure coefficients that are simultaneously optimal for all canonical flows; 2) the calibration data are affected by observational uncertainties; and 3) the final values retained in some models are not the best fit to the data, but a compromise with respect to other requirements, *e.g.* numerical robustness. A discussion of uncertainties associated with turbulence models can be found in [24].

An alternative line of research has consisted in the development of so-called scale-resolving

approaches (Large Eddy Simulation, LES, wall-modeled LES, and hybrid RANS/LES), which directly resolve a more or less extended range of turbulent structures while modeling scales smaller than a certain filter length. Such approaches have been successful for increasingly realistic flow configurations. However, their computational cost remains significant and still prohibitive for applications involving swarm simulations, such as massive parametric studies, optimization or uncertainty quantification.

In recent years, there has been a growing interest in applying Machine Learning (ML) techniques for turbulence modeling. Machine Learning can be used to analyze large amounts of data and discover non-trivial patterns. Since all RANS models involve some degree of empiricism [1, 2], including those that were initially derived from exact manipulations of the Navier–Stokes equations, the use of Machine Learning can then be seen as a natural way to systematise the development of RANS models and discover formulations suitable for improving their performance for more complex flows (see the reviews of [25, 24, 26, 27]). Early studies mostly dealt with the quantification of uncertainties associated with turbulence models by using interval analysis or statistical inference tools. The analysis was conducted either by perturbing directly the Reynolds stress anisotropy tensor computed with a baseline LEVM [28, 29, 30] or by treating model closure coefficients as random variables with associated probability distributions [31],[32],[33],[34]. The first approach takes into account structural uncertainties in the constitutive relation of the Reynolds stresses, while the second one only accounts for parametric uncertainties. On the other hand, the tensor perturbation approach is intrusive, as it implies modifications of the Reynolds stress representation in the RANS solver, while the parametric approach only requires a modification of the turbulence closure coefficients introduced in the CFD solver.

Early attempts to quantify model-form uncertainty in a probabilistic framework can be found in [35], where the Demster-Shafer evidence theory is adopted, and multiple models are used for predicting a given flow configuration. More recently, [36] explored a Bayesian framework named Bayesian Model-Scenario Averaging (BMSA) to calibrate and combine in an optimal way the predictions obtained from a set of competing baseline LEVM models calibrated

on various data sets (scenarios). BMSA has been successfully applied to provide stochastic predictions for a variety of flows, including 3D wings [37] and compressor cascades [38, 39]. Since Bayesian model averaging builds a convex linear combination of the underlying models, its prediction accuracy cannot be better than the best model in the considered set, even if it outperforms the worst one.

With the aim of reducing modeling inadequacies, data-driven methods for turbulence modeling have been introduced in recent years, mostly relying on supervised Machine Learning. Examples of early contributions can be found in [40, 41], who proposed field inversion to learn corrective terms for the turbulent transport equations, along with ML to express the correction as a black-box function of selected flow features and to extrapolate it to new flows. Other contributions to the field-inversion and ML approach can be found, *e.g.*, in [42, 43, 44]. One of the advantages of field inversion is that the "goal-oriented" correction can be inferred from sparse data or even global performance parameters. On the other hand, the ML step uses flow features estimated with a baseline RANS model to infer the corresponding correction. This may cause a feature mismatch for severe flow cases, since the baseline RANS flow field may completely miss flow features expected in the high-fidelity solution. In order to improve the consistency of the data-driven correction, an iterative procedure has been proposed in [45]. Additionally, in most of the above mentioned works, the features are hand-picked for a class of flows at stake, making the data-driven correction unsuitable to radically different configurations. In fact, the efficacy of the learned correction is strongly dependent on the features used to map it to unseen flow cases [46]. Even when an appropriate set of feaures is available, the learned corrections tend to lack generality, and cannot be applied to flows significantly different from those used to learn the correction [47]. Using a similar framework, Xiao *et al.* [48, 49] (see also [50]) performed a truncated Karhunen-Loeve expansion to get a lower-dimensional representation of Reynolds-stress anisotropy across the computational domain, and then applied Bayesian inference to infer posterior distributions of the augmented model coefficients. Both approaches provided improved solutions with uncertainty interval estimates for the training cases, but their applicability outside the training set remained limited. On the other hand, Edeling *et al.* [51]

proposed a "return-to-eddy-viscosity" model, which relies on transport equations with a source term describing the Reynolds-stress anisotropy discrepancy. The model coefficients in the PDEs can be calibrated by using data and Bayesian inference, and the calibrated equations can be further used for predictions. As the preceding ones, this approach involves an expensive Bayesian inference step, although the cost can be relieved using surrogate models.

The seminal work of [52] introduced a novel neural network architecture (Tensor Basis Neural Network, TBNN) that allows frame invariance constraints to be incorporated into the learned explicit Reynolds stress anisotropy correction. The idea is to project the correction term onto a minimal integrity basis, as in the extended eddy viscosity model of [19], leading to a form of generalized Explicit Algebraic Reynolds Stress model, whose function coefficients are regressed from high-fidelity data using ML. [48, 49, 53, 54, 50] combined ML techniques for identifying regions of high RANS modeling uncertainty along with other ML algorithms for inferring model corrections from data, and for predicting new configurations. The procedure, initially relying on the assimilation of full high-fidelity fields, has been subsequently extended to the assimilation of sparse data by using end-to-end differentiation [55]. In [56], the use of vector cloud Machine Learning upholds the desired invariance properties of constitutive models, accurately reflects the physical region of influence, and can be applied to various spatial resolutions; however, it still needs to integrate the information about the turbulent length and velocity scales via transport equations to provide a better description of the Reynolds stresses. Other recent contributions are given by [57] who used tensor-basis Random Forests to learn data-driven corrections of the Reynolds stress tensor, and [58] who proposed a general principled framework for deriving deep learning turbulence model corrections using deep neural network (DNN) while embedding physical constraints and symmetries. The aforementioned approaches to modeling turbulence from data use so-called black-box ML, such as neural networks or Random Forests. They allow a flexible approximation of complex functional relationships, but do not provide an explicit, physically interpretable mathematical expression for the learned correction. Recent attempts to interpret ML-augmented turbulence models rely on nonlinear sensitivity analysis tools, *e.g.* Shapley factor analyses [59]. Another downside of black-box methods is their difficult

implementation in a CFD solver to make robust predictions of new flows.

An interesting alternative is represented by so-called open-box ML approaches, which consist in selecting explicit mathematical expressions and/or operators from a pre-defined dictionary to build a suitable regressor for the data. Examples of open-box ML include Genetic Programming (GEP) [60] and symbolic identification [61, 62, 63]. In the latter, the Reynolds-stress anisotropy is projected onto Pope's [19] minimal integrity basis and ML is used to regress the function coefficients of the decomposition. Sparsity-promoting formulations of the cost function are used in [61, 62] to minimize the number of active terms and limit the occurrence of overfitting. The resulting models correspond to data-driven EARSM with fully explicit analytic expressions, but again without estimates of their predictive uncertainty. Open-box ML approaches have been applied successfully to the development of data-driven models for a variety of applications [64, 65, 66, 67, 68, 69]. Although less expressive than black-box ML, due to the quickly escalating complexity of the search procedure in large mathematical operator dictionaries, the open-box models have the merit of delivering tangible mathematical expressions, which can be easily integrated into existing CFD solvers and interpreted in light of physical considerations. Both a priori and "CFD-in-the-loop" training procedures have been proposed [70, 71], the latter allowing the use of incomplete data, at the cost of solving a large optimization problem.

Regardless of their formulation and training procedure, both black-box and open-box suffer from some common drawbacks. First, the learned corrections are generally non-local, meaning that they may alter the predictions of RANS models also where the baseline LEVM already gives good results. Second, such models tend to behave well only on narrow classes of flows and operating conditions, which implies that they must be often retrained as soon as a new flow configuration must be addressed. This has recently fostered attempts to merge or combine models trained for different settings to make accurate predictions of wider ranges of flows. Additionally, the necessity of simultaneously improving turbulent flow predictions and efficiently estimating uncertainty intervals for the predictions, especially when the model is applied to configurations significantly different from the training ones, remains of the utmost importance for providing reliable flow predictions and is far from being achieved. The identification of flow

regions of greater sensitivity to turbulence modeling errors also represents valuable information for designers.

In this optic, we find several attempts in the literature for building composite physics-based turbulence models, with different corrective terms intended to capture specific phenomena (*e.g.*, transition, rotation), being added to a baseline RANS model. An example of such a procedure is given by the various corrections of the SA model, both physics-based (several variants of the Spalart–Allmaras model are reported on the NASA Turbulence modeling Resource[†]) and, more recently, data-driven [72]. Another example is given by the so-called GEKO (GEneralized k-ω) model, introduced by [73]. The idea behind GEKO is to add several localized, tunable corrections that the model users can adapt to their own problem. This means that although the model structure is intended to be very flexible, the associated coefficients are case-dependent and data-driven. Another drawback is that GEKO is implemented in a commercial code: the model details are not publicly accessible. More recently, [74] combined field inversion and Random Forests classifiers to train a correction field for the k-ω SST model on bump flow cases involving various kinds of turbulent dynamics (boundary layer, separation), then to classify subregions in a space of features, used to extract optimal local corrections for new cases. This approach showed promising results, indicating the interest of zonal models for improving the generalizability of ML-augmented turbulence models. [75] used field inversion along with Gaussian process emulators (GPE) to build stochastic regional models. The latter are trained on various flow data sets, then the individual GPE are blended together as convex linear combination, with weights given by the model inverse variances, which depend in turn on the local flow features. The approach is successfully applied to separated flow cases. The stochastic nature of the model helps in assimilating various data sources, but it is not exploited for quantifying the uncertainty associated with model predictions. [76] have recently proposed a so-called building-block approach for data-driven LES wall modeling. The model is formulated to account for various flow configurations, such as wall-attached turbulence, wall turbulence under favorable / adverse pressure gradients, separated turbulence, statistically unsteady tur-

[†]`https://turbmodels.larc.nasa.gov`

bulence, and laminar flow. The model relies on a classifier that recognizes local similarities of the predicted flow with a collection of known building-block flows; subsequently, a predictor based on neural networks leverages the information of the classifier together with the input to generate the wall shear stress prediction via combination of the building-block flows from the database, and a confidence score is assigned to the prediction. The preceding approaches make use of "internal" combinations of competing models for the Reynolds stresses or for the wall laws. This means that the composite model is trained with high-fidelity full-field data, then implemented within a CFD solver and used to predict a new case. The internal combination can lead to numerical difficulties if the transition between component models is not smooth enough. Additionally, except in the case of [76], no confidence estimate is associated with the predictions, which can be problematic when predicting flow cases that differ significantly from the training ones. An alternative approach consists in using multi-model ensemble predictions in an uncertainty quantification (UQ) setting. One of the first attempts can be found in [36], where the BMSA methodology was used to combine the solutions of a set of competing LEVM models calibrated on various data sets (scenarios). In BMSA, each component model is used to make a prediction of a new flow, and an aggregated estimate of the solution is obtained as a linear combination of the competing model predictions weighted by the posterior model probabilities. The solution variance can also be evaluated as the result of the uncertain model parameters, model structure, and choice of the calibration data, thus delivering an estimate of the predictive confidence intervals. BMSA has been successfully applied to provide stochastic predictions for a variety of flows, including 3D wings [37] and compressor cascades [77, 78]. BMSA, and Bayesian Model Averaging (BMA) [79, 80] from which it originates, may be interpreted as stochastic variants of a multi-model framework called Model Aggregation [81, 82, 83]. Such methods combine multiple predictions stemming from various models –also termed experts or forecasters– to provide a global, enhanced solution. The above-mentioned methods, however, assign to each model the same weight throughout the domain. Since, in practice, models perform better or worse depending on the local flow physics, a better strategy consists in assigning higher weights to the best performing models in each region. Other classes of en-

semble methods allow space-varying weights. Specifically, so-called Mixture-of-Experts models [84] or Mixture Models, softly split the input feature space (covariate space) into partitions where the locally best-performing models are assigned higher weights. The soft partitioning is accomplished through parametric gate functions, or a network of hierarchical gate functions [85], that rank the model outputs with probabilities. In this spirit, [86] (to which we refer for a more complete literature review on ensemble models) recently proposed a method for spatially combining the predictions of a set of well-known LEVM taken from the literature, called space-dependent Model Aggregation (X-MA). For that purpose, a cost function is introduced to evaluate the local model performance with respect to some training data, which is used to build the model weights. To make predictions, the weights are regressed in a space of flow features (representative of various flow phenomena) by using Random Forests. Similarly to BMSA, an estimate of the predictive uncertainty can be inferred by measuring the level of agreement of the component model predictions. In [86] the X-MA methodology was successfully applied to predict flows through a compressor cascade. Although the X-MA results do improve over the baseline component models overall, the local X-MA predictions cannot be more accurate than the best LEVM component model by construction.

1.2 Objective and Plan of the thesis

In this thesis, our primary objective is to address and rectify the structural deficiencies present in the baseline $k-\omega$ SST model. Specifically, after introducing the numerical tools employed in this thesis in Chapter 2, along with an overview of the selected flow cases used for both training and results evaluation in Chapter 3, our focus will center in Chapter 4 on deriving data-driven, sparse and stochastic corrections of the baseline $k-\omega$ SST using the general Explicit Algebraic Reynolds Stress Models (EARSM) formulation. These corrections are designed to possess specific attributes: they must be realizable, physically interpretable, Galilean frame-invariant, and resilient to overfitting. To achieve this objective, we employ a Bayesian formulation of the learning problem utilizing the Sparse Bayesian Learning (SBL) algorithm. This approach transforms the corrective coefficients of our models from deterministic

values into probability distributions. This probabilistic nature offers the distinct advantage of being able to quantify structural uncertainties surrounding various quantities of interest. Additionally, it allows us to assess the sensitivity of these quantities with respect to each corrective coefficient's distribution. Our framework is initially applied to derive specific corrections to turbulent separated flow cases — a common scenario where the baseline $k-\omega$ SST model tends to overpredict the recirculation region. This preliminary study will serve as a proof of concept. Subsequently in Chapter 5, we apply the same framework to derive customized corrections for a range of typical flow cases, encompassing canonical flat plates, free shear layers, jet flows, near-equilibrium wall-bounded turbulence and separated flow cases. Our ultimate goal is then to develop a method for automatically aggregating the predictions of these derived corrections following the local underlying physics in the flow by the mean of local model weights. This aggregation occurs post individual convergence of every candidate model, utilizing weights that have been trained to identify regions where each model is most likely to outperform the others. These weights are also linked to local physics through the utilization of local flow features. However, this aggregation method that we termed "non-intrusive X-MA", can be viewed as a post-processing technique, results in a final aggregated solution that does not inherently satisfy the conservation equations. In order to tackle this challenge, we introduce in Chapter 6 an intrusive method for aggregating a set of previously derived customized models. This method involves the creation of a composite EARSM-type model that is applied locally during the simulation, utilizing in every cell of the computational domain the correction(s) with the highest likelihood. We refer to this approach as "intrusive X-MA" and its performance is compared to that of the non-intrusive method.

Chapter 2

Governing equations and numerical tools

Contents

This section presents RANS equations and the numerical tools used in the present work. We recall in Section 2.1 mainly the governing equations for turbulent compressible flows and the RANS models used in the study. Then, in Section 2.2 we give a brief introduction to *OpenFOAM* software employed to implement the different models and methodologies.

2.1 Reynolds-Averaged Navier Stokes (RANS) equations

In the field of fluid dynamics, the application of Reynolds-Averaged Navier-Stokes (RANS) equations stands as a widely accepted methodology for analyzing turbulent flows. The core principle of RANS involves the temporal averaging of the fundamental Navier-Stokes equations. This process entails decomposing the flow field into its mean and fluctuating component, leading to the derivation of equations governing the behavior of mean flow quantities. While this temporal averaging results in the loss of instantaneous fluctuating information, it simplifies the

computational resolution of turbulent flows. The averaging process in the RANS framework leads to concentrate the complex turbulent dynamics within a tensorial entity referred to as the Reynolds stress tensor, which necessitates modeling for accurate representation. This modeling undertaking involves both identifying appropriate closure models, deriving transport equations for the underlying physical quantities and calibrating the associated closure coefficients to enhance predictive accuracy.

2.1.1 RANS formulation

We focus on turbulent closures for the steady incompressible RANS equations:

$$\begin{cases} \dfrac{\partial U_i}{\partial x_i} = 0 \\ U_j \dfrac{\partial U_i}{\partial x_j} = -\dfrac{1}{\rho}\dfrac{\partial P}{\partial x_i} + \dfrac{\partial}{\partial x_j}\left(\nu \dfrac{\partial U_j}{\partial x_J} - \tau_{ij}\right) \end{cases} \tag{2.1}$$

with U_i the i-th mean velocity component, P the mean pressure, ρ the fluid density and ν the kinematic viscosity. Equations (2.1) are supplemented with a LEVM model for the Reynolds stress tensor $\tau_{ij} = \langle u_i' u_j' \rangle$, with u_i' the i-th fluctuating velocity component and $< . >$ the statistical mean value.

Splitting the Reynolds stress tensor into an isotropic and an anisotropic part

$$\tau_{ij} = \underbrace{\frac{2k}{3}\delta_{ij}}_{\text{isotropic}} + \underbrace{a_{ij}}_{\text{anisotropic}}, \text{where} \quad k = \frac{\tau_{ii}}{2} \quad \text{and} \quad \delta_{ij} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}, \tag{2.2}$$

k being the turbulent kinetic energy and δ_{ij} the Kronecker symbol. A LEVM is obtained by assuming that the Reynolds stress anisotropy a_{ij} is a linear function of the mean strain rate tensor $\mathbf{S}$ (Boussinesq hypothesis):

$$a_{ij} = -2\nu_t S_{ij} = 2k b_{ij}^0, \quad \text{where} \quad S_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right), \tag{2.3}$$

b_{ij}^0 being is the normalized anisotropy tensor of the Boussinesq model. Most often, the eddy viscosity coefficient ν_t is computed via well-chosen turbulent properties obtained by solving

auxiliary transport equations, such as in the $k-\omega$ SST model [12] used in the following of this study.

The models considered in the present study are further described in the following subsections.

2.1.2 Menter's $k-\omega$ SST model

Menter's Shear Stress Transport turbulence model [12], or SST, is a widespread and robust two-equation turbulence model used in CFD. The model combines the $k-\omega$ of Wilcox [10] and $k-\epsilon$ [87] turbulence models such that the $k-\omega$ is used in the inner region of the boundary layer and switches to the $k-\epsilon$ in the free shear flow. The SST two equation turbulence model was introduced by Menter in 1992 to deal with the strong freestream sensitivity of the $k-\omega$ turbulence model and improve the predictions under adverse pressure gradients. The formulation of the SST model is based on physical considerations and attempts to predict solutions to typical engineering problems. Over the last two decades the model has been altered to more accurately reflect certain flow conditions. The two variables calculated are interpreted so k is the turbulence kinetic energy and ω is the specific rate of dissipation of the eddies.

$$\begin{cases} \dfrac{\partial k}{\partial t} + U_j \dfrac{\partial k}{\partial x_j} = P_k - \beta^* k\omega + \dfrac{\partial}{\partial x_j}\left[(\nu + \sigma_k \nu_t)\dfrac{\partial k}{\partial x_j}\right] \\ \dfrac{\partial \omega}{\partial t} + U_j \dfrac{\partial \omega}{\partial x_j} = \dfrac{\gamma}{\nu_t} P_k - \beta\omega^2 + \dfrac{\partial}{\partial x_j}\left[(\nu + \sigma_\omega \nu_t)\dfrac{\partial \omega}{\partial x_j}\right] + 2(1-F_1)\dfrac{\sigma_{\omega 2}}{\omega}\dfrac{\partial k}{\partial x_j}\dfrac{\partial \omega}{\partial x_j} \end{cases} \tag{2.4}$$

The production of turbulent kinetic energy is computed as follow:

$$\begin{cases} P_k & = \min\left(-2\nu_t S_{ij}\dfrac{\partial U_i}{\partial x_j}, 10\beta^*\omega k\right) \\ \nu_t & = \dfrac{a_1 k}{\max\left(a_1\omega, SF_2\right)} \\ F_1 & = \tanh\left(arg_1^4\right) \\ arg_1 & = \min\left[\max\left(\dfrac{\sqrt{k}}{\beta^*\omega y}, \dfrac{500\nu}{y^2\omega}\right), \dfrac{4\sigma_{\omega 2}k}{CD_{k\omega}y^2}\right] \\ CD_{k\omega} & = \max\left(2\sigma_{\omega 2}\dfrac{1}{\omega}\dfrac{\partial k}{\partial x_j}\dfrac{\partial \omega}{\partial x_j}, 10^{-20}\right) \\ F_2 & = \tanh\left(arg_2^2\right) \\ arg_2 & = \max\left(2\dfrac{\sqrt{k}}{\beta^*\omega y}, \dfrac{500\nu}{y^2\omega}\right) \end{cases} \tag{2.5}$$

The constants β, σ_k, σ_ω are computed by a blend from the corresponding constants via the following formula:

$$\begin{cases} \Phi & = F_1\Phi_1 + (1-F_1)\Phi_2, \\ \beta & = \left(\dfrac{3}{40}, 0.0828\right), \\ \sigma_k & = (0.85, 1.0), \\ \sigma_\omega & = (0.5, 0.856) \end{cases} \tag{2.6}$$

where Φ_1 and Φ_2 are respectively the values of the constant Φ (here β, σ_k and σ_ω) in $k-\epsilon$ and $k-\omega$. The remaining terms are $\beta^* = 0.09$, $a_1 = 0.31$ and $S = \sqrt{2S_{ij}S_{ij}}$. The $k-\omega$ SST model will serve as the "baseline" turbulence model for this thesis. Multiple corrections will be derived specifically for this turbulence model.

2.1.3 Baseline (BSL) EARSM model of Menter

We here recall the formulation of the BSL-EARSM of Menter *et al.* [23]. This model is based on the EARSM formulation of Wallin and Johansson [21] (WJ model) for the stress-strain relationship. In the WJ model, the stress-strain relationship is combined with the k-ω transport equations of Wilcox [5]. In the BSL-EARSM, in order to avoid the freestream sensitivity of the Wilcox model, the WJ stress-strain relationship is combined with the BSL k-ω model of Menter [88].

Following [19], the Reynolds stress anisotropy tensor a_{ij} is projected onto a tensor basis:

$$a_{ij} = \beta_1 T_{1,ij} + \beta_2 T_{2,ij} + \beta_3 T_{3,ij} + \beta_4 T_{4,ij} + \beta_6 T_{6,ij} + \beta_9 T_{9,ij} \tag{2.7}$$

where

$$\begin{cases} T_{1,ij} = S^*_{ij}; \quad T_{2,ij} = S^*_{ik}S^*_{kj} - \dfrac{1}{3}I_1\delta_{ij}; \quad T_{3,ij} = \Omega^*_{ik}\Omega^*_{kj} - \dfrac{1}{3}I_2\delta_{ij}; \\ T_{4,ij} = S^*_{ik}\Omega^*_{kj} - \Omega^*_{ik}S_{kj}; \quad T_{6,ij} = S^*_{ik}\Omega^*_{kl}\Omega^*_{lj} + \Omega^*_{ik}\Omega^*_{kl}S_{lj} - \dfrac{2}{3}I_4\delta_{ij} - I_2 S^*_{ij}; \\ T_{9,ij} = \Omega^*_{ik}S^*_{kl}\Omega^*_{lm}\Omega^*_{mj} - \Omega^*_{ik}\Omega^*_{kl}S^*_{lm}\Omega^*_{mj} + \dfrac{1}{2}I_2(S^*_{ik}\Omega^*_{kj} - \Omega^*_{ik}S^*_{kj}), \end{cases} \tag{2.8}$$

with S^*_{ij} and Ω^*_{ij}, the non-dimensional mean strain rate and rotation rate defined as follows :

$$S^*_{ij} = \frac{\tau}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right), \quad \Omega^*_{ij} = \frac{\tau}{2}\left(\frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i}\right) \tag{2.9}$$

where τ is a turbulent time scale with a Kolmogorov limiter [17]:

$$\tau = \max\left(\frac{1}{C_\mu \omega}, 6\sqrt{\frac{\nu}{C_\mu k \omega}}\right), \quad C_\mu = 0.09. \tag{2.10}$$

The tensor invariants I_1, I_2 and I_4 read:

$$I_1 = S^*_{ij}S^*_{ji}, \quad I_2 = \Omega^*_{ij}\Omega^*_{ji}, \quad I_4 = S^*_{ik}\Omega^*_{kj}\Omega^*_{ji}. \tag{2.11}$$

The coefficients of the tensor basis β_i in 2.7 are defined as :

$$\beta_1 = -\frac{N}{Q}, \quad \beta_2 = 0, \quad \beta_3 = -\frac{2I_4}{NQ_1}, \quad \beta_4 = -\frac{1}{Q}, \quad \beta_6 = -\frac{N}{Q_1}, \quad \beta_9 = \frac{1}{Q_1}, \tag{2.12}$$

with

$$Q = \frac{(N^2 - 2I_2)}{A_1}, \quad Q_1 = \frac{Q}{6}(2N^2 - I_2) \tag{2.13}$$

where

$$N = C_1' + \frac{9}{4}\frac{\tilde{P}_k}{\epsilon} \tag{2.14}$$

and

$$A_1 = 1.2, \quad C_1' = \frac{9}{4}(C_1 - 1) \quad \text{and} \quad C_1 = 1.8. \tag{2.15}$$

N is a solution of the cubic equation :

$$N^3 - C_1' N^2 - (2.7 I_1 + 2 I_2) N + 2 C_1' I_2 = 0 \tag{2.16}$$

which is given by :

$$\begin{cases} N = \dfrac{C_1'}{3} + \left(P_1 + \sqrt{P_2} \right)^{1/3} + \text{sign}(P_1 - \sqrt{P2}) \mid P_1 - \sqrt{P_2} \mid^{1/3} & \text{at} \quad P_2 \geq 0 \\ N = \dfrac{C_1'}{3} + 2(P_1^2 - P_2)^{1/6} \cos\left(\dfrac{1}{3} \arccos\left(\dfrac{P_1}{\sqrt{P_1^2 - P_2}} \right) \right) & \text{at} \quad P_2 < 0 \end{cases} \tag{2.17}$$

with

$$P_1 = C_1' \left(\frac{C_1'^2}{27} + \frac{9}{20} I_1 - \frac{2}{3} I_2 \right), \quad P_2 = P_1^2 - \left(\frac{C_1'^2}{9} + \frac{9}{10} I_1 + \frac{2}{3} I_2 \right)^3. \tag{2.18}$$

The BSL-EARSM constitutive equation is then supplemented by transport equations for k and ω. These are the same as in 2.1.2. The reader is referred to [23] for further details. In this chapter, we have introduced in particular the BSL-EARSM model solely for the purpose of comparison. This choice stems from the fact that all the data-driven models developed in this thesis fall under the EARSM formulation. Thus, it is pertinent to assess their performance against a model of the same type, but rather derived using physical considerations.

2.2 *OpenFOAM*

OpenFOAM (Open source Field Operation And Manipulation) is an object-oriented C++ framework that can be used to build a variety of computational solvers for problems in continuum mechanics and fluid dynamics with a focus on finite volume discretization. *OpenFOAM*

also includes several ready solvers, utilities, and applications that can be directly used. At the core of these libraries are a set of object classes that allow the programmer to manipulate meshes, geometries, and discretization techniques at a high level of coding. *OpenFOAM* uses numerous solvers, each one adapted to a typical flow category. In our study, we will be interested in *SimpleFoam* solver. *SimpleFoam* is a steady-state solver for incompressible, turbulent flow, using the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm. The convective terms in the transport equation are discretized using linear upwinding and viscous terms with 2^{nd} order central difference scheme. The solution is advanced to the steady state using a Gauss-Seidel smoother.

The implementation and calculations of the EARSM-type models as will be presented in the future chapters are performed using a modified version of *OpenFOAM* [89]. The latter has been extended by implementing the general EARSM form. The corresponding code can be downloaded from the public *github* repository[†].

[†]`https://github.com/shmlzr/general_earsm.git`

Chapter 3

Flow cases: reference data and computational setup

Contents

In this chapter, we present the multiple flow cases used in the study and that can be categorized into three distinct groups: firstly, flat plate cases featuring different pressure gradients, featuring both equilibrium and near-equilibrium boundary layers; secondly, jet flow cases, including both subsonic and near-sonic conditions with a consistent geometry; and lastly, turbulent separated flow cases characterized by varying geometries and Reynolds numbers. Some flow cases are canonical flows, on which the baseline Linear Eddy Viscosity Models have already

been calibrated on (like the zero pressure gradient turbulent boundary layer and the very far shear layer in the jet flows) and therefore will serve as validation cases, while others are more challenging. The main objective is to present a diverse range of flow scenarios, encompassing diverse geometries, physical phenomena and operational conditions, suitable for both training and testing purposes. For every case, we present the computational setup, comprising meshes and operating conditions, along with the high-fidelity data for each case.

3.1 Flat plates flow cases

3.1.1 Turbulent channel flows (CHAN)

This flow case serves as a comparative study between the predictions of various RANS models and Direct Numerical Simulation (DNS) data. The mesh configuration employed for the corresponding RANS calculations (80×120) is presented in Figure 3.1. For DNS, several distinct simulations of fully developed flow in a plane channel were used for comparison, each at different friction Reynolds numbers Re_τ (180, 395, and 590 from [90], and 1000, 2000 and 5000 from [91]), using the spectral numerical method developed by [92].

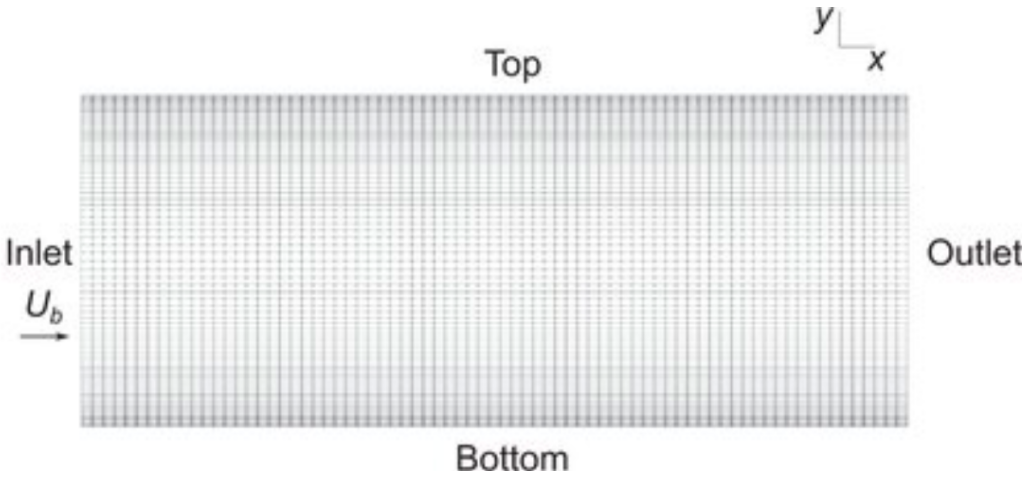
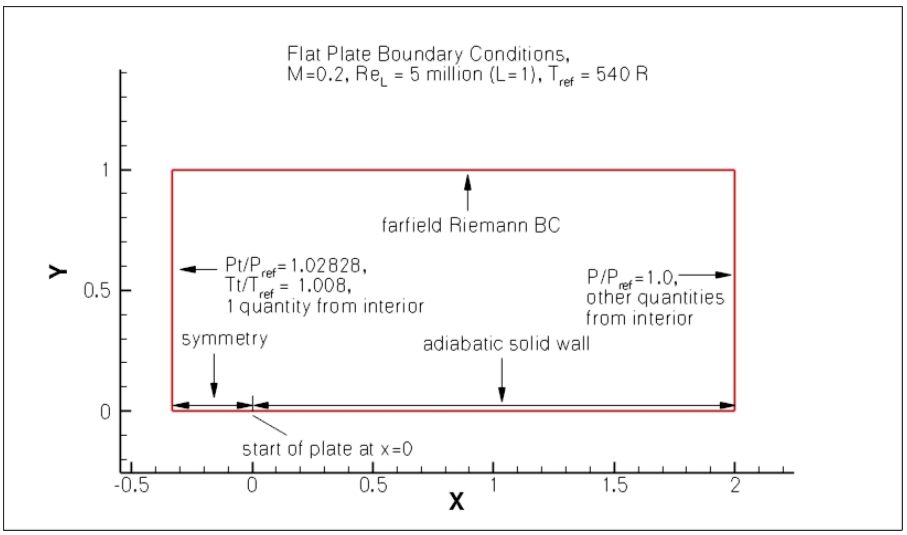


Figure 3.1: Mesh used for CHAN computations.

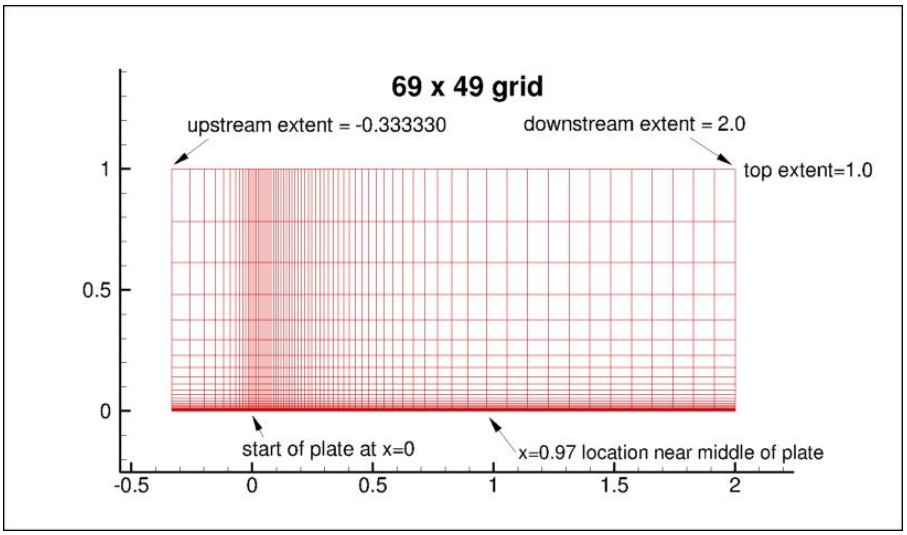
3.1.2 Zero pressure gradient turbulent boundary layers (ZPG)

The 2D Zero Pressure Gradient (ZPG) Flat Plate flow case serves as a dedicated validation setup for turbulence models, specifically designed to assess their accuracy under well-defined incompressible operating conditions. In this flow case, we employ a Mach number of 0.2, ensuring incompressible conditions.

The setup utilizes a series of nested grids to enable comprehensive analysis and comparisons with reference data. To achieve the desired Re_θ levels, the Reynolds number per unit length (Re_L) for this flow case is set to 5 million. The flat plate geometry layout is straightforward, and typical boundary conditions are applied (see Figure 3.2). An essential consideration is the maximum boundary layer thickness, which is approximately 0.03 times the plate length (L). To minimize any potential influence on the simulation results, the grid height ($y = L$) is strategically positioned at a significant distance from the boundary layer. Sensitivity tests have verified the robustness of the chosen grid setup, showing that adjustments to the upper extent (*e.g.*, $y = 0.48L$) have negligible impact on the integrated drag or skin friction at $x = 0.97$, with variations remaining below 0.2%. This 2D ZPG Flat Plate flow case provides a well-defined geometric setup and operating conditions for validating turbulence models. Its focus on practical flow characteristics enhances the credibility of CFD simulations, making it a valuable tool for assessing turbulence modeling performance in real-world engineering applications. The grid used is a 3-D grid, with two identical $x - z$ planes, separated by a distance $y = 1$, giving one spanwise cell for all grid levels. The high-fidelity data available is DNS data (velocity and other turbulent quantities like Reynolds stresses) of [93] at $Re_\theta = 670, 1000, 1410, 2000, 2540, 3030, 3270, 3630, 3970$ and 4060.



(a) Boundary conditions for ZPG computations.



(b) Mesh used for ZPG computations.

Figure 3.2: ZPG computational setup.

3.1.3 Adverse pressure gradient turbulent boundary layers (APG)

These flow cases are used to assess the predictions of RANS models in predicting near-equilibrium boundary layers. The latter are characterized through the Clauser pressure-gradient parameter $\beta = \delta^*/\tau_w \frac{dP_e}{dx}$, where δ^* is the displacement thickness, τ_w the Reynolds wall shear stress and $\frac{dP_e}{dx}$ is the streamwise pressure gradient. In order to fulfill the near–equilibrium conditions, the freestream velocity was prescribed such that it followed a power-law distribution

$U_\infty = C(x - x_0)^m$, where x is the streamwise velocity, x_0 is the power-law virtual origin, and m has to be larger than $\frac{-1}{3}$ in order to obtain near equilibrium conditions. 5 adverse pressure gradient turbulent boundary layers are assessed and compared to wall-resolved LESs of [94]. Details about the flow cases are provided in Table 3.1. Simulations are conducted on a Cartesian grid of size 58×298, with $y^+ \simeq 1$ near the wall.

Case	Range of Re_θ under study	Range of β	m	x_0
b1	$910 \leq Re_\theta \leq 3360$	$\simeq 1$	-0.14	110
b2	$940 \leq Re_\theta \leq 4000$	$\simeq 2$	-0.18	110
m13	$990 \leq Re_\theta \leq 3515$	$0.96 \leq \beta \leq 1.51$	-0.13	60
m16	$1010 \leq Re_\theta \leq 4000$	$1.95 \leq \beta \leq 2.78$	-0.16	60
m18	$990 \leq Re_\theta \leq 4320$	$3.15 \leq \beta \leq 4.47$	-0.18	60

Table 3.1: Description of APG flow cases.

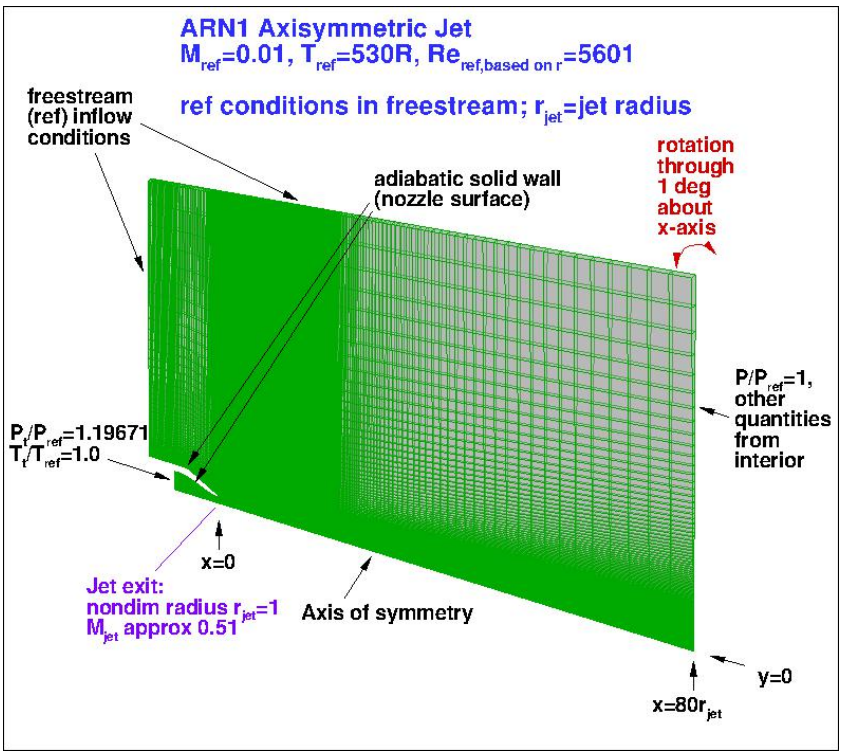
3.2 Jet flow cases

3.2.1 Axisymmetric subsonic jet (ASJ)

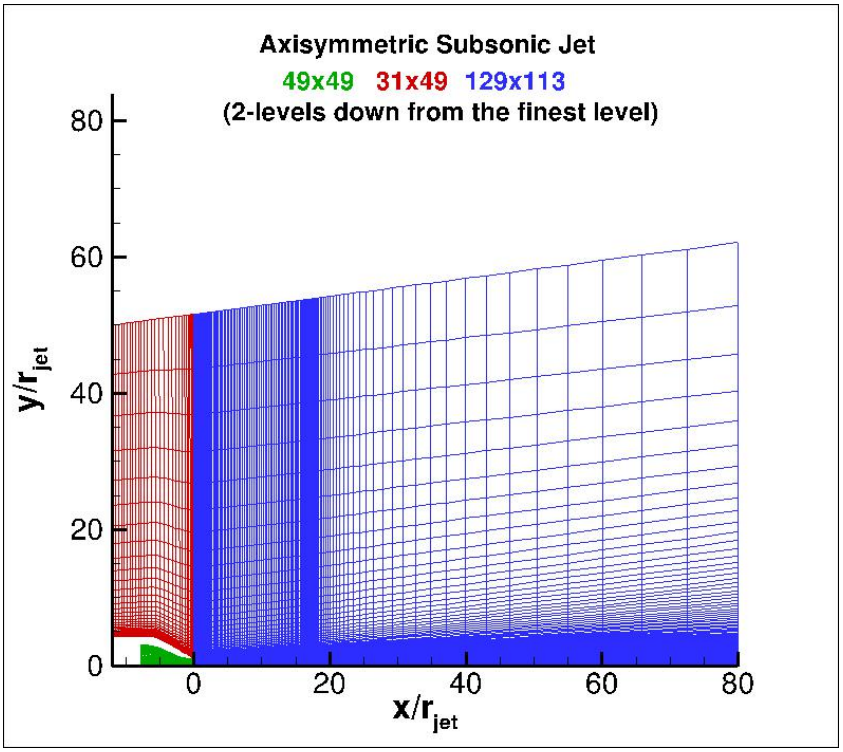
The subsonic jet validation case serves the purpose of validating turbulence models by comparing Computational Fluid Dynamics (CFD) results against experimental data, establishing the model's ability to accurately reproduce the underlying physics. To facilitate rigorous analysis, a comprehensive set of nested grids belonging to the same family is provided (see Figure 3.3). The experiment involves a jet known as Acoustic Research Nozzle 2 (ARN2), characterized by a radius of 1 inch ($25.4mm$). In this specific case, the jet exit Mach number (M_{jet}) is approximately 0.51, while the "acoustic Mach number" ($\frac{u_{jet}}{a_{ref}}$) is approximately 0.5. The jet discharges into a quiescent (non-moving) air environment during the experiment. However, to accommodate certain CFD codes where achieving flow into quiescent air is challenging, the CFD computations are conducted with a very low background ambient condition ($M_{ref} = 0.01$), moving from left to right, aligned with the jet direction. Although this boundary condition difference does have some effect, extensive testing has revealed that its influence is relatively small, and $M_{ref} = 0.01$ represents a reasonable compromise. The appropriate jet conditions are estab-

lished by setting total pressure and temperature at the inflow face within the jet, as depicted in Figure 3.3.

It is important to emphasize that this axisymmetric case makes use of a periodic (rotated) grid system with appropriate boundary conditions on the periodic sides of the grid. Notably, a grid with a significantly larger domain (1.5 times larger radial extent and twice the distance upstream) has also been run, yielding CFD results almost identical to those obtained from the current grid provided. For our study, we use a 3D axisymmetric grids (two planes rotated through 1° from each other; one plane rotated $+0.5°$ from the $x - z$ plane, and the other plane rotated $-0.5°$ from the $x - z$ plane). The available experimental data are measured velocities as well as turbulence quantities downstream of the jet exit using PIV [95].



(a) Boundary conditions for ASJ computations.



(b) Mesh used for ASJ computations.

Figure 3.3: ASJ computational setup.

3.2.2 Axisymmetric near-sonic jet (ANSJ)

The axisymmetric near sonic jet validation case uses the same Acoustic Research Nozzle 2 (ARN2) as in section 3.3, featuring a radius of $1inch$ ($25.4mm$). In this specific case, the jet exit Mach number (M_{jet}) is approximately 0.985, while the "acoustic Mach number" $\frac{u_{jet}}{a_{ref}}$) is approximately 0.9. In the experiment, the axisymmetric jet exits also into a quiescent (non-moving) air environment, making it an ideal setup for validation purposes. However, the practical implementation of this scenario in certain CFD codes poses challenges due to difficulties in achieving flow into quiescent air. As a result, the CFD computations are conducted with a very low background ambient condition ($M_{ref} = 0.01$), featuring a slow flow moving from left to right, aligned with the jet direction. Despite this boundary condition difference, thorough testing has shown that its influence on the results is relatively small. To achieve the desired jet conditions, total pressure and temperature at the inflow face within the jet are meticulously set (see Figure 3.4). The accompanying figure illustrates the specific setup used to achieve the appropriate flow conditions. The available high-fidelity data is PIV data of [95, 96].

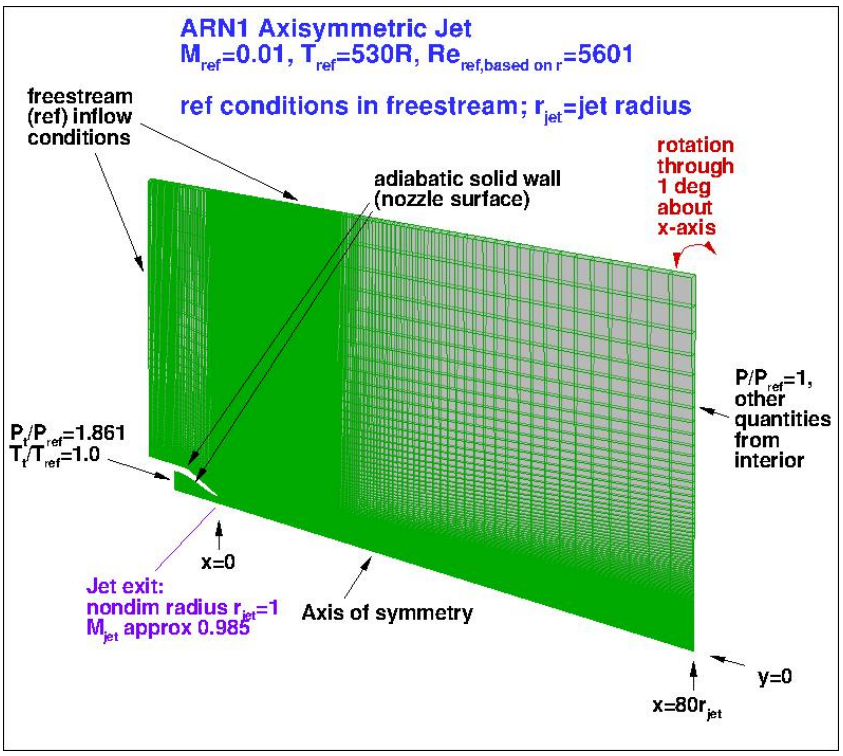


Figure 3.4: Mesh used for ANSJ computations.

3.3 Separated flow cases

3.3.1 Converging diverging channel (CD)

This configuration corresponds to a 2D channel of half-height H with an asymmetric bump of height $h \simeq \frac{2}{3H}$ located on the bottom wall. The Reynolds number (based on the channel half-height and inlet conditions) is $Re_H = 12600$. A small separation occurs near the throat of the bump. For this test case, high-fidelity DNS data from [97] are available. The RANS simulations are based on a mesh of 140×100 cells (see Figure 3.5). A velocity profile obtained from a companion channel-flow simulation is imposed at the inlet of the computational domain.

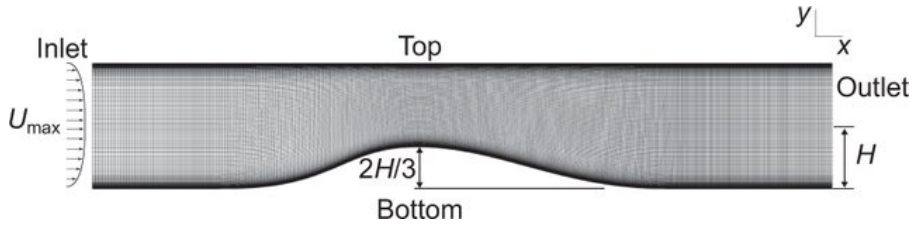


Figure 3.5: Mesh used for CD computations.

3.3.2 Curved backward facing step (CBFS)

The case consists in a 2D flow over a gently-curved backward-facing step of height H, producing a separation bubble. The upstream channel height is $8.52H$ and the Reynolds number Re_H, based on the inlet velocity and step height, is 13700. High-fidelity LES data from [98] are used for training. For the RANS simulations, the mesh consists of 140×150 cells (see Figure 3.6). Slip conditions are used at the upper boundary, and a velocity profile obtained from a fully-developed boundary layer simulation is set at the domain inlet.

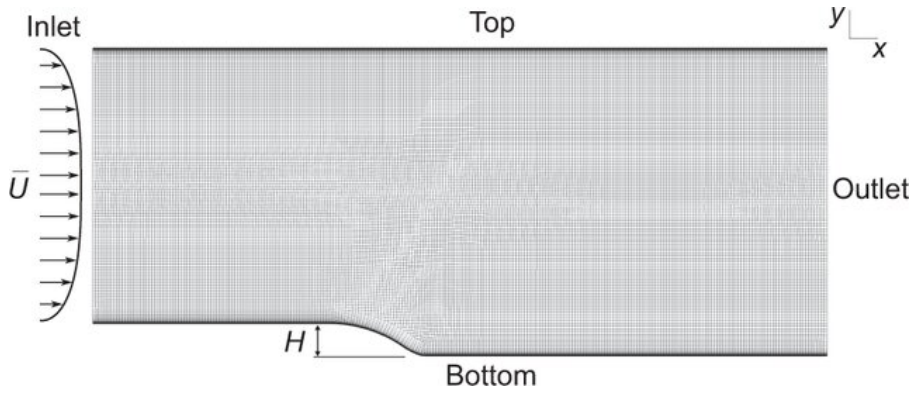


Figure 3.6: Mesh used for CBFS computations.

3.3.3 Periodic hills (PH)

This case consists of a flow through a channel constrained by periodic restrictions (hills) of height H. For a channel segment comprised between two adjacent hills, the flow separates on the lee-side of the first hill and reattaches between the hills. The test case has been widely investigated in the literature, both experimentally and numerically. The high-fidelity LES data

used in the present work are from [99] for $Re_H = 10595$, where Re_H is a Reynolds number based on the bulk velocity in the restricted section and the hill height. Our RANS simulations use a computational grid consisting of 120×130 cells (see Figure 3.7). Cyclic boundary conditions are used at the inlet and outlet and a forcing term is applied to maintain a constant flow rate through the channel.

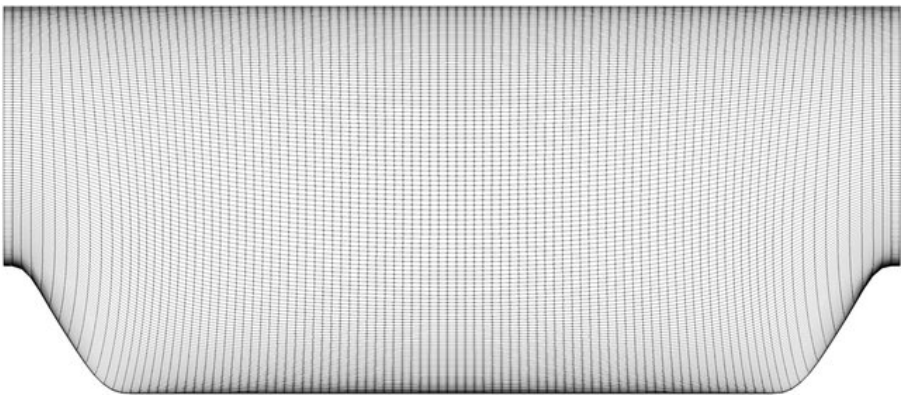


Figure 3.7: Mesh used for PH computations.

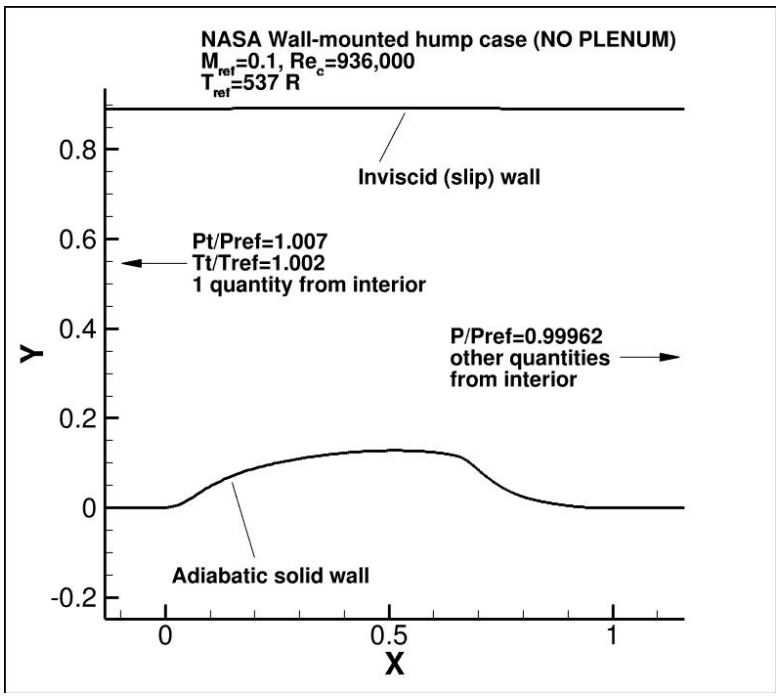
3.3.4 NASA 2D wall-mounted hump (2DWMH)

The main objective of this flow case is to evaluate the performance of various turbulence models in predicting 2D separation from a smooth body due to an adverse pressure gradient, as well as the subsequent reattachment and recovery of the boundary layer. Since its inception, this specific case, along with similar cases involving flow control, has posed a considerable challenge for all existing RANS models. Notably, these models tend to underestimate the turbulent shear stress in the separated shear layer, leading to an overestimation of the length of the separation bubble.

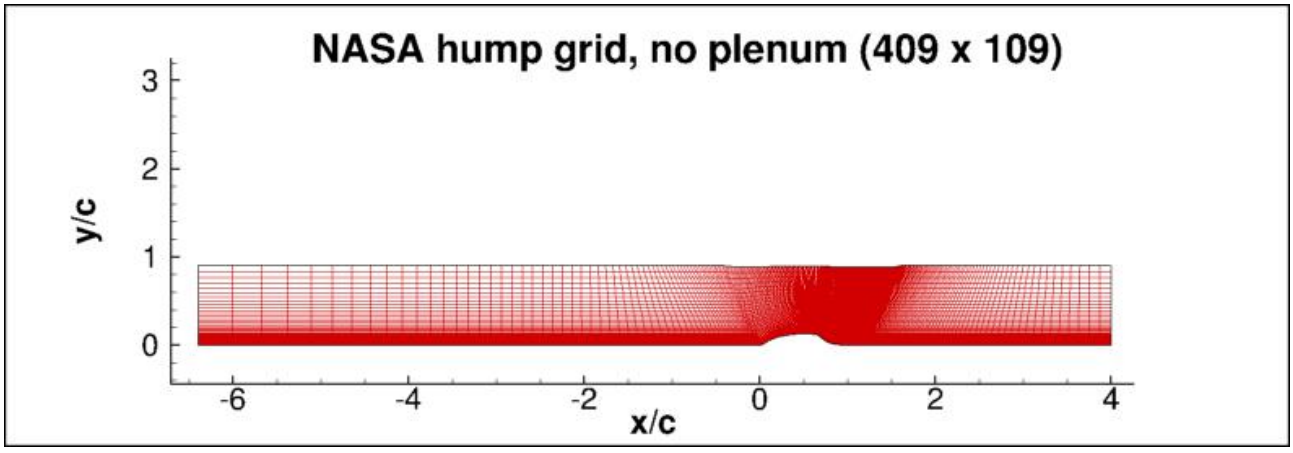
In this investigation, the reference freestream velocity is maintained at approximately $34.6m/s$ (corresponding to a Mach number of 0.1). The incoming fully turbulent boundary layer thickness at position $x/c = -2.14$ is approximately $35mm$, or approximately 8% of the bump's "chord" c (which measures $420mm$). The back pressure is adjusted to achieve the desired flow conditions. To facilitate the natural development of a fully turbulent boundary layer upstream of the hump, the upstream "run" length is carefully chosen. Additionally, the upper boundary

in the CFD simulations is modeled as an inviscid (slip) wall and is adjusted with a contour to approximately account for the blockage effects caused by the end plates in the corresponding experimental setup. The provided figure depict the configuration and boundary conditions employed in this particular case. The notations used include P_t for total pressure, P for static pressure, and T_t for total temperature. The Reynolds number Re_c chosen for this study aligns with the one utilized in the NASA CFDVAL2004 workshop[†], specifically 936,000 (see Figure 3.8). The CFD simulation is run on a 409×109 mesh. It is worth mentioning that a minor disparity exists when compared to the Reynolds number reported in the reference by [100] in 2006, which stands at $Re_c = 929,000$. Nevertheless, this discrepancy, amounting to less than 1%, is considered to be of negligible significance.

[†]https://turbmodels.larc.nasa.gov/Other_exp_Data/cfdval2004_exp.html



(a) Boundary conditions for WMH computations.



(b) Mesh used for WMH computations.

Figure 3.8: WMH computational setup.

Chapter 4

Sparse Bayesian Learning of data-driven model corrections for turbulent separated flows

Contents

In this chapter, we introduce the SBL-SpaRTA framework for deriving sparse stochastic EARM-type corrections for turbulence models in the specific case of turbulent separated flows. To do so, we start in Section 4.1 by detailing our methodology to prepare high-fidelity data

for addressing the turbulence model learning problem. Specifically, we review the k-corrective-frozen procedure[61] and Pope's decomposition[19] for expressing the required corrections. Moving on, Section 4.2 provides an extensive examination of the Sparse Bayesian Learning (SBL) algorithm. Finally, in Section 4.3 we apply the SBL algorithm to the specific case of separated flow in a dedicated manner. The main findings of this chapter have also been documented and published in [101].

4.1 Frame-invariant model corrections

4.1.1 Problem formulation

Following [61], we seek to correct the Boussinesq constitutive model for b_{ij} by introducing a second-order symmetric and traceless tensor $\mathbf{b}^{\Delta} = \left(b_{ij}^{\Delta}\right)$, referred-to as the extra-anisotropy, such that:

$$\tau_{ij} = 2k\left(\frac{1}{3}\delta_{ij} + b_{ij}^{0} + b_{ij}^{\Delta}\right), \quad b_{ij}^{0} = -\frac{\nu_t}{k}S_{ij}, \quad b_{ij} = b_{ij}^{0} + b_{ij}^{\Delta} \tag{4.1}$$

Based on [102, 60, 61], the extra-anisotropy is assumed to be a function of the mean velocity gradient only. By virtue of the Cayley–Hamilton theorem, $\mathbf{b}^{\Delta}$ can then be projected onto a minimal integrity basis of ten tensors polynomials with coefficients depending on the five invariants of the velocity tensor gradient for the general case of 3D flow [19]:

$$b_{ij}^{\Delta} = \sum_{n=1}^{10} T_{ij}^{(n)} \alpha_n^{\Delta}(I_1, ..., I_5) \tag{4.2}$$

For the 2D flows considered in this work, only the first three tensors are linearly independent, and only two invariants are nonzero:

$$b_{ij}^{\Delta} = \sum_{n=1}^{3} T_{ij}^{(n)} \alpha_n^{\Delta}(I_1, I_2) \tag{4.3}$$

where :

$$\begin{cases} T_{ij}^{(1)} = S_{ij}^{*} \\ T_{ij}^{(2)} = S_{ik}^{*}\Omega_{kj}^{*} - \Omega_{ik}^{*}S_{kj}^{*} \\ T_{ij}^{(3)} = S_{ik}^{*}S_{kj}^{*} - \frac{1}{3}\delta_{ij}S_{mn}^{*}S_{mn}^{*} \\ I_1 = S_{mn}^{*}S_{mn}^{*} \\ I_2 = \Omega_{mn}^{*}\Omega_{mn}^{*} \end{cases} \tag{4.4}$$

$\mathbf{S}^* = \{S^*_{ij}\}$ is the non-dimensional strain rate tensor, $\mathbf{\Omega}^* = \{\Omega^*_{ij}\}$ the non-dimensional rotation rate tensor, and α_n^Δ are function coefficients depending on the first two invariants of the velocity gradient tensor (I_1 and I_2). As in [19, 20], the tensors $T_{ij}^{(n)}$ are made non-dimensional with the timescale ω^{-1}.

At this point, the general representation adopted for the Reynolds stress anisotropy in 2D consists of the baseline Boussinesq term $\mathbf{b}^0$ and of the three additional tensor terms of equation (4.3). The first of such terms is a linear function of the normalized strain rate $\mathbf{S}^*$ and, as a consequence, it has the same formal structure as $\mathbf{b}^0$. It can be interpreted as an additive correction of the linear eddy viscosity ν_t. The other two terms are quadratic in $\mathbf{S}^*$ and $\mathbf{\Omega}^*$, and their structure is similar to the nonlinear terms introduced, *e.g.* in the EARSM models of [20] and [21]. In such models, the scalar coefficients α_n^Δ are obtained by repeated application of the Cayley-Hamilton theorem, along with regularization assumptions, and the closure coefficients are determined by identification with existing RSM models. The latter are in turn calibrated for a narrow set of so-called "canonical" flows (such as decaying isotropic homogeneous turbulence, homogeneous free shear flows, etc.). We refer to the above-mentioned references for further details.

In the following, we search expressions of the function coefficients $\alpha_n^\Delta(I_1, I_2)$ through Sparse Bayesian learning (see section 4.2). Our goal is to develop customized model terms that best fit the data available for the class of flows at hand by automatically selecting terms from a redundant functions dictionary. This differs from the traditional turbulence modeling approach in that a completely data-driven (or *"openly empirical"* [2]) model structure is adopted, while model coefficients are still calibrated for a reduced set of flows. However, incompressible separated flows are here considered instead of canonical flows. Of course, there is in principle no guarantee that the resulting models generalize well to radically different flow configurations. This is acceptable as long as the scope is not to develop a "universal" model but rather a specialized model for a particular class of flows. Furthermore, since the present models rely on the assumption that the Reynolds stress is a function of the local velocity gradient and of a

single timescale, they are subject to similar restrictions. Specifically, they are expected to work for nearly-homogeneous, high Reynolds flows. More general formulations are currently under investigation, and will make the object of forthcoming studies.

In the following, the functions α_n^{Δ} are sought by a supervised ML procedure whose first step is to extract the LEVM model error with respect to the high-fidelity data, *i.e.* :

$$b_{ij}^{\Delta,hf} = b_{ij}^{HF} - b_{ij}^{0,hf} \tag{4.5}$$

Several studies have shown that when Reynolds-stress corrections learned from DNS data are propagated through the RANS equations, output quantities such as the velocity fields are not error-free. The reason for that is the ill-conditioning of the Navier-Stokes operator. A strategy for circumventing ill-conditioning problems consists in separating the correction into an "implicit" part, where $\mathbf{b}^0$ is corrected by using an "exact" linear eddy viscosity learned from data instead of the baseline one, and an "explicit" part corresponding to a purely nonlinear extra anisotropy term (see [103]).

In this work we adopt instead the approach of [104], which consists in correcting the linear eddy viscosity and the turbulent time scale ω^{-1} indirectly, by accounting for model-form errors in the auxiliary turbulent transport equations. This favors the discovery of a model formulation that is consistent with the RANS solver, thus reducing propagation errors. More precisely, we use the k-corrective-frozen methodology of [61]. The latter consists in solving the turbulent transport equations with frozen high-fidelity values for all quantities but ω:

$$\begin{cases} \dfrac{\partial k}{\partial t} + U_j \dfrac{\partial k}{\partial x_j} = P_k + R^{HF} - \beta^* k\omega + \dfrac{\partial}{\partial x_j}\left((\nu + \sigma_k \nu_t)\dfrac{\partial k}{\partial x_j}\right) \\ \dfrac{\partial \omega}{\partial t} + U_j \dfrac{\partial \omega}{\partial x_j} = \dfrac{\gamma}{\nu_t}(P_k + R^{HF}) - \beta\omega^2 + \dfrac{\partial}{\partial x_j}\left((\nu + \sigma_\omega \nu_t)\dfrac{\partial \omega}{\partial x_j}\right) \end{cases} \tag{4.6}$$

In the preceding equations, k and U are evaluated using high-fidelity data; the production of turbulent kinetic energy is computed by adding to the Boussinesq Reynolds tensor the high-fidelity extra anisotropy:

$$P_k = \min\left(2k\left(-\frac{\nu_t}{k}S_{ij} + b_{ij}^{\Delta,hf}\right)\frac{\partial U_i}{\partial x_j}, 10\beta^*\omega k\right) \tag{4.7}$$

The additional corrective term R (referred to as R^{HF} in Equation 4.6) has been introduced in the equations for k and ω. This decision followed the resolution of the ω transport equation using the frozen flow fields (U and tau_{ij}), and subsequently using the resulting field in the k equation. It was observed that the residual remained non-null throughout this process. This outcome was anticipated, as the RANS equations do not correspond to the exact formulation of the Navier-Stokes equations and are known for their ill-conditioning. Consequently, the introduction of this term into the closure problem was deemed necessary. Finally, β^*, σ_k and σ_ω are the $k-\omega$ SST constants and can be found in [12].

A modeling ansatz for the residual R^{HF} is obtained by rewriting it in a form similar to the turbulent kinetic energy production:

$$R^{HF} \approx R = 2kb_{ij}^R \frac{\partial U_i}{\partial x_j} \tag{4.8}$$

with the fundamental difference that it can take both positive (extra production) and negative (under-production) values. The tensor $\mathbf{b}^R$ is projected onto the same integrity basis as $\mathbf{b}^\Delta$:

$$b_{ij}^R = \sum_{n=1}^{3} T_{ij}^{(n)} \alpha_n^R(I_1, I_2) \tag{4.9}$$

thus introducing a new set of unknown functions α_n^R that are sought by Machine Learning as the α_n^Δ, using now R^{HF} as the learning target.

4.1.2 Final regression task

To identify an expression for α_n^Δ and α_n^R, we select a library $\boldsymbol{\mathcal{B}}$ of monomials of the invariants I_1 and I_2:

$$\boldsymbol{\mathcal{B}} = \{I_1^l,\ I_2^m,\ I_1^p I_2^q \parallel 0 \leq l, m \leq 9,\ 2 \leq p+q \leq 4\} \tag{4.10}$$

leading to 25 candidate terms for each function ($\{\alpha_n^\Delta\}_{n=1}^3$ and $\{\alpha_n^R\}_{n=1}^3$), *i.e.* a total of 25 functions $\times$ 2 corrections $\times$ 3 tensors $= 150$ candidate functions. With such a large number of functional terms used to represent the learning targets, an efficient learning procedure is needed to fastly select a parsimonious model (*i.e.* a sparse model involving a small subset

of functions selected from the initial redundant dictionary) and limit the risk of overfitting the data.

Of note, hereafter we prefer to learn $a_{ij}^{\Delta} = \tau_{ij} - (\frac{2}{3}k\delta_{ij} - 2\nu_t S_{ij}) = 2kb_{ij}^{\Delta}$ rather than b_{ij}^{Δ} as in [104] because the value of b_{ij}^{Δ} at the wall is mathematically undetermined:

$$\lim_{y^+\to 0} b_{ij}^{\Delta} = \lim_{y^+\to 0} \frac{a_{ij}^{\Delta}}{2k} = \frac{0}{0}$$

making use of data and physical interpretation of the results difficult close to the wall. By multiplying b_{ij}^{Δ} by $2k$, we ensure that our learning target, as well as the basis functions, have a determined physical value at the wall, and we prevent numerical errors. With this choice, the learning problem thus becomes:

$$\begin{cases} \mathbf{t}^{\Delta} = \mathbf{C}_{b^{\Delta}} \boldsymbol{\theta}_{b^{\Delta}} \\ \mathbf{t}^{R} = \mathbf{C}_{R} \boldsymbol{\theta}_{R} \end{cases} \tag{4.11}$$

where:

$$\begin{cases} \mathbf{t}^{\Delta} = 2k(b_{11|k=0}^{\Delta}, ..., b_{11|k=K}^{\Delta}, ..., b_{33|k=0}^{\Delta}, ..., b_{33|k=K}^{\Delta})^T \\ \mathbf{t}^{R} = (R_{|k=0}, ..., R_{|k=K})^T \\ \mathbf{C}_{b^{\Delta}} = 2k \begin{bmatrix} T_{11|k=0}^{(1)} & I_1 T_{11|k=0}^{(1)} & ... & I_1^2 I_2^2 T_{11|k=0}^{(3)} \\ T_{11|k=1}^{(1)} & I_1 T_{11|k=1}^{(1)} & ... & I_1^2 I_2^2 T_{11|k=1}^{(3)} \\ ... & ... & ... & ... \\ T_{33|k=K}^{(1)} & I_1 T_{33|k=K}^{(1)} & ... & I_1^2 I_2^2 T_{33|k=K}^{(3)} \end{bmatrix} \\ \mathbf{C}_{R} = 2k \begin{bmatrix} T_{ij}^{(1)} \partial_j U_{i|k=0} & I_1 T_{ij}^{(1)} \partial_j U_{i|k=0} & ... & I_1^2 I_2^2 T_{ij}^{(3)} \partial_j U_{i|k=0} \\ ... & ... & ... & ... \\ T_{ij}^{(1)} \partial_j U_{i|k=K} & I_1 T_{ij}^{(1)} \partial_j U_{i|k=K} & ... & I_1^2 I_2^2 T_{ij}^{(3)} \partial_j U_{i|k=K} \end{bmatrix} \end{cases}$$

4.2 The Sparse Bayesian Learning (SBL) algorithm

This section provides details of the SBL algorithm for regression, introduced by Tipping [105] in the case of Support Vector Machine (SVM) models. The approach is then adapted to our turbulence model learning problem.

4.2.1 Problem formulation

First, let us consider a data set of input-output pairs $\{\mathbf{x}_n, t_n\}_{n=1}^N, \mathbf{x}_n \in \mathbf{R}^{N_\mathbf{x}}, t_n \in \mathbf{R}$.

We follow the standard probabilistic formulation where we consider that the targets $\mathbf{t} = (t_1, ..., t_N)^T$ are sampled from a linear model $\{\mathbf{c}; \boldsymbol{\theta}\}$ with additive noise ϵ:

$$t(\mathbf{x}; \boldsymbol{\theta}) = \Big(\mathbf{c}(\mathbf{x})\Big)^T \boldsymbol{\theta} + \epsilon = \sum_{j=1}^{M} c_j(\mathbf{x})\theta_j + \epsilon \tag{4.12}$$

such that:

$$\mathbf{t} = \mathbf{C}\boldsymbol{\theta} + \boldsymbol{\epsilon} = \sum_{j=1}^{M} \theta_j \mathbf{C}_j + \boldsymbol{\epsilon} \tag{4.13}$$

where $\mathbf{C} \in \mathbf{R}^{N \times M}$, $\{\mathbf{C}\}_{ij} = c_j(\mathbf{x}_i)$, is the design matrix, $\mathbf{C}_j \in \mathbf{R}^M$ is the j^{th} column of the design matrix $\mathbf{C}$, $\boldsymbol{\theta} = (\theta_1, ..., \theta_M)^T$ is the vector of the model's parameters, and $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_N)^T$ is a vector of independent noise processes assumed to be Gaussian of zero-mean and variance σ^2.

Following these assumptions, one can derive the likelihood of observing the data given the model parameters $\boldsymbol{\theta}$ and σ^2:

$$p(\mathbf{t}|\boldsymbol{\theta}, \sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma^2}||\mathbf{t} - \mathbf{C}\boldsymbol{\theta}||^2\right) \tag{4.14}$$

The parameters are traditionally determined by maximizing the logarithmic likelihood of observing the data knowing the model parameters, also known as the "type–I log likelihood", *i.e.*

$$\mathcal{L}_\mathcal{I} = \log p(\mathbf{t}|\boldsymbol{\theta}, \sigma^2) = -\frac{1}{2}\left(N \log(2\pi\sigma^2) + \frac{1}{\sigma^2}||\mathbf{t} - \mathbf{C}\boldsymbol{\theta}||^2\right) \tag{4.15}$$

However, the maximum-likelihood estimate can easily lead to severe overfitting as the number of model parameters grow, ending up with a model capturing the data noise rather than their proper dynamics.

4.2.2 Hierarchical priors specification

The SBL algorithm proposes an alternative to the maximization of the likelihood $\mathcal{L}_{\mathcal{I}}$. The idea motivated by the work of [106], and suggests to constrain model's parameters by defining an explicit prior distribution over them and their underlying hyperparameters, and genuinely use this structure under Bayes' theorem to infer the "relevance" of the model's parameters. The principle is similar to the Automatic Relevance Determination (ARD) initially introduced by [107, 108]. We follow the choice made in [105] of zero-mean Gaussian prior distribution over $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{i=1}^{M} \mathcal{N}(\theta_i|0, \alpha_i^{-1}) \tag{4.16}$$

with $\boldsymbol{\alpha}$ a vector of M unknown hyperparameters, controlling the width of the marginal prior for the parameters θ_i, *i.e.* the relevance of such parameters. When a hyperparameter α_i is high, the prior distribution of θ_i becomes narrowly centered around 0, thus making the coefficient *irrelevant*. The inference problem now consists in estimating the unknown joint distribution of the hyperparameters $\boldsymbol{\alpha}$ and σ^2. For that purpose, we use a hierarchical prior approach, where we assign $\boldsymbol{\alpha}$ a Laplace prior probability distribution:

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{M} \frac{\lambda_i}{2} \exp(-\frac{\lambda_i}{2\alpha_i}) \tag{4.17}$$

where $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_M)^T$ are additional hyperparameters that must be specified by the modeler. The choice of such priors, referred to as *demi-Bayesian LASSO* and introduced by [109], is motivated by the improved sparsity conferred to the algorithm with respect to the original formulation [105] where Gamma prior distributions were adopted. By increasing λ_i, we impose sharper prior distributions for the $\frac{1}{\alpha_i}$, *i.e.* higher values of the hyperparameter α_i become more likely, and consequently the corresponding model coefficient probability distribution $p(\theta_i)$ become sharply centered around 0. When reaching a certain user-defined limit, the very sharp

probability distribution is considered as an indicator of an irrelevant model coefficient and therefore the latter is removed. For convenience, we fix $\lambda_1 = ... = \lambda_M = \lambda$. The formula is completed by specifying a uniform hyper-prior on $\frac{1}{\sigma^2}$ (over a logarithmic scale).

Now, using Bayes' rule, we seek for the posterior joint distribution of $\boldsymbol{\theta}$, $\boldsymbol{\alpha}$ and σ^2:

$$p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2|\mathbf{t}) = \frac{p(\mathbf{t}|\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2)}{p(\mathbf{t})} \tag{4.18}$$

where $p(\mathbf{t}|\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2)$ is the model likelihood (4.14) where we made explicit the dependency on hyperparameter vector $\boldsymbol{\alpha}$. $p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2)$ is the joint prior probability and $p(\mathbf{t})$ is the model evidence. Then, given a new test point $\mathbf{x}_*$, predictions are made for the corresponding target t_*, in terms of the predictive distribution:

$$p(t_*|\mathbf{t}) = \int p(t_*|\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2|\mathbf{t})d\boldsymbol{\theta} \tag{4.19}$$

Unfortunately, $p(\mathbf{t})$ is a multi-dimensional integral

$$p(\mathbf{t}) = \int p(\mathbf{t}|\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2)d\boldsymbol{\theta}d\boldsymbol{\alpha}d\sigma^2 \tag{4.20}$$

and is generally not straightforward to compute. Sampling strategies like Markov Chain Monte-Carlo could be used to approximate the integral, but they generally require a very large number of samples. Instead, the joint prior distribution is rewritten:

$$p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \sigma^2|\mathbf{t}) = p(\boldsymbol{\theta}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t}) \tag{4.21}$$

In the preceding equation, $p(\boldsymbol{\theta}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as a result of (4.14), (4.16) and (4.17), where the posterior covariance and mean are respectively:

$$\begin{cases} \boldsymbol{\Sigma} = \left(\mathbf{A} + \frac{1}{\sigma^2}\mathbf{C}^T\mathbf{C}\right)^{-1} \\ \boldsymbol{\mu} = \frac{1}{\sigma^2}\boldsymbol{\Sigma}\mathbf{C}^T\mathbf{t} \end{cases}, \quad \mathbf{A} = \text{diag}(\alpha_1, ..., \alpha_M). \tag{4.22}$$

On the other hand, regarding $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t})$, we follow [105] and adopt a point estimate method by considering that the hyperparameter posterior $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t})$ can be represented as a multi-dimensional Dirac function centered at the most probable set of values $\boldsymbol{\alpha}_{MP}$ and σ^2_{MP}. We do so on the basis that this point-estimate is representative of the posterior in the sense that functions

generated utilizing the posterior mode values are near-identical to those obtained by sampling from the full posterior distribution. It is important to realize that this does not necessitate that the entire mass of the posterior be accurately approximated by the delta-function:

$$p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t}) \simeq \delta(\boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) \tag{4.23}$$

The SBL algorithm becomes the search for these most probable values of this posterior, *i.e.* the maximization of $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t})$ with respect to $\boldsymbol{\alpha}$ and σ^2.

4.2.3 Optimization of hyperparameters

Using Bayes' rule, the hyperparameter posterior is of the form:

$$p(\boldsymbol{\alpha}, \sigma^2|\mathbf{t}) \propto p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha})p(\sigma^2) \tag{4.24}$$

and hence $p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha})p(\sigma^2)$ must be maximised. Since we made the choice of a uniform prior for σ^2, the loss function to maximize becomes:

$$\begin{aligned}
\mathcal{L}_{\mathcal{II}} &= \log\Big(p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha})\Big) \\
&= \log\left(\int p(\mathbf{t}|\boldsymbol{\theta}, \sigma^2)p(\boldsymbol{\theta}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})d\boldsymbol{\theta}\right) \\
&= -\frac{1}{2}\left[\log|\sigma^2\mathbf{I} + \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T| + \mathbf{t}^T(\sigma^2\mathbf{I} + \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T)\mathbf{t}\right] - \lambda\sum_{i=0}^{M}\frac{1}{\alpha_i}
\end{aligned} \tag{4.25}$$

$\mathcal{L}_{\mathcal{II}}$ is referred to as *the marginal likelihood* or *the evidence of hyperparameters*, and its maximisation as the *type-II maximum likelihood method* [110] or *the evidence procedure* [106].

In contrast with traditional regression methods where the optimal values of hyperparameters are determined by a cross-validation, the SBL algorithm determines their optimal values by iteratively maximizing the evidence of hyperparameters $\mathcal{L}_{\mathcal{II}}$ with respect to α_i and $\frac{1}{\sigma^2}$, leading to:

$$\begin{cases}
\alpha_i^{new} = \dfrac{1 + \sqrt{1 + 8\lambda(\mu_i^2 + \Sigma_{ii})}}{2(\mu_i^2 + \Sigma_{ii})} \\
(\sigma^2)^{new} = \dfrac{||\mathbf{t} - \mathbf{C}\boldsymbol{\mu}||^2}{N - \sum_{i=1}^{M}(1 - \alpha_i\Sigma_{ii})}
\end{cases} \tag{4.26}$$

For more details see [105, 109]. Finally, equation (4.26) still depends on the sparsity-promoting hyperparameter λ and its optimal value for the learning problem can be determined via a grid search.

4.2.4 Making Predictions

At convergence of the hyperparameters estimation procedure, we make predictions based on the posterior distribution over the weights, conditioned on the maximizing values $\boldsymbol{\alpha}_{MP}$ and σ^2_{MP}. We can then compute the predictive distribution for a new datum $\mathbf{x}_*$ using:

$$p(t_*|\mathbf{t}, \boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) = \int p(t_*|\boldsymbol{\theta}, \sigma^2_{MP}) p(\boldsymbol{\theta}|\mathbf{t}, \alpha_{MP}, \sigma^2_{MP}) d\boldsymbol{\theta} \tag{4.27}$$

Since both terms in the integrand are Gaussian, this is readily computed, giving:

$$p(t_*|\mathbf{t}, \boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) = \mathcal{N}(t_*|y_*, \sigma^2_*) \tag{4.28}$$

with:

$$\begin{cases} y_* &= \boldsymbol{\mu}^T \mathbf{C}(\mathbf{x}_*) \\ \sigma^2_* &= \sigma^2_{MP} + \mathbf{C}(\mathbf{x}_*)^T \boldsymbol{\Sigma} \mathbf{C}(\mathbf{x}_*) \end{cases} \tag{4.29}$$

4.2.5 Application: The Relevance Vector Machine (RVM)

In this example, we are given a set $N+1$, $(N = 50)$ of uniformly distributed input points between 0 and 1, $\left\{x_n = \frac{n}{N}\right\}_{n=0}^{N}$, along with corresponding targets $\{t_n\}_{n=0}^{N}$, generated using a radial basis function (RBF) kernel $\mathbf{C}$, a sparse vector of parameters $\boldsymbol{\theta}$ and a Gaussian noise $\boldsymbol{\epsilon}$:

$$t_n(x_n, \boldsymbol{\theta}) = \sum_{i=0}^{N} C(x_n, x_i)\theta_i + \epsilon_n \tag{4.30}$$

with

$$\begin{cases} C(x_n, x_i) = r^2 \exp\left(-\dfrac{-(x_n - x_i)^2}{2l^2}\right) \\ \epsilon_n \sim \mathcal{N}(0, \sigma^2) \end{cases} \tag{4.31}$$

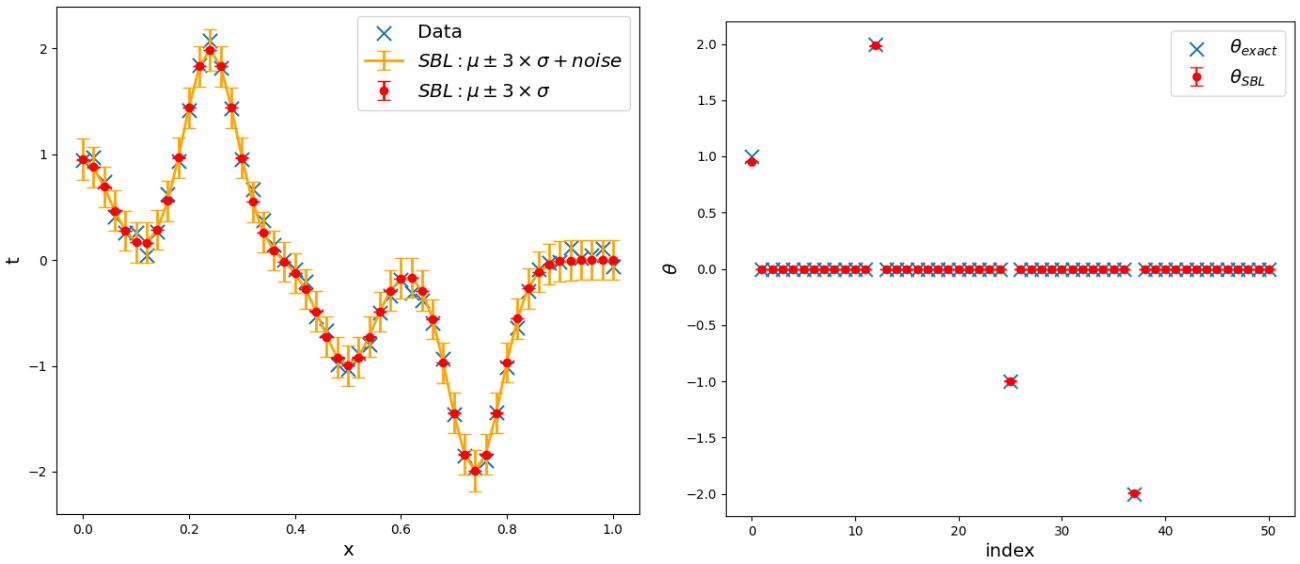
The learning problem using this kernel and the Sparse Bayesian Learning is known as the "Relevance Vector Machine" [105]. The vector $\boldsymbol{\theta}$ is deliberately designed with all coefficients

being zero, except at four arbitrary locations:

$$\theta_0 = 1, \quad \theta_{12} = 2, \quad , \theta_{24} = -1, \quad , \theta_{37} = -2 \tag{4.32}$$

As for the other parameters:

$$\sigma = 0.05, \quad r = 1, \quad l = 0.05 \tag{4.33}$$



(a) SBL model's output vs. exact training data output.

(b) θ_{exact} vs. θ_{SBL}.

Figure 4.1: Training results of the SBL model over the example's synthetic data.

Upon reaching convergence, the model parameters, estimated noise as well as the prediction of the output targets are provided in Figure 4.1 and described below:

$$\begin{cases} \theta_0 & = 0.95247783 \pm 3 \times 0.00141239 \\ \theta_{12} & = 1.98586961 \pm 3 \times 0.00088691 \\ \theta_{24} & = -0.99715879 \pm 3 \times 0.00087792 \\ \theta_{37} & = -1.99187662 \pm 3 \times 0.00088693 \\ \theta_{i \neq 0,12,24,37} & = 0 \\ \sigma_{MP} & = 0.06300819224654165 \end{cases} \tag{4.34}$$

The application of the Sparse Bayesian Learning (SBL) algorithm to the dataset yields notable outcomes with potential implications for data analysis and modeling. Firstly, the consistency

between the sparsity of the SBL-derived coefficient vector and the exact model coefficients vector is a positive indicator of the algorithm's ability to effectively identify relevant features while reducing the model complexity, as one can see in both Figures 4.1a and 4.1b. This characteristic is particularly beneficial in high-dimensional data sets, where the risk of overfitting becomes higher.

Secondly, the proximity of the SBL-estimated coefficients to the exact values, with relatively narrow standard deviations, as witnessed in Figure 4.1b suggests that the algorithm is capable of producing precise parameter estimates. The narrow standard deviations imply that the model coefficients are well-informed and less influenced by random variations, enhancing the reliability of the results.

Moreover, the accurate estimation of the observation noise, as evidenced by the closeness of the estimated σ value to the known value used in data generation, indicates the algorithm's capability of handling noisy data. This ability is essential for robust modeling in real-world scenarios where data imperfections are common.

The SBL algorithm's success in capturing the dynamics among noisy data as displayed in Figure 4.1a. A noteworthy observation is its ability to reproduce complex relationships and patterns, enhancing the algorithm's versatility for various data analysis tasks.

Additionally, the SBL algorithm's provision of probability distributions for the model coefficients enables an intrinsic uncertainty quantification. By characterizing both parametric and observational uncertainties, it provides valuable insights into the confidence and reliability of the model's predictions. Such uncertainty estimation is critical for making informed decisions and improving the interpretability of the results.

4.3 SBL algorithm for data-driven turbulence modeling

In the following, we consider three learning configurations or scenarios, summarized in Table 4.1. Models are trained using full-field high-fidelity data for two flow cases out of three, and tested on the third one (as discussed later). For a given data set scenario, various models are obtained according to the value of the regularization parameter λ. All models obtained from the same scenario s are noted as $\mathbf{M}^{(s)}$.

Scenario	Training set	Model idle
1	CBFS and PH	$\mathbf{M}^{(1)}$
2	CD and PH	$\mathbf{M}^{(2)}$
3	CBFS and CD	$\mathbf{M}^{(3)}$

Table 4.1: Learning scenarios and nomenclature for the resulting models.

4.3.1 Cross-validation methodology

Data-driven SBL models are trained using data available for each scenario in Table 4.1 using different values of the regularization parameter λ within the set $\{10^2, 10^3, 10^4, 10^5, 2\times10^5\}$. The resulting models $M = (\mathbf{M}_{\mathbf{b}^\Delta}, \mathbf{M}_{\mathbf{b}^R})$ (given in Appendix B) take the form:

$$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta} = \sum\limits_n \left(\sum\limits_{l,m} (\mu_{l,m}^{\Delta^{(n)}} \pm \sigma_{l,m}^{\Delta^{(n)}}) I_1^l I_2^m \right) \mathbf{T}^{(n)} \pm \mathbf{1}\epsilon^\Delta \\ \mathbf{M}_{\mathbf{b}^R} = \sum\limits_n \left(\sum\limits_{l,m} (\mu_{l,m}^{R^{(n)}} \pm \sigma_{l,m}^{R^{(n)}}) I_1^l I_2^m \right) \mathbf{T}^{(n)} \pm \mathbf{1}\epsilon^R \end{cases} \tag{4.35}$$

where $\mu_{l,m}^{\Delta^{(n)}}$ and $\sigma_{l,m}^{\Delta^{(n)}}$ (resp. $\mu_{l,m}^{R^{(n)}}$ and $\sigma_{l,m}^{R^{(n)}}$) are respectively the mean and the standard deviation of the probability density function of the coefficient associated to $I_1^l I_2^m$ in $\mathbf{T}^{(n)}$ expansion in $\mathbf{M}_{\mathbf{b}^\Delta}$ (resp. $\mathbf{M}_{\mathbf{b}^R}$), $\mathbf{1}$ is a second order tensor with all elements equal to one, and ϵ^Δ (resp. ϵ^R) is the standard deviation of the noise associated with model $\mathbf{M}_{\mathbf{b}^\Delta}$ (resp. $\mathbf{M}_{\mathbf{b}^R}$).

In Appendix B we report $\mathbf{M}^{(1)}, \mathbf{M}^{(2)}$ and $\mathbf{M}^{(3)}$ derived for each value of λ. One observe that when λ is increased, models $\mathbf{M}_{\mathbf{b}^\Delta}$ become sparser. On the other hand, $\mathbf{M}_{\mathbf{b}^R}$ reduces to only one term regardless of λ, the regularization affecting only the magnitude of the corrective

term, which decreases as λ increases. For values of λ greater than 2×10^5, both $\mathbf{M}_{\mathbf{b}^\Delta}$ and $\mathbf{M}_{\mathbf{b}^R}$ vanish, thus leaving the $k-\omega$ SST model unchanged.

The models are cross-validated by feeding the Maximum–A–Posteriori (MAP) values of the posterior distributions of the model coefficients θ_i to the CFD solver, and computing the Mean Square Error (MSE) over the whole domain with respect to high-fidelity values available for various quantities of interest (QoI):

$$MSE = \frac{1}{N_D} \sum_{\mathbf{x} \in D} \Big(QoI(\mathbf{x}) - QoI^{HF}(\mathbf{x}) \Big)^2 \tag{4.36}$$

where $\mathbf{x}$ is a point of the discrete computational domain D for a given flow case and N_D is the number of grid points in D. The errors obtained for various QoI are reported in Figures 4.2,4.3, 4.4 and 4.5 under the form of histograms, where column patterns refer to dataset used to train the model, while the abscissas indicate the value of the regularization parameter λ. In the figures, grey-shaded bars are used to highlight for each model the test scenario whose data are not used for training. White bars are used for post-diction scenarios, *i.e.* scenarios used to extract the training data. Note that, even for such scenarios, not all of the predicted quantities have been used for training, *i.e.* they also constitute testing data.

The results are compared to those obtained using the baseline model and the three models of [104], obtained by deterministic SpaRTA algorithm. The latter are noted $\mathbf{M}^{(k)}_{Spa}$, with k the number used in the above-mentioned reference. All errors are normalized with respect to the error of the baseline model for the same QoI. The latter is then always assigned an MSE equal to 1.

Figure 4.2 shows the MSE for the streamwise velocity. All learned models have much lower MSE than the baseline, with more complex models (including the $\mathbf{b}^\Delta$ correction) performing somewhat better than the sparser ones, according to the flow configurations and training sets. However, sparser models generalize better through the data sets. More precisely, $\mathbf{M}^{(1)}$ at $\lambda = 2 \times 10^5$ gives the most accurate results for both CD and PH cases, whereas $\mathbf{M}^{(3)}$ at $\lambda = 10^4$ gives the most accurate prediction of horizontal velocity for CBFS case. The selected SBL models exhibit a better or comparable accuracy than the SpaRTA models for this QoI.

Figures 4.3, 4.4 and 4.5 display MSE for the friction coefficient, the turbulent kinetic energy and the Reynolds shear stress, respectively. The figures show that sparser models, such as those obtained for $\lambda = 2 \times 10^5$, tend to be more accurate than complex ones. Of note, best-performing models have $\mathbf{b}^{\Delta} = 0$ and only the $\mathbf{b}^{R}$ correction of the production term is applied, with a coefficient that contributes to increasing the amount of eddy viscosity generated in the separated region to reattach the boundary layer. The amount of such a correction differs from a model to another depending on the training cases. The $\mathbf{b}^{R}$ correction also results in increased turbulent kinetic energy and turbulent shear stress levels throughout the flow, and namely at the wall. The best-performing SBL models tend to be more accurate than the baseline for most QoI and flow cases, but none of the learned models is able to improve results for all QoI and all flow cases simultaneously.

In order to guide the selection of a "best-performing model" as the model providing the best compromise over all cases and QoI, we define a global error metric as follows:

$$m(\lambda) = \sum_i \left(1 - \frac{MSE_{SBL}(QoI_i, \lambda)}{MSE_{BSL}(QoI_i))}\right) \gamma_i = \sum_i m_i(\lambda)\gamma_i \tag{4.37}$$

where $m_i(\lambda)$ measures the relative improvement with respect to the baseline in terms of MSE, γ_i is a weighting coefficient assigned to QoI_i and $m(\lambda)$ is the general model score. If $m_i(\lambda)$ is positive (resp. negative), the model improves (deteriorates) the MSE for QoI_i with respect to the baseline model. The choice of the weighting coefficients determines the variables to be used in cross-validation as well as their relative importance. Since each model is applied to three flow cases, we define for each candidate model an average improvement metric as follows:

$$m^G = \frac{m^{PD_1} + m^{PD_2} + 2m^{TEST}}{4}, \tag{4.38}$$

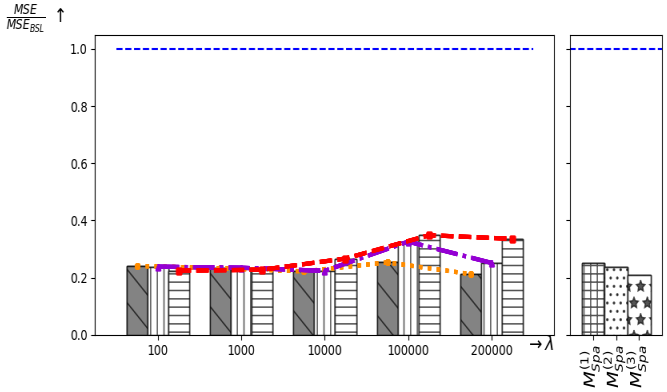
where PD_1 and PD_2 refer to the post-dictions of the training flows and $TEST$ to the test scenario. The latter is assigned a double weight, so that flows in the training and test sets contribute equally to model selection. Finally, the best model $\mathbf{M}^*$ is chosen as the one that maximizes $m^G(\lambda)$:

$$\mathbf{M}^* = \mathbf{M}(\arg_\lambda \max(m^G(\lambda))) = \mathbf{M}(\lambda^*) \tag{4.39}$$

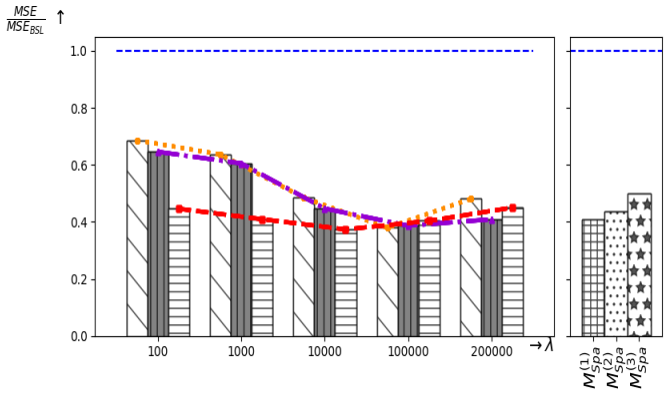
In Table 4.2, we summarize cross-validation results for different choices of the cross-validation (CV) variables for $\mathbf{M}^{(1)}$, $\mathbf{M}^{(2)}$ and $\mathbf{M}^{(3)}$. For this purpose, we denote $\lambda^{*(1)}$ (resp. $\lambda^{*(2)}$ and $\lambda^{*(3)}$) the value of λ that maximizes the average improvement metric for $\mathbf{M}^{(1)}$ (resp. $\mathbf{M}^{(2)}$ and $\mathbf{M}^{(3)}$). The most accurate models over the training and test sets (the ones with the highest m^G) are characterized by a null $\mathbf{b}^\Delta$. $\mathbf{M}^{(2)}$ and $\mathbf{M}^{(3)}$ insure an improved MSE over all the QoI ($\simeq 30\%$) than $\mathbf{M}^{(1)}$ (6%). The latter fails also at giving better results than the baseline model when cross-validated over k, C_f, (k, C_f), (τ_{xy}, C_f) and (k, τ_{xy}, C_f) (marked with "–" in Table 4.2), which suggests that $\mathbf{M}^{(1)}$ needs probably to be further regularized. On the contrary, $\mathbf{M}^{(2)}$ and $\mathbf{M}^{(3)}$ for $\lambda^{*(2)} = 2 \times 10^5$ and $\lambda^{*(3)} = 2 \times 10^5$ resp. combine both a good accuracy and generalizability over all the QoI and flow cases. However, if only the streamwise velocity is selected as the cross-validation target, models with a non null $\mathbf{b}^\Delta$ ($\lambda^{*(2)} = 10^4$ and $\lambda^{*(3)} = 10^4$) outperform models correcting the turbulent production only. This is a limitation intrinsic to the chosen representation of the Reynolds stress tensor as a function of the mean velocity gradient only, and must be addressed in the future, for instance by enlarging the feature set as in [54] or [58].

CV variables	$\lambda^{*(1)}$	$m^G(\lambda^{*(1)})$	$\lambda^{*(2)}$	$m^G(\lambda^{*(2)})$	$\lambda^{*(3)}$	$m^G(\lambda^{*(3)})$
U, k, τ_{xy}, C_f	2×10^5	6.0 %	2×10^5	28.%	2×10^5	31. %
U	2×10^5	74.0 %	10^4	68.0%	10^4	77.0 %
k	–	– %	2×10^5	26.0 %	2×10^5	36.0 %
τ_{xy}	2×10^5	15.0 %	2×10^5	14.0 %	2×10^5	23.0 %
C_f	–	– %	2×10^5	7.0 %	2×10^5	12.0 %
U, k	2×10^5	36.0 %	2×10^5	46.0 %	10^5	46.0 %
U, τ_{xy}	2×10^5	45.0 %	2×10^5	41.0 %	2×10^5	39.0 %
U, C_f	2×10^5	5.0 %	2×10^5	37.0 %	2×10^5	33.0 %
k, τ_{xy}	2×10^5	6.0 %	2×10^5	20.0 %	2×10^5	30.0 %
k, C_f	–	– %	2×10^5	16.0 %	2×10^5	24.0 %
τ_{xy}, C_f	–	– %	2×10^5	10.0 %	2×10^5	17.0 %
U, k, τ_{xy}	2×10^5	29.0 %	2×10^5	36.0 %	2×10^5	38.0 %
U, τ_{xy}, Cf	2×10^5	8.0 %	2×10^5	29.0 %	2×10^5	30.0 %
U, k, Cf	2×10^5	2.0 %	2×10^5	33.0 %	2×10^5	34.0 %
k, τ_{xy}, Cf	–	– %	2×10^5	16.0 %	2×10^5	24.0 %

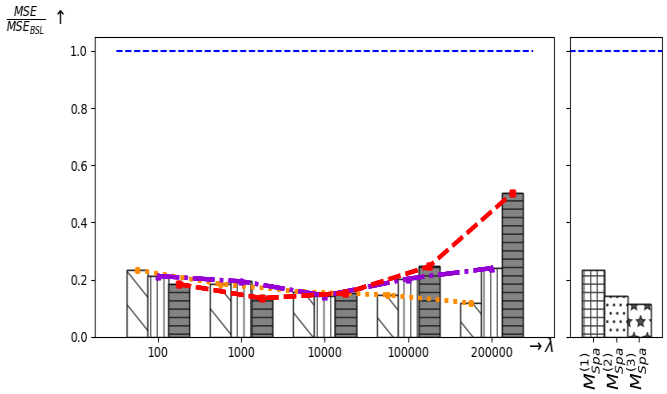
Table 4.2: Cross-validation statistics: best models and general improvement metrics.



(a) Converging-diverging channel.

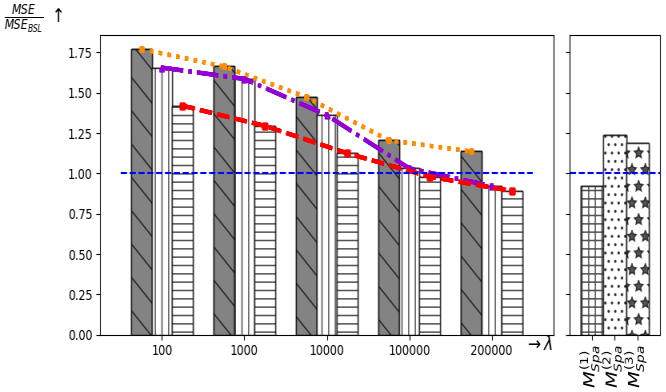


(b) Curved Backward-Facing Step.

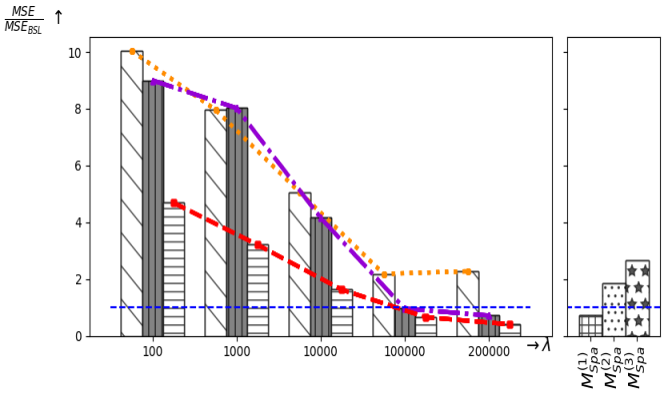


(c) Periodic Hills.

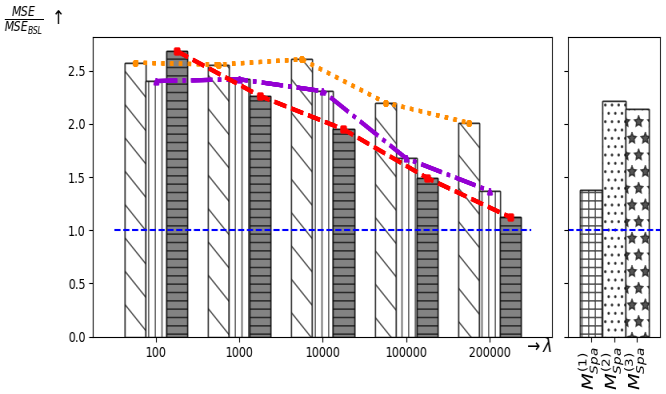
Figure 4.2: MSE of the streamwise velocity (relative to MSE of the baseline model $k-\omega$ SST (- - -)) as a function of λ. Left panel: $\mathbf{M}^{(1)}$ (▨, ·····), $\mathbf{M}^{(2)}$ (▥, -·-·-), $\mathbf{M}^{(3)}$ (▤, - - -); right panel: comparison with models from [61]: $\mathbf{M}^{(1)}_{Spa}$ (▦), $\mathbf{M}^{(2)}_{Spa}$ (⁝⁝), $\mathbf{M}^{(3)}_{Spa}$ (✶).



(a) Converging-diverging channel.

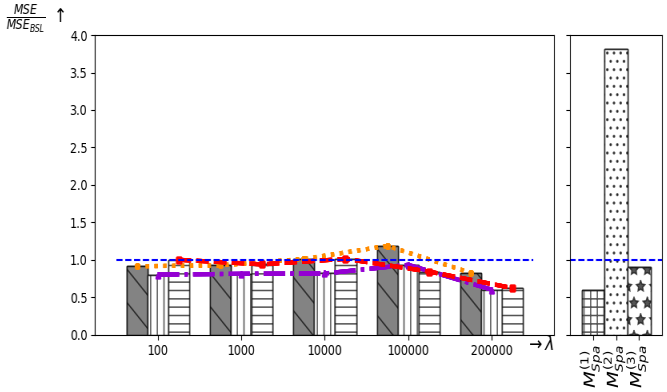


(b) Curved Backward-Facing Step.

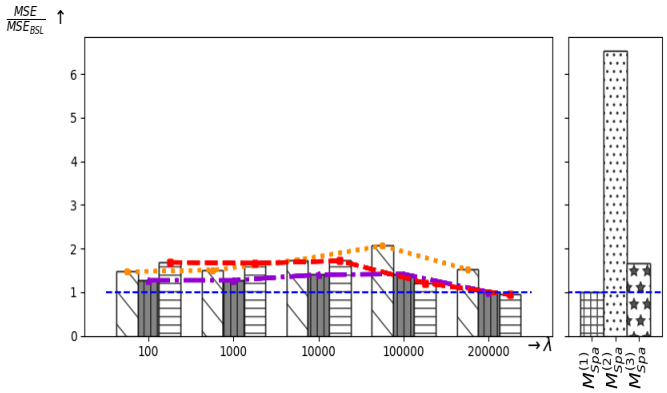


(c) Periodic Hills.

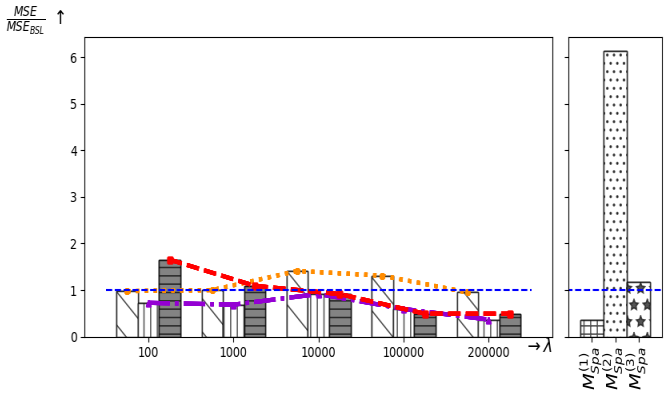
Figure 4.3: MSE of wall friction coefficient (relative to MSE of the baseline model $k-\omega$ SST (- - -)) as a function of λ. Left panel: $\mathbf{M}^{(1)}$ (▨, ·····), $\mathbf{M}^{(2)}$ (▥, -·-·-), $\mathbf{M}^{(3)}$ (▤, - - -); right panel: comparison with models from [61]: $\mathbf{M}^{(1)}_{Spa}$ (▦), $\mathbf{M}^{(2)}_{Spa}$ (⁝⁝), $\mathbf{M}^{(3)}_{Spa}$ (✶).



(a) Converging-diverging channel.

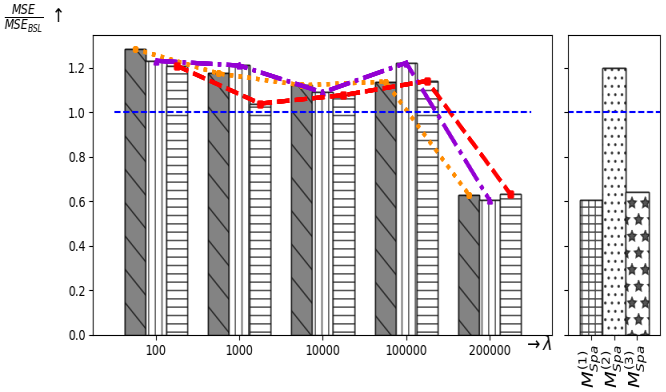


(b) Curved Backward-Facing Step.

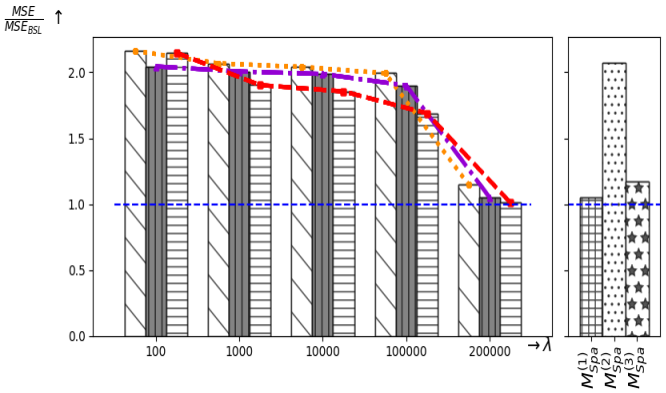


(c) Periodic Hills.

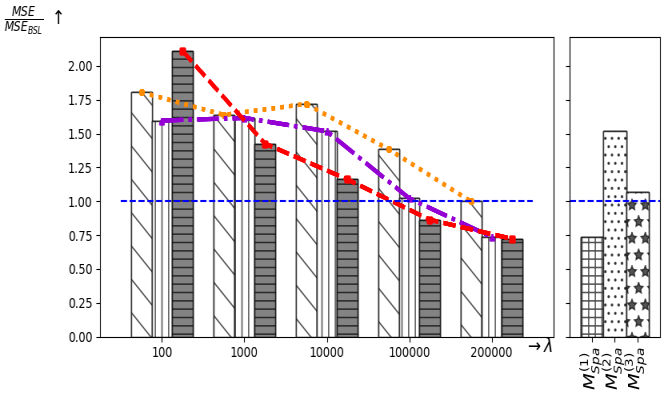
Figure 4.4: MSE of the turbulent kinetic energy (relative to MSE of the baseline model $k-\omega$ SST (- - -)) as a function of λ. Left panel: $\mathbf{M}^{(1)}$ (▨, ·····), $\mathbf{M}^{(2)}$ (▥, -·-·-), $\mathbf{M}^{(3)}$ (▤, - - -); right panel: comparison with models from [61]: $\mathbf{M}^{(1)}_{Spa}$ (▦), $\mathbf{M}^{(2)}_{Spa}$ (⁝⁝), $\mathbf{M}^{(3)}_{Spa}$ (★).



(a) Converging-diverging channel.



(b) Curved Backward-Facing Step.



(c) Periodic Hills.

Figure 4.5: MSE of Reynolds shear stress (relative to MSE of the baseline model $k-\omega$ SST (- - -)) as a function of λ. Left panel: $\mathbf{M}^{(1)}$ (▨, ·····), $\mathbf{M}^{(2)}$ (▥, -·-·-), $\mathbf{M}^{(3)}$ (▤, - - -); right panel: comparison with models from [61]: $\mathbf{M}^{(1)}_{Spa}$ (▦), $\mathbf{M}^{(2)}_{Spa}$ (⁝⁝), $\mathbf{M}^{(3)}_{Spa}$ (★).

4.3.2 Results

As a result of the preceding cross-comparison process, we select models representing the best compromise in terms of predicting the four precedent QoI, one for each of the three training scenarios. More precisely, models $\mathbf{M}^{(1)}(\lambda^{*(1)} = 2 \times 10^5)$, $\mathbf{M}^{(2)}(\lambda^{*(2)} = 2 \times 10^5)$ and $\mathbf{M}^{(3)}(\lambda^{*(3)} = 2 \times 10^5)$ are retained (see Appendix B for their mathematical expressions). In the following, we drop the dependency on λ to simplify model notations.

In the following, we focus on the selected "best" models and we look more closely to the posterior predictive distributions of selected output QoI. For that purpose, the posterior probability distributions of the stochastic parameters are propagated through the CFD solver by means of a non-intrusive sparse polynomial chaos method (see Appendix A). Since the posterior distributions of the coefficients are Gaussian by construction, Hermite polynomials are selected, and the expansion is truncated to second order. The selected models are very sparse, and governed by a single stochastic coefficient. With these settings, only three CFD simulations for each model are required to compute the statistical moments of the stochastic CFD predictions with satisfactory accuracy. Specifically, in the following discussion we focus on the statistical average and standard deviation of the predicted CFD solution.

In Figures 4.6, 4.7, and 4.8 respectively, we display selected profiles of the streamwise velocity profiles, friction coefficient distribution along the bottom wall and turbulent kinetic energy profiles, and for the three flow configurations at stake. The latter correspond to averages of the posterior predictive distributions for $\mathbf{M}^{(1)}$, $\mathbf{M}^{(2)}$ and $\mathbf{M}^{(3)}$. For model $\mathbf{M}^{(3)}$ (the best model in term of accuracy and generalization across the QoI and flow configurations - see Table 4.2–) as well as for $\mathbf{M}^{(1)}$ and $\mathbf{M}^{(2)}$, we report three-standard deviation confidence intervals. Baseline $k-\omega$ SST results and high-fidelity data are also included for reference. The three SBL models provide rather similar predictions of the velocity profiles (Figure 4.6) and clearly outperform the baseline model in matching the high-fidelity data. The improvement is more evident in the recirculation regions of the CBFS and PH flows, where the present models accurately predict the back flow. Uncertainty intervals for $\mathbf{M}^{(3)}$ velocity profiles are too narrow to capture everywhere the LES data, which are rather encompassed by those of $\mathbf{M}^{(1)}$, whose predictions for the

streamline velocity are closer to the LES data (see orange dotted curves in Figure 4.6 and MSE in Table 4.2). Interestingly, the solution confidence intervals are represented symmetrically with respect to the average solution ($\pm 3\sigma$). It is however likely that, despite the parameter posterior is symmetric, the solution posterior distribution is not, due to the non-linearity of the Navier–Stokes operator. Recovering the full posterior solution pdf would however require an increased computational effort. Of note, the uncertainty intervals correctly become larger in regions of higher discrepancy between the SBL model and the LES, thus warning the user about model reliability in such regions, which is the main goal of the proposed stochastic approach.

The predicted friction coefficient, shown in Figure 4.7, is also in nice agreement with high-fidelity data, providing more accurate estimates of the separation and reattachment points than the baseline. The confidence intervals encompass the separation and reattachment locations of the LES.

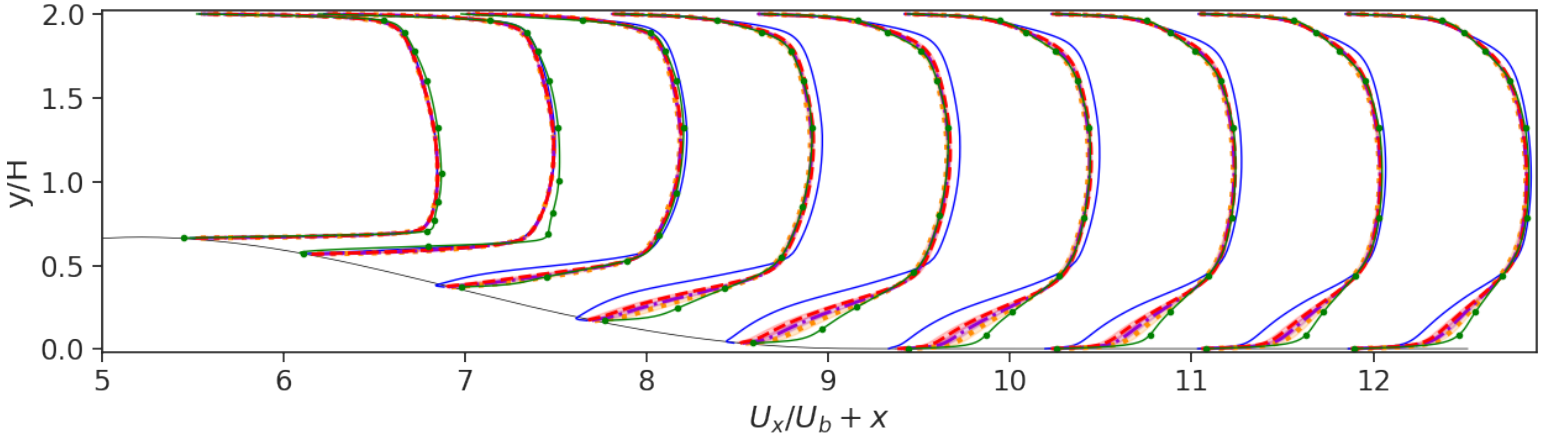
Figures 4.6 and 4.7 show that, despite general improvement over the baseline is observed for both velocity profiles and skin friction, none of the learned models is able to predict the exact location of the small separation bubble observed in LES results for the CD case, and confidence intervals are not large enough to capture the LES. The bubble is much smaller than in the $k-\omega$ SST solution, but it is shifted to the right and its size is still bigger than the LES. The velocity profiles are affected accordingly, although the solution is generally more accurate than the baseline. Our interpretation is the following. For all separated flows under investigation, the discovered models tend to reduce the large recirculation bubble produced by the baseline model by increasing eddy viscosity. This is beneficial in terms of representation of the separated region but, as a side effect, increases skin friction in attached flow regions significantly. The models then find a compromise solution between the opposite requirements of reducing the recirculation bubble without generating extra friction in attached regions. The CBFS and PH cases being massively separated, the error is effectively reduced by increasing turbulent viscosity and reducing the recirculation bubble. For CD, only marginal separation is present. Increasing eddy viscosity tends to increase skin friction upstream of the bubble, delaying separation.

While this kind of correction improves overall the solution over the baseline, it does not lead to a satisfactory solution everywhere in the flow field.

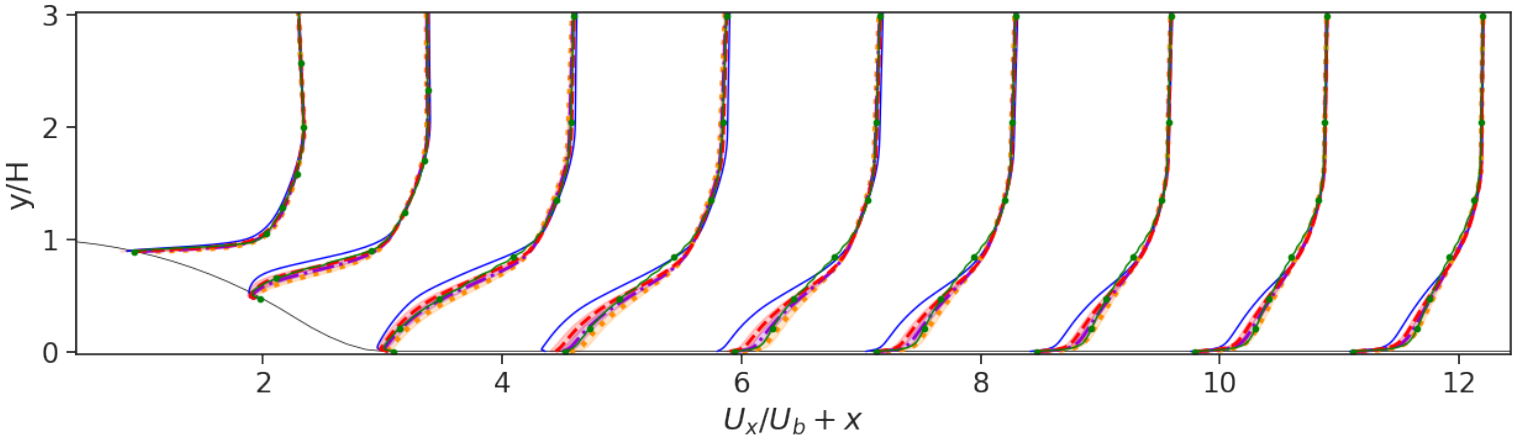
We also observe that, for the CBFS case, the predicted friction coefficient is slightly less accurate than the baseline in attached flow regions. This is most visible for $\mathbf{M}^{(1)}$, characterized by the highest mean value of the $\mathbf{b}^R$ correction among the selected models. This is again a consequence of the extra turbulent kinetic energy introduced to correct the separated region, which is transported in regions where the baseline performs well and does not need such correction. Interestingly, the high-fidelity values are still contained in the confidence intervals of our stochastic models.

The turbulent kinetic energy, presented in Figure 4.8, is a difficult quantity to be captured by RANS models. The three data-driven models perform overall slightly better than the baseline (as shown by the preceding analysis of MSE). However, they all overpredict k in the separated region, where an increased amount of k is generated to reduce the reattachment length. For this QoI, large confidence intervals are predicted for all cases.

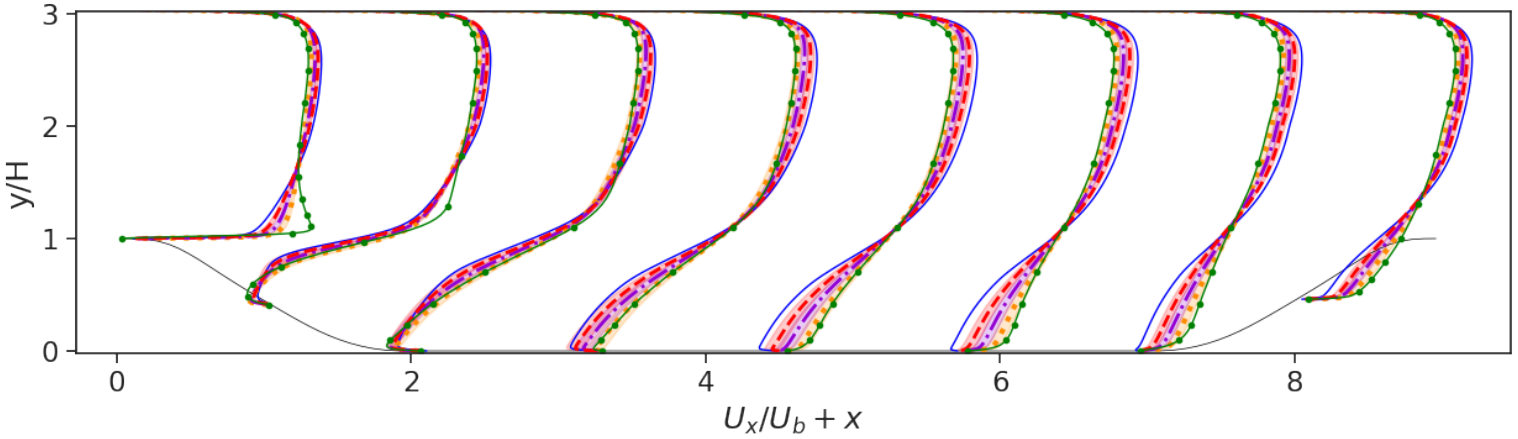
The reason for the above-mentioned defects of learned models is most likely Pope's representation of the Reynolds stresses. As mentioned in Section 2, the latter relies only on two invariants I_1 and I_2 and a time scale ω^{-1}. Including more features, as in [54, 58], or developing localized corrections that are selectively activated only where the baseline model exhibit high discrepancies [67] are promising options for further model improvement, which will be considered in future research. Despite such limitations, the present extremely simple models (with a single corrective term in the transport equations) are, for the class of flows at hand, almost as robust and cheap as the baseline $k-\omega$ SST but perform significanly better than, *e.g.* the physics-based RSM $k-\omega$ model of Ref. [23] available in the *OpenFoam* code, which involves nonlinear corrections to the Reynolds stresses with complex function coefficients. Comparisons between the physics-based EARSM and SBL models are reported in Appendix B.1.



(a) Converging-diverging channel.

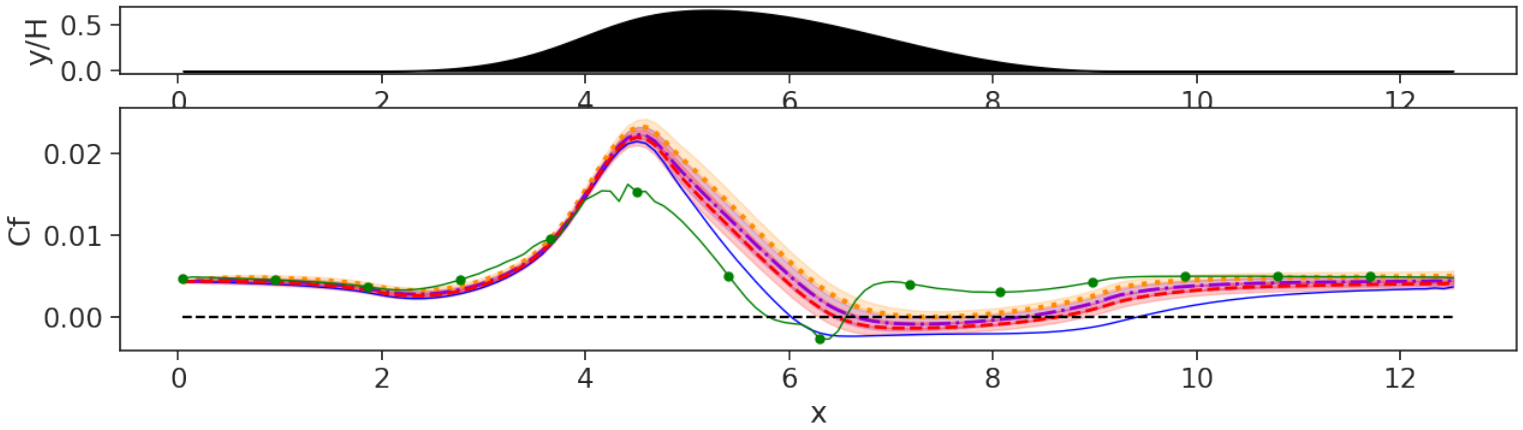


(b) Curved Backward-Facing Step.

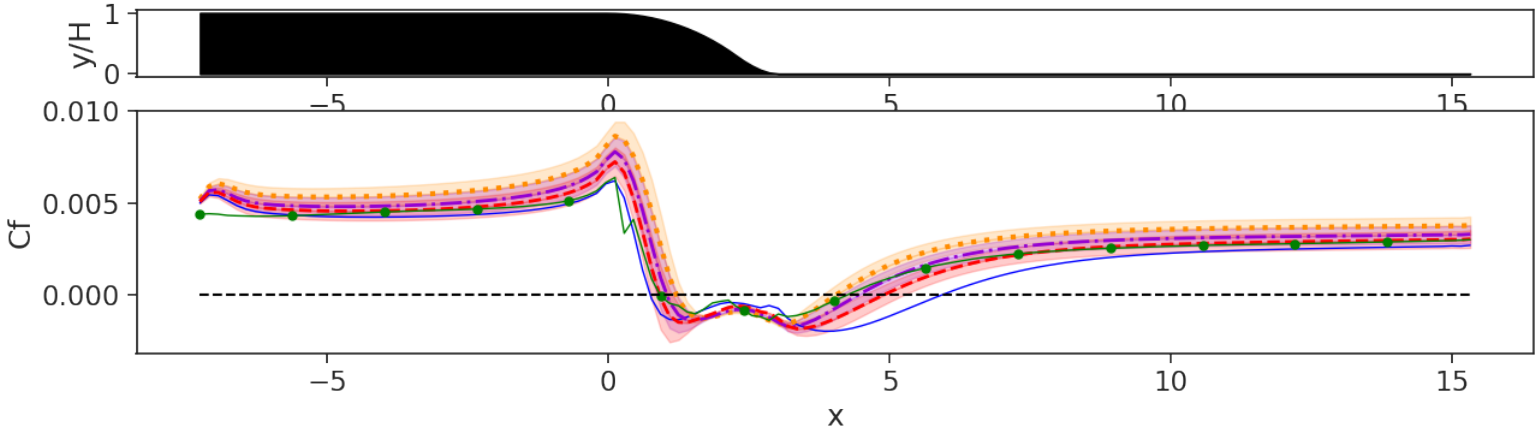


(c) Periodic Hills.

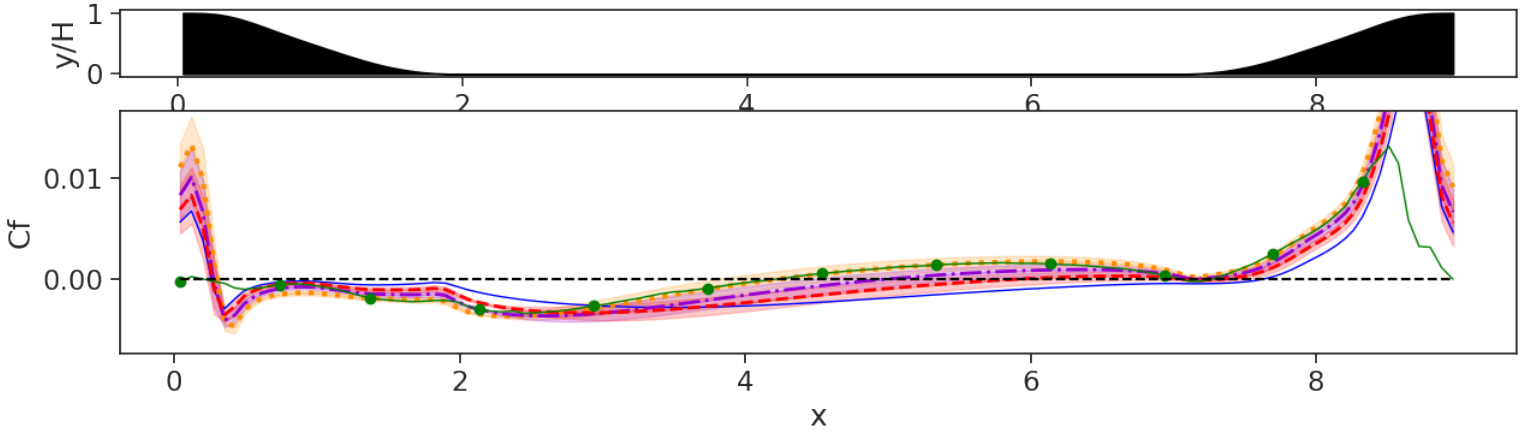
Figure 4.6: Streamwise velocity profiles: baseline $k-\omega$ SST (—), LES (—●—), and SBL models with $\pm$ 3 standard deviation confidence intervals: $\mathbf{M}^{(1)}$ (·····,), $\mathbf{M}^{(2)}$ (-·-·-,) and $\mathbf{M}^{(3)}$ (- - -,).



(a) Converging-diverging channel.

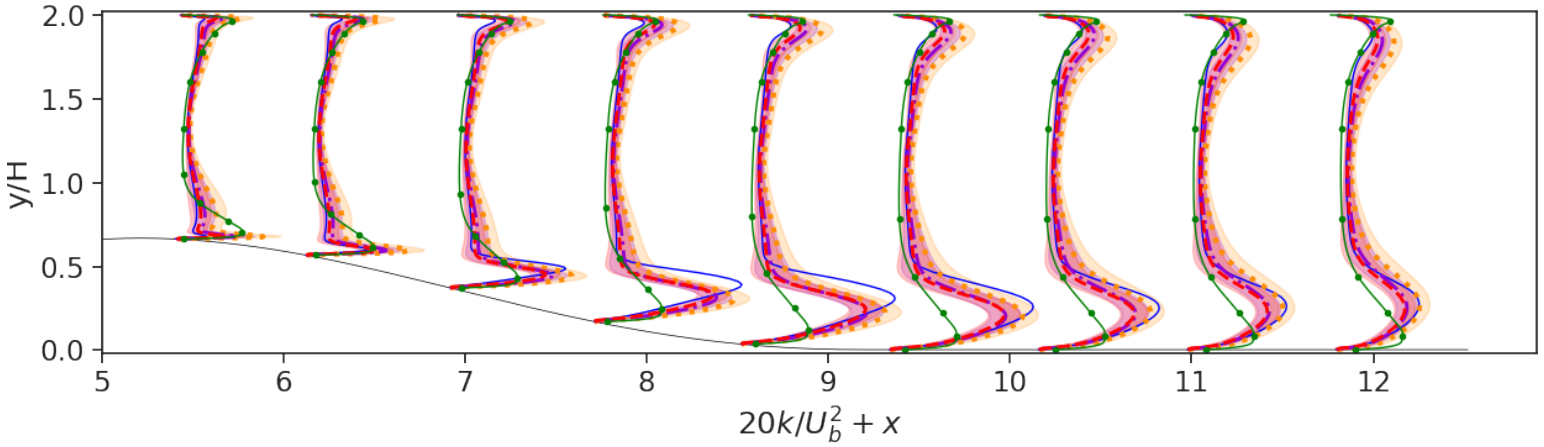


(b) Curved Backward-Facing Step.

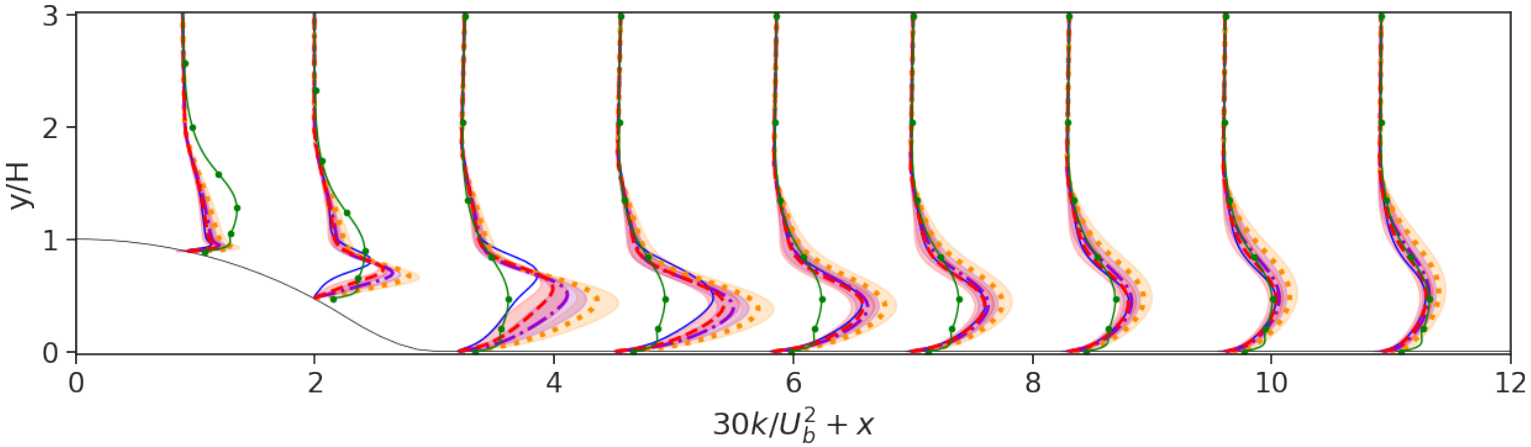


(c) Periodic Hills.

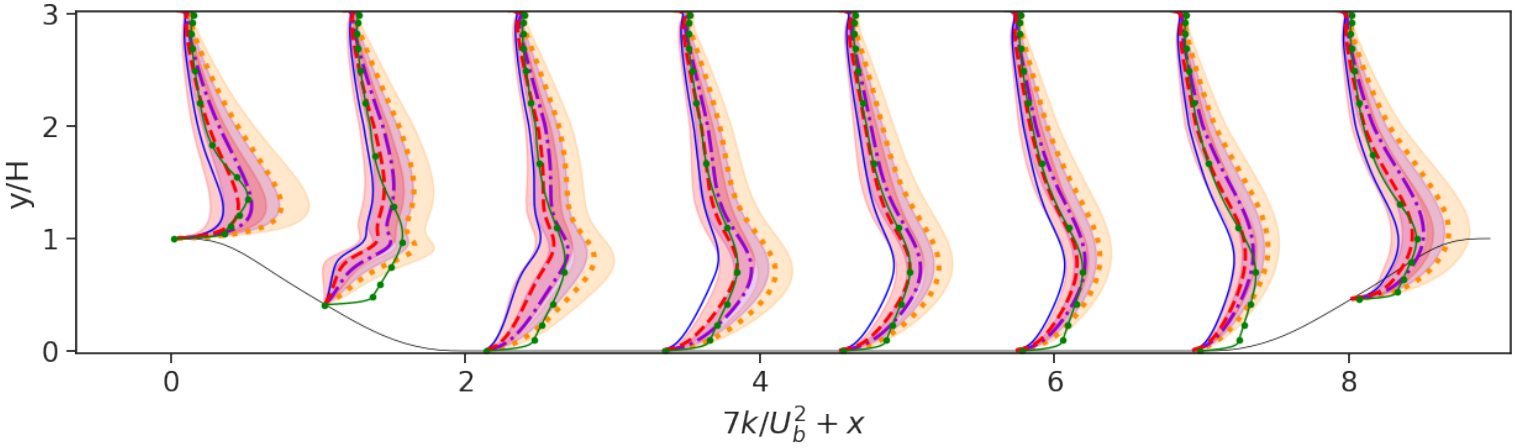
Figure 4.7: Friction coefficient distributions along the bottom wall: baseline $k-\omega$ SST (—), LES (—●—), and SBL models with $\pm$ 3 standard deviation confidence intervals: $\mathbf{M}^{(1)}$ (·····,), $\mathbf{M}^{(2)}$ (-·-·-,) and $\mathbf{M}^{(3)}$ (- - -,).



(a) Converging-diverging channel.



(b) Curved Backward-Facing Step.



(c) Periodic Hills.

Figure 4.8: Turbulent kinetic energy profiles: baseline $k-\omega$ SST (—), LES (—●—), and SBL models with $\pm$ 3 standard deviation confidence intervals: $\mathbf{M}^{(1)}$ (·····,), $\mathbf{M}^{(2)}$ (-·-·-,) and $\mathbf{M}^{(3)}$ (- - -,).

4.3.3 Sensitivity analysis

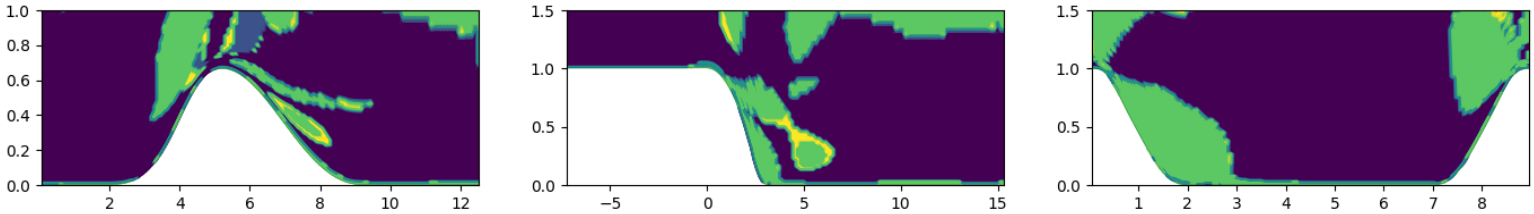
In the preceding section, a reduced-cost Uncertainty Quantification (UQ) method has been used for propagating the stochastic turbulence models through the CFD solver and use the converged solutions to compute the mean predictions and confidence intervals. Using the same UQ procedure (see Appendix A.0.1), a Sobol sensitivity analysis of the models to the stochastic parameters is conducted. To investigate the role of the various corrective tensor terms, results are reported hereafter for a more complex model than those selected in Section 4.2. Specifically, we consider the model providing the highest accuracy (77%) on the streamwise velocity, *i.e.* $\mathbf{M}^{(3)}(\lambda^{*(3)})$ for U (see Table 4.2), which involves a non-null $\mathbf{b}^{\Delta}$ correction.

We denote S_1 (resp. S_2 and S_3) the Sobol index of a QoI with respect to the first (resp. second and third) stochastic parameter contained in $\mathbf{b}^{\Delta}$, and S_4 the Sobol index associated with the only stochastic parameter of $\mathbf{b}^{R}$. Similarly, we denote S_{ij} the second-order Sobol indexes, representing interactions between parameters taken two by two. Parameters of even higher order were computed, but we found their contributions negligible. Areas of high sensitivity are identified as areas where the corresponding Sobol index is higher than 0.5; since Sobol indexes sum up to one, the remaining indices are then less than 0.5 in these regions.

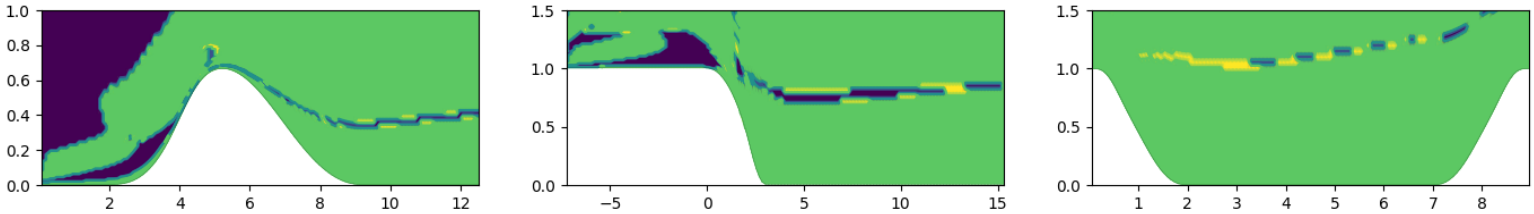
In Figure 4.9a we display as an example the map of dominance of Sobol sensitivity indices corresponding to principal effects (S_1, S_2, S_3 and S_4) and to interactions ($\sum_{i<j} S_{ij}$) for the shear component of the anisotropy tensor correction a_{12}^{Δ}. The figure shows that Reynolds anisotropy is mostly affected by the parameter governing the $\mathbf{b}^{R}$ correction close to walls, and more particularly near the separation - reattachment points. Sensitivity to $\mathbf{b}^{\Delta}$ is mostly observed outside of the boundary layer, and specifically in the recirculation bubble, the most relevant term in $\mathbf{b}^{\Delta}$ being again the linear term, *i.e.* the term involving tensor $\mathbf{T}^{(1)}$. This shows that LEVM are overall sufficient for such 2D separated flows, provided that the eddy viscosity coefficient is properly tuned.

Figures 4.9b, 4.9c and 4.9d show the map of dominance of Sobol sensitivity indices corresponding to principal effects and interactions for the streamwise velocity, turbulent kinetic

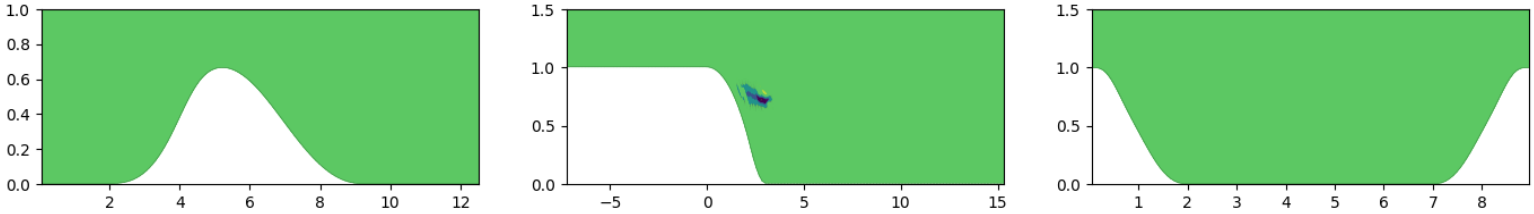
energy and Reynolds shear stress, respectively. All of the considered QoI are mostly sensitive to the $\mathbf{b}^R$ correction, especially in the near-wall region, whereas $\mathbf{b}^\Delta$ is activated in highly distorted regions, such as shear layers and recirculation bubbles. Within the latter, interaction terms also play an important role. The high accuracy of $\mathbf{M}^{(3)}(\lambda^{*(3)})$ in predicting streamwise velocity with respect to the sparsest models is then likely due to its ability to capture some of the anisotropy effects in the most highly distorted flow regions.



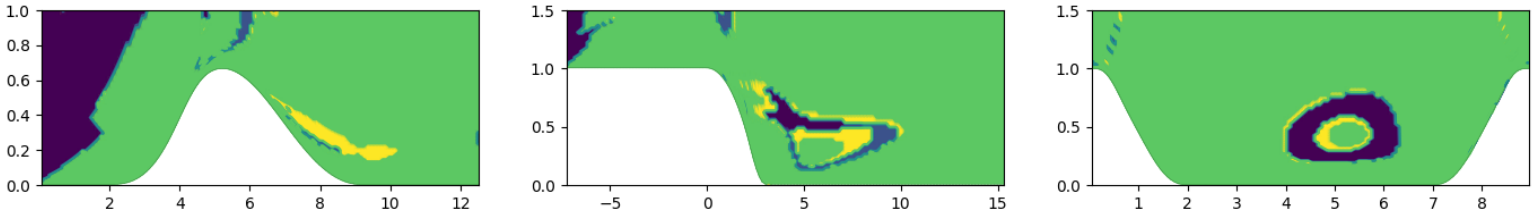
(a) a_{12}^{Δ}.



(b) Streamwise velocity U.



(c) Turbulent kinetic energy k.



(d) Reynolds shear stress τ_{xy}.

Figure 4.9: Maps of Sobol sensitivity indices dominance for various QoI: CD (left), CBFS (center) and PH (right) cases; S_1 (■), S_2 (■), S_3 (■), S_4 (■), $\sum_{i,j} S_{ij}$ (■).

4.4 Conclusions

In this chapter, a novel Sparse Bayesian Learning (SBL) framework was introduced for generating stochastic Explicit Algebraic Reynolds Stress Models (EARSM) for the Reynolds-Averaged Navier–Stokes equations, and demonstrated for a class of incompressible separated flows. The resulting models, called SBL-SpaRTA, exhibit a higher sparsity, physical interpretability, and a consistent form of the correction when tested on different geometries where turbulent separation occurs. The stochastic nature of the correction allows to quantify the uncertainty around different QoI by the mean of precious high-density intervals reflecting the trustworthiness of the solution. The SBL-SpaRTA models exhibit superior accuracy than the baseline LEVM in terms of velocity fields and skin friction, and the solutions generally in good agreement with the reference high-fidelity data. For some other quantities (such as the turbulent kinetic energy or the turbulent shear stress) the solution is not perfect, but overall more accurate than the LEVM. Interestingly, comparisons with a physics-based EARSM show that the present simple machine-learned models provide more accurate solutions for the considered class of flows, in the face of lower complexity. Model cross-validation and sensitivity analyses show that, for the present 2D separated flows, nonlinear corrections of the Reynolds stress tensor have little influence on the results in most of the flow, and that correcting the turbulent kinetic energy production term is generally sufficient for improving the match between the RANS model and the LES. This is consistent with the results of [61, 71], where similar models were discovered for the same class of flows using two different data-driven symbolic model identification strategies.

As a downside of the data-driven approach, we noticed that the present SBL-SpaRTA models result in a correction that is applied throughout the flow, affecting also regions where the baseline $k-\omega$ SST already yields good results. In addition, the SBL-SpaRTA correction for the turbulent separated flows is *a priori* not necessarily the optimal one for a general flow field where different physical phenomena take place. In the aim of moving towards more generalizable data-driven models, in the next chapter we investigate a machine-learning technique for aggregating

various data-driven models, each one specialized for a specific task (*i.e.* a class of flows).

Chapter 5

Non-intrusive space-dependent aggregation of SBL-SpaRTA models

Contents

In the preceding chapter, we developed customized stochastic SBL-SpaRTA corrections tailored for turbulent separated flows. The learned models were shown to improve the solution of the baseline $k-\omega$ SST model, particularly in its prediction of separation regions. Building upon this achievement, the focus of this chapter is to find a data-driven modeling framework that is able to encompass a broader range of flow cases. For that purpose, we first learn customized models for a diverse set of flows. Subsequently, we show that such models are not

generalizable outside the narrow class of flows for which they are trained. Finally, we propose a model aggregation technique to blend together the solutions of different customized models according to a set of local flow features. The local blending coefficients can be interpreted as model probabilities, and are used to estimate the average (expected) solution, as well as uncertainty intervals.

5.1 Customized SBL-SpaRTA models for building-block flows

5.1.1 Model training

The SBL framework is used to learn stochastic EARSM (SBL-SpaRTA) for a set of well-chosen flow cases, listed in Table 5.1. The training cases are representative of diverse physical situations, including turbulent plane channel flows, flat plates subjected to various pressure gradients, separated flows, and a jet flow. A similar strategY was independently proposed in [76]. For each of these flow cases, high-fidelity data corresponding to velocity $\boldsymbol{U}$, turbulent kinetic energy k, turbulent dissipation rate ω, and Reynolds stress tensor $\boldsymbol{\tau}$ are used to construct the target learning vectors and function dictionaries for the SBL-SpaRTA procedure. The training data are taken from evenly-distributed profiles across the computing domain. The training is performed using different values of the hyper-parameter λ within the set $\{1, 10, 10^2, 5 \times 10^2, 10^3\}$. The resulting models $\mathbf{M} = (\mathbf{M}_{\mathbf{b}^\Delta}, \mathbf{M}_{\mathbf{b}^R})$ (given in Table 5.2) take the general form:

$$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta} = \sum_{n=1}^{3} \left(\sum_{l,m} \left(\mu_{(l,m)}^{\Delta^{(n)}} \pm \sigma_{l,m}^{\Delta^{(n)}} \right) I_1^l I_2^m \right) \mathbf{T}^{(n)} \pm \mathbf{1}\epsilon^\Delta \\ \mathbf{M}_{\mathbf{b}^R} = \sum_{n=1}^{3} \left(\sum_{l,m} \left(\mu_{(l,m)}^{R^{(n)}} \pm \sigma_{l,m}^{R^{(n)}} \right) I_1^l I_2^m \right) \mathbf{T}^{(n)} \pm \mathbf{1}\epsilon^R \end{cases} \tag{5.1}$$

where $\mu_{(l,m)}^{\Delta^{(n)}}$ and $\sigma_{l,m}^{\Delta^{(n)}}$ (resp. $\mu_{(l,m)}^{R^{(n)}}$ and $\sigma_{l,m}^{R^{(n)}}$) are the mean and the standard deviation, respectively, of the probability density function of the coefficient associated to the term $I_1^l I_2^m$ in the tensor expansion of $\mathbf{M}_{\mathbf{b}^\Delta}$ (resp. $\mathbf{M}_{\mathbf{b}^R}$), $\mathbf{1}$ is a second-order tensor with all elements equal to one, and ϵ^Δ (resp. ϵ^R) is the standard deviation of the noise associated with model $\mathbf{M}_{\mathbf{b}^\Delta}$ (resp. $\mathbf{M}_{\mathbf{b}^R}$).

Training cases	Description	Source
ZPG	DNS of a zero pressure gradient turbulent boundary layer $670 \leq Re_\theta \leq 4060$	[111]
CHAN	DNS of turbulent channel flows $180 \leq Re_\tau \leq 5000$	[90] [112]
APG	LES of adverse pressure-gradient TBL $Re_\theta \leq 4000$, $\beta \leq 4$, 5 different pressure gradients	[94]
ANSJ	PIV of near sonic axisymmetric jet	[95]
SEP	LES of Periodic Hills (PH) at $Re = 10595$ DNS of converging-diverging channel (CD) at $Re = 13600$ LES of curved backward facing step (CBFS) at $Re = 13700$	[99] [97] [98]

Table 5.1: List of flow cases used to train customized SBL-SpaRTA corrections.

Training case	Model
ZPG	$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(ZPG)} = & [(0.152 \pm 0.0430)(I_1 - I_2)]\mathbf{T}^{(1)} + \pm 0.167 \\ \mathbf{M}_{\mathbf{b}^R}^{(ZPG)} = & [0] \pm 3.01 \times 10^{-3} \end{cases}$
CHAN	$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(CHAN)} = & [0] + \pm 0.0914 \\ \mathbf{M}_{\mathbf{b}^R}^{(CHAN)} = & [0] \pm 4.61 \times 10^{-3} \end{cases}$
APG	$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(APG)} = & [(2.99 \pm 0.00726)]\mathbf{T}^{(2)} + \pm 0.000277 \\ \mathbf{M}_{\mathbf{b}^R}^{(APG)} = & [0] \pm 6.55 \times 10^{-5} \end{cases}$
ANSJ	$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(ANSJ)} = & [(0.33 \pm 0.0189)]\mathbf{T}^{(1)} + \pm 0.00622 \\ \mathbf{M}_{\mathbf{b}^R}^{(ANSJ)} = & [0] \pm 3.45 \times 10^{-3} \end{cases}$
SEP	$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(SEP)} = & [(5.21 \pm 0.0173)]\mathbf{T}^{(2)} + \pm 0.0348 \\ \mathbf{M}_{\mathbf{b}^R}^{(SEP)} = & [(0.681 \pm 0.02)]\mathbf{T}^{(1)} \pm 0.0318 \end{cases}$

Table 5.2: Customized SBL-SpaRTA corrections obtained for various training flow cases.

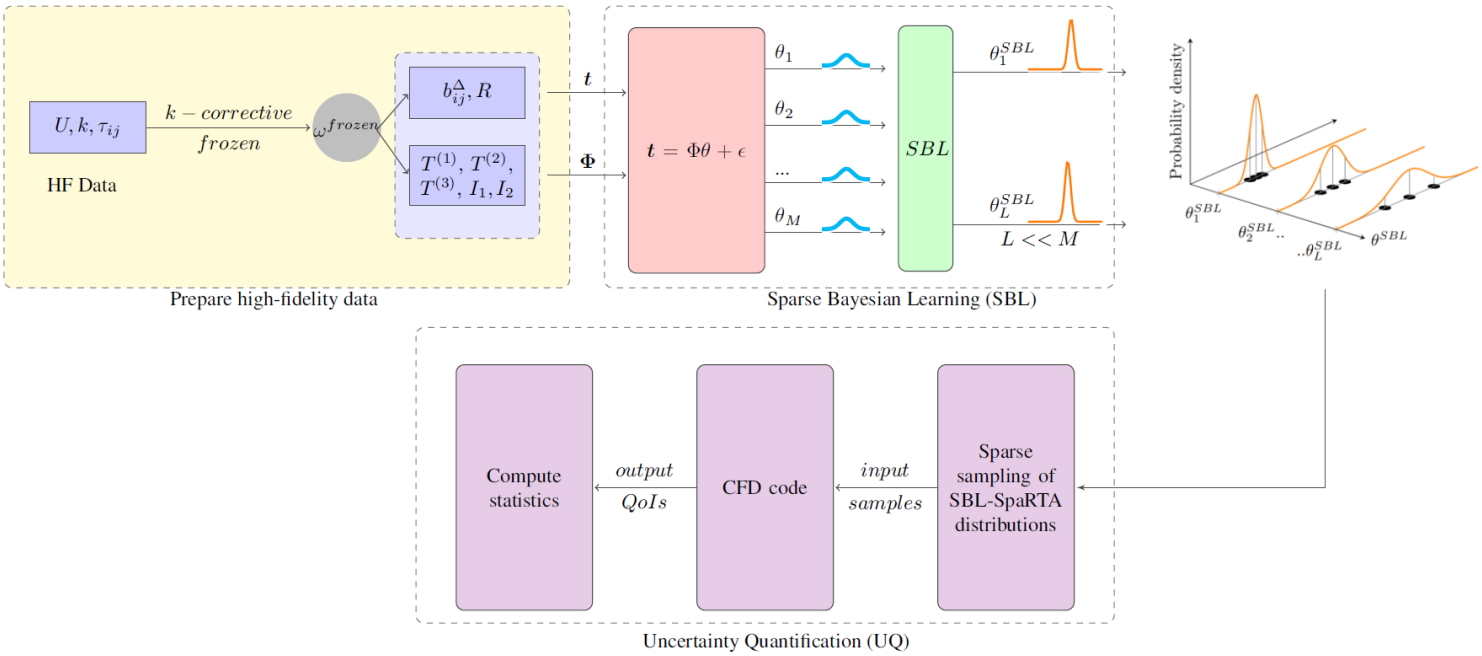


Figure 5.1: SBL-SpaRTA framework.

For each training case, a cross-validation strategy is used to select the sparsity-promoting hyperparameter λ. To avoid overcharging the training data, all discovered models are propagated through the CFD solver, and the models providing the smallest predictive error on the horizontal velocity U are retained as the best ones (see [101] for further discussion of the cross-validation step). The model corrections discovered for the various cases after the training and cross-validation steps are presented in Table 5.2.

5.1.2 Stochastic flow predictions

The posterior probability distributions of the selected model parameters, along with the associated tensor terms, constitute a stochastic turbulence model correction. The latter can be propagated through the flow solver by using a suitable Uncertainty Quantification (UQ) algorithm, to obtain a stochastic prediction of a new flow. In the following numerical studies, we use again the *equadratures* library of [113] already used in the preceding chapter and described in Appendix A. The main steps of the SBL-SpaRTA framework with the Uncertainty Quantification (UQ) procedure are presented in Figure 5.1.

5.1.3 Customized model performance

A clear advantage of symbolic identification Machine Learning algorithms, such as SBL-SpaRTA, is that they provide tangible analytical expressions that allow the effects of the discovered model corrections to be interpreted. For the sake of simplicity, we discuss below the effects of the models of Table 5.2 by assuming that the stochastic model parameters are fixed to their mean values. Since model posteriors are generally well informed, *i.e.* sharply peaked around their mean, this is sufficient to understand the model main trend (the solution computed at the mean value being a first-order approximation of the mean of the stochastic solution). To better illustrate the behavior of the discovered models, we also report in Figures 5.2, 5.3 and 5.4 the solutions of the various learned models for selected cases among those of Table 5.1. This allows also to test the generalizability of a model learned for a given data set to a different class of flows. The following considerations are in order:

- The optimal correction for the channel flow is zero to within observation noise, confirming that the baseline model is well fit for this canonical flow, at least as long as the main goal is to provide a good prediction of the velocity field. As a consequence, in the following discussions, the CHAN model coincides with the baseline $k-\omega$ SST model.

- For the zero pressure gradient boundary layer (ZPG case), a small anisotropy correction dependent on the linear term $\mathbf{T}^{(1)} = \mathbf{S}/\omega$ is introduced. By taking the mean values of the coefficients, we see that the correction adds to the Boussinesq term and leads to the corrected anisotropy relation:

$$2k\mathbf{b} = \left[-2\nu_t + 0.152(I_1 - I_2)\frac{k}{\omega}\right]\mathbf{S} = -\frac{2k}{\omega}\alpha_{ZPG}\mathbf{S}, \quad \alpha_{ZPG} \approx 1 - 0.076(I_1 - I_2) \quad (5.2)$$

  The coefficient α_{ZPG} corresponds to a very small correction (decrease) of eddy viscosity ν_t in the external region, as shown in Figure 5.2b for the CHAN flow case. The slope of the velocity profiles in log layer remain essentially unchanged with respect to the baseline (Figure 5.2a). Note that, for incompressible boundary layer flows, $I_1 \approx -I_2 \approx \frac{1}{2\omega^2}\left(\frac{\partial U}{\partial y}\right)^2$, meaning that the correction is active when the shear timescale becomes

smaller than the turbulent timescale. On the other hand, no additional corrective term is added to the turbulent kinetic energy equation on average, except for the small negative contribution of $\mathbf{b}^{\Delta}$ to the production term. This indicates that the discovered optimal model is very close to the baseline also for the ZPG boundary layer case.

- For adverse pressure gradient boundary layers (APG cases), a nonlinear anisotropy correction is discovered, resulting in a constitutive relation of the form:

$$2k\mathbf{b} = -2\nu_t\mathbf{S} + 2.99\frac{2k}{\omega^2}\left[\mathbf{S\Omega} - \mathbf{\Omega S}\right] \tag{5.3}$$

  Here again no additional corrective term is selected for the turbulent kinetic energy equation, and the contribution of $\mathbf{b}^{\Delta}$ to the TKE production is zero because the tensor product of $\mathbf{T}^{(2)}$ with the velocity gradient tensor is null. This can be seen in Figure 5.2b, where the profile of eddy viscosity across the CHAN flow predicted by the APG model is superimposed with those of the baseline (or CHAN) models. The correction also has essentially no effect on the velocity profile for both the CHAN case shown in Figure 5.2a and the ANSJ case (Figure 5.3a), and it provides skin friction profiles of separated cases such as the CD and PH flows in close agreement with the CHAN model (see Figure 5.4), showing that the learned correction for APG plays a very minor role for a large variety of cases.

- The turbulent near-sonic jet (ANSJ) model consists of a linear anisotropy correction, resulting in a decreased eddy viscosity compared to the baseline:

$$2k\mathbf{b} = \left[-2\nu_t + 0.33\frac{2k}{\omega}\right]\mathbf{S} = -\frac{2k}{\omega}\alpha_{ANSJ}\mathbf{S}, \quad \alpha_{ANSJ} \approx 0.67 \tag{5.4}$$

  Such a modification contributes to improving the spreading rate for the training flow, as shown in Figure 5.3a, where the horizontal velocity along x-axis is reported for all discovered models, along with the high-fidelity data. The computed eddy viscosity for the ANSJ case is shown in Figure 5.3b for all models. As a counterpart, the ANSJ model leads to a severe underestimation of the eddy viscosity in the CHAN case, and in particular in the logarithmic zone, resulting in an erroneous slope of the log-law, as shown

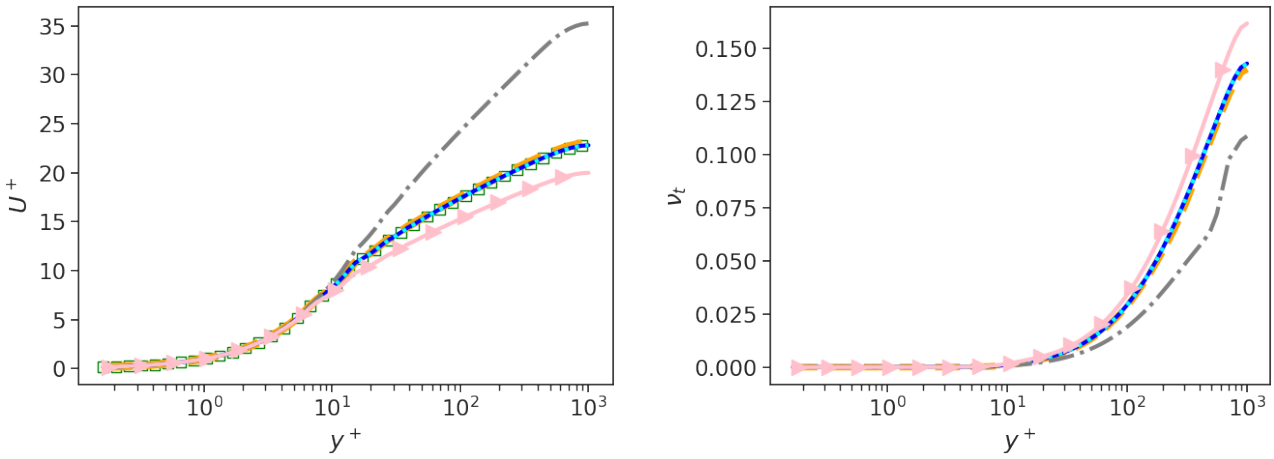
in Figures 5.2a and 5.2b. This also results in an underestimated skin friction and in a too large separation bubble for both CD and PH flows, reported in Figures 5.4a and 5.4b.

- Finally, the separated flow (SEP) model involves both a nonlinear correction to the extra anisotropy and a linear $\mathbf{b}^R$ correction.

$$\begin{cases} 2k\mathbf{b} & = -2\nu_t\mathbf{S} + 5.21\frac{2k}{\omega^2}\left[\mathbf{S\Omega} - \mathbf{\Omega S}\right] \\ 2k\mathbf{b}^R & = -\frac{k}{\omega}\alpha_{SEP}\mathbf{S}, \quad \alpha_{SEP} \approx 1.362 \end{cases} \tag{5.5}$$

  While the non-linear correction of $\mathbf{b}^\Delta$ does not affect the TKE production, the $\mathbf{b}^R$ correction tends to increase the eddy viscosity. Inspection of the skin friction distribution for the CD and PH flows (Figures 5.4a and 5.4b) shows that the SEP model significantly improves the agreement with the high-fidelity data in terms of size and position of the recirculation bubble in the PH case (Figure 5.4b) compared to all other models, and it also results in a more satisfactory overall agreement for the CD case, but it misses the small separation bubble in divergent (Figure 5.4a). The reason is that the model overcorrects the baseline, resulting for instance in an overdissipation for the CHAN case (see Figure 5.2b) and in an underestimation of the log-law slope (Figure 5.2a). Furthermore, the SEP model provides inaccurate predictions of the velocity distribution for the ANSJ case (Figure 5.3a).

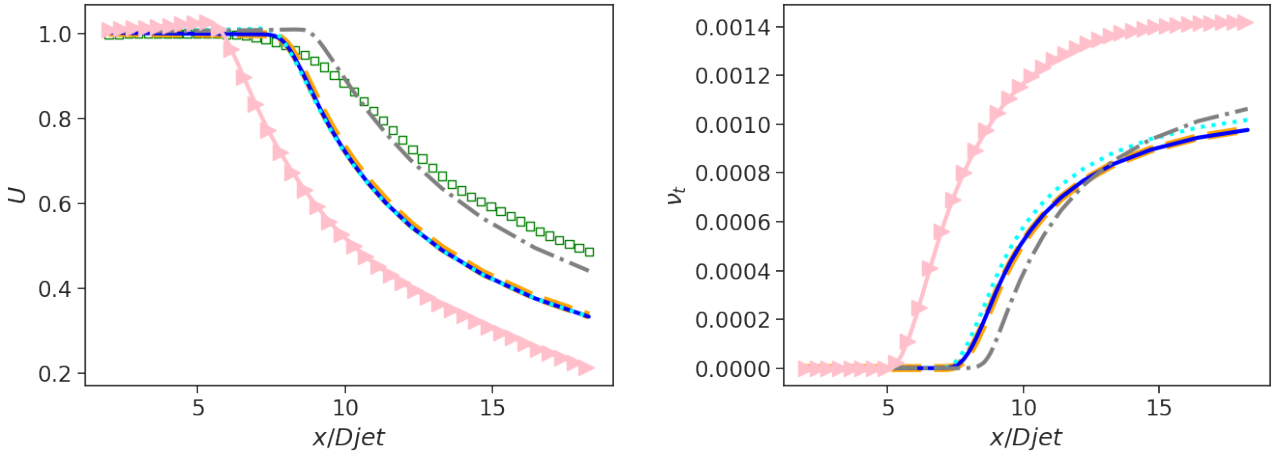
The preceding discussion shows that the SBL-SpaRTA models trained on a class of flows are generally not well-suited for predicting different flow configurations, *i.e.*, they are not generalizable outside the class of flows for which they have been developed. This is why the discovered models are qualified of "customized" models and do not constitute an universal general-purpose model. In an attempt of constructing a more generalizable model that combines the properties of the various discovered corrections, in the next section we discuss a model mixture procedure that combines the results of various customized models to make predictions of unseen flow configurations.



(a) Horizontal velocity profiles in wall units.

(b) Eddy viscosity profiles.

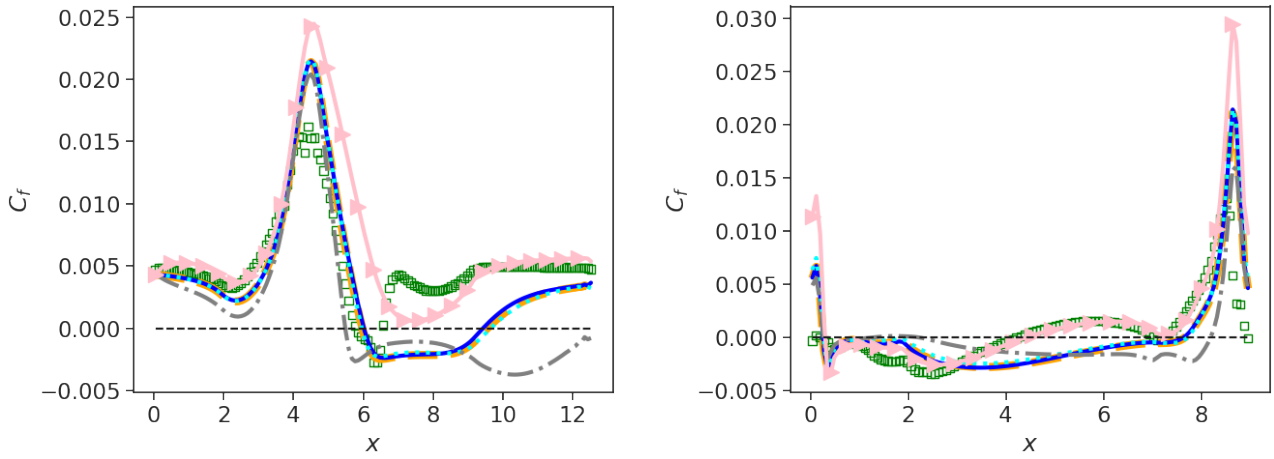
Figure 5.2: Incompressible turbulent channel flow at $Re_\tau = 1000$ - various SBL-SpaRTA models. $\mathbf{M}^{(ZPG)}$ (- - -), $\mathbf{M}^{(CHAN)}$ (——), $\mathbf{M}^{(APG)}$ (·····), $\mathbf{M}^{(ANSJ)}$ (-·-·-), $\mathbf{M}^{(SEP)}$ (—▸), High-fidelity data (□).



(a) Distribution of the horizontal velocity along the jet axis.

(b) Distribution of the eddy viscosity.

Figure 5.3: Axisymmetric near-sonic jet flow - various SBL-SpaRTA models. $\mathbf{M}^{(ZPG)}$ (- - -), $\mathbf{M}^{(CHAN)}$ (——), $\mathbf{M}^{(APG)}$ (·····), $\mathbf{M}^{(ANSJ)}$ (-·-·-), $\mathbf{M}^{(SEP)}$ (—▸), High-fidelity data (□).



(a) Converging diverging channel flow.

(b) Periodic hills flow.

Figure 5.4: Skin-friction distributions along the bottom wall - various SBL-SpaRTA models. $\mathbf{M}^{(ZPG)}$ (- - -), $\mathbf{M}^{(CHAN)}$ (——), $\mathbf{M}^{(APG)}$ (·····), $\mathbf{M}^{(ANSJ)}$ (-·-·), $\mathbf{M}^{(SEP)}$ (—➤), High-fidelity data (□).

5.2 Space-dependent Model Aggregation

The customized SBL-SpaRTA models improve the solution of the baseline model for the class of flows for which they were trained, but generally have poor performance for other classes of flows. In addition, it may be difficult to know a priori which model will perform better for an unseen flow that shares common characteristics with different flow classes.

With the goal of providing improved and more generalizable flow predictions while estimating turbulence modeling uncertainties, we introduce a model-aggregation procedure, referred to as X-MA, in which the predictions of multiple models are combined using weighting functions that can vary across the computational domain. Such functions are trained to automatically assign high weights to models that are likely to perform better in a given flow regime or flow region, and low weights to models that are likely to perform poorly. More specifically, in the following we build on the space-dependent model aggregation approach originally proposed by [86] for combining the predictions of a set of LEVMs from the literature, and we develop a methodology that locally combines the solutions of a set of competing SBL-SpaRTA models, learned for different flow environments, to predict unseen flows.

5.2.1 X-MA methodology

Let us consider K SBL-SpaRTA models, learned in different environments, and let $d(\mathbf{x})$ be any QoI predictable as an output of a RANS flow solver at some spatial location $\mathbf{x}$ (*e.g.* the predicted velocity or pressure fields, the skin friction distribution, etc.). In order to make predictions of d that are robust to the choice of the data-driven turbulence model for an unseen flow scenario, we borrow the "Mixture-of-Experts" concept [84] and we build an ensemble solution by aggregating the individual solutions d_k of the K SBL-SpaRTA models:

$$d_{\text{X-MA}}(\mathbf{x}) = \sum_{k=1}^{K} w_k(\mathbf{x}) d_k(\mathbf{x}) \tag{5.6}$$

In the above, $w_k(\mathbf{x})$ is the weighting function assigned to the k^{th} component model, and $d_{\text{X-MA}}(\mathbf{x})$ is the model ensemble or aggregated prediction. Following the approach of [86], we look for weighting functions satisfying the conditions:

$$\begin{cases} 0 \leq w_k(\mathbf{x}) \leq 1 \quad \forall k = 1, ..., K \\ \sum_{k=1}^{K} w_k(\mathbf{x}) = 1 \quad \forall \mathbf{x} \end{cases} \tag{5.7}$$

Given the preceding properties, they can be interpreted as the probability of model k to contribute to the aggregated prediction $d_{\text{X-MA}}(\mathbf{x})$ at location $\mathbf{x}$.

The weighting functions are computed as the Exponentially Weighted Average (EWA) of the component model prediction errors:

$$w_k\Big(\delta^{(k)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w\Big) = \frac{g_k\Big(\delta^{(k)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w\Big)}{\sum_{l=1}^{K} g_l\Big(\delta^{(l)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w\Big)} \tag{5.8}$$

where g_k is a gating function of the form

$$g_k\Big(\delta^{(k)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w\Big) = \exp\left(- \frac{\Big(\delta^{(k)}(\mathbf{x}) - \bar{\delta}(\mathbf{x})\Big)^T . \Big(\delta^{(k)}(\mathbf{x}) - \bar{\delta}(\mathbf{x})\Big)}{\sqrt{Var(\bar{\delta})} \quad \times \quad 2\sigma_w^2} \right) \tag{5.9}$$

with

- $\bar{\delta}$ is a vector of high-fidelity data (δ may, or may not, be equal to d),
- $\delta^{(k)}$ is the k^{th} model's output for $\bar{\delta}$,
- σ_w is a hyperparameter.

This gating function corresponds to an exponential transformation of the mean square error of the k^{th} model prediction d_k with respect to the high-fidelity data or, put in other terms, to the "score" assigned to the k^{th} model. The term $\sqrt{Var(\bar{\delta})}$ is introduced as a normalizing constant to ensure that the ratio inside the exponential is made non-dimensional and, consequently,

independent of the choice of the Quantity of Interest (QoI) used for weights' construction. This term is equal to the standard deviation among the training data points. For the rest of the study, we choose δ to be the horizontal velocity U.

The gating function depends on an hyperparameter σ_w, whose role is to discriminate more or less sharply the component model scores: when σ_w is large, all models tend to be assigned approximately equal weights, *i.e.* uncertainty on model choice is high, whereas when σ_w tends to zero, a single model is selected, which leads to better results if the right model is selected, but may lead to large errors if the wrong model is applied. The choice of σ_w is determined by means of a grid search whose details are given in Section 5.2.2.

From Equation (5.9), it appears that the gating functions are a way of comparing each SBL-SpaRTA based model with a set of high-fidelity values. Consequently, they can only be calculated at the locations $\mathbf{x}_i$ in the dataset. For making the gating functions, and subsequently the weights of each SBL-SpaRTA model, to give a value for each point $\mathbf{x}$ in the domain, a surface response must be constructed such that:

$$\mathbf{x} \rightarrow w_k(\mathbf{x})$$

by means of a regression algorithm. Nonetheless, in such circumstances, the regression procedure would be restricted to the training domain and so would be the blending approach.

To expand the blending approach predictions to any new domain, the regression is based on a set of features $\boldsymbol{\eta}(\mathbf{x})$ that represent the local flow instead:

$$\boldsymbol{\eta}(\mathbf{x}) \rightarrow w_k(\mathbf{x})$$

More precisely, we choose some of the features introduced by [52] and summarised in Table 5.3.

Feature	Description	Formula	Feature	Description	Formula
η_1	Normalized Q criterion	$\frac{\|\|\mathbf{\Omega}\|\|^2 - \|\|\mathbf{S}\|\|^2}{\|\|\mathbf{\Omega}\|\|^2 + \|\|\mathbf{S}\|\|^2}$	η_6	Viscosity ratio	$\frac{\nu_T}{100\nu + \nu_T}$
η_2	Turbulence intensity	$\frac{k}{0.5U_iU_i + k}$	η_7	Ratio of pressure normal stresses to normal shear stresses	$\frac{\sqrt{\frac{\partial P}{\partial x_i}\frac{\partial P}{\partial x_i}}}{\sqrt{\frac{\partial P}{\partial x_j}\frac{\partial P}{\partial x_j}} + 0.5\rho\frac{\partial U_k^2}{\partial x_k}}$
η_3	Turbulent Reynolds number	$\min\left(\frac{\sqrt{k}\lambda}{50\nu}, 2\right)$	η_8	Non-orthogonality marker between velocity and its gradient [29]	$\frac{\left\|U_kU_l\frac{\partial U_k}{\partial x_l}\right\|}{\sqrt{U_nU_nU_i\frac{\partial U_i}{\partial x_j}U_m\frac{\partial U_m}{\partial x_j}} + \left\|U_iU_j\frac{\partial U_i}{\partial x_j}\right\|}$
η_4	Pressure gradient along streamline	$\frac{U_k\frac{\partial P}{\partial x_k}}{\sqrt{\frac{\partial P}{\partial x_j}\frac{\partial P}{\partial x_j}U_iU_i} + \left\|U_l\frac{\partial P}{\partial x_l}\right\|}$	η_9	Ratio of convection to production of k	$\frac{U_i\frac{\partial k}{\partial x_i}}{\|\overline{u_j'u_l'}S_{jl}\| + U_l\frac{\partial k}{\partial x_l}}$
η_5	Ratio of turbulent timescale to mean strain time scale	$\frac{\|\|\mathbf{S}\|\|k}{\|\|\mathbf{S}\|\|k + \varepsilon}$	η_{10}	Ratio of total Reynolds stresses to normal Reynolds stresses	$\frac{\|\|\overline{u_i'u_j'}\|\|}{k + \|\|\overline{u_i'u_j'}\|\|}$

Table 5.3: List of input features used to construct the X-MA weighting functions.

The selected features are based on domain knowledge, and they include variables such as the Q criterion for vortical flow detection, the turbulent kinetic energy, and the turbulent dissipation rate among others. Additionally, we consider an extra feature $\eta_{11} = \frac{P_k}{P_k + \epsilon}$, suggested by [114]. The latter allows to integrate the information turbulent regimes for which the baseline model (identical to the CHAN model) provides reliable solutions. This is the case of freely decaying turbulence, corresponding to $P_k \to 0$, *i.e.* $\eta_{11} \to 0$ or of equilibrium turbulence, where $P_k \to \epsilon$, which implies $\eta_{11} \to \frac{1}{2}$.

The regression is computed by using the Random Forests (RF) algorithm available through the *python* package *scikitlearn*[†]:

$$\underbrace{\boldsymbol{\eta}(\mathbf{x}) = \Big(\eta_1(\mathbf{x}), ..., \eta_{11}(\mathbf{x})\Big)}_{\text{local flow features}} \xrightarrow[\mathcal{W}]{RF} \underbrace{\Big(w_1\Big(\delta^{(1)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w\Big), ..., w_K\Big(\delta^{(K)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w\Big)\Big)}_{\text{local models weights}} \tag{5.10}$$

The data used to train the RF are the horizontal velocity fields of the flow cases CHAN, APG, ANSJ, and SEP. The local flow features are calculated by using the baseline k-ω SST model. Of note, neither the velocity fields nor the flow features are used for learning the SBL-SpaRTA

[†]`https://scikit-learn.org`

corrections.

Ultimately, the model aggregation now reads

$$d_{\text{X-MA}}(\mathbf{x}) = \sum_{k=1}^{K} w_k(\boldsymbol{\eta}(\mathbf{x}))d_k(\mathbf{x}). \tag{5.11}$$

from which we compute the mean and variance of $d_{\text{X-MA}}(\mathbf{x})$

$$\begin{cases} \mathbb{E}\Big(d_{\text{X-MA}}(\mathbf{x})\Big) = \sum_{k=1}^{K} w_k(\boldsymbol{\eta}(\mathbf{x}))\mathbb{E}\Big(d_k(\mathbf{x})\Big) \\ Var\Big(d_{\text{X-MA}}(\mathbf{x})\Big) = \sum_{k=1}^{K} w_k^2(\boldsymbol{\eta}(\mathbf{x}))Var\Big(d_k(\mathbf{x})\Big) \end{cases} \tag{5.12}$$

where the models are assumed to be independent.

The workflow of the regression training is depicted in Figure 5.5 and an overview of the X-MA approach is plotted in Figure 5.6. The complete learning process is detailed in Section 5.2.2

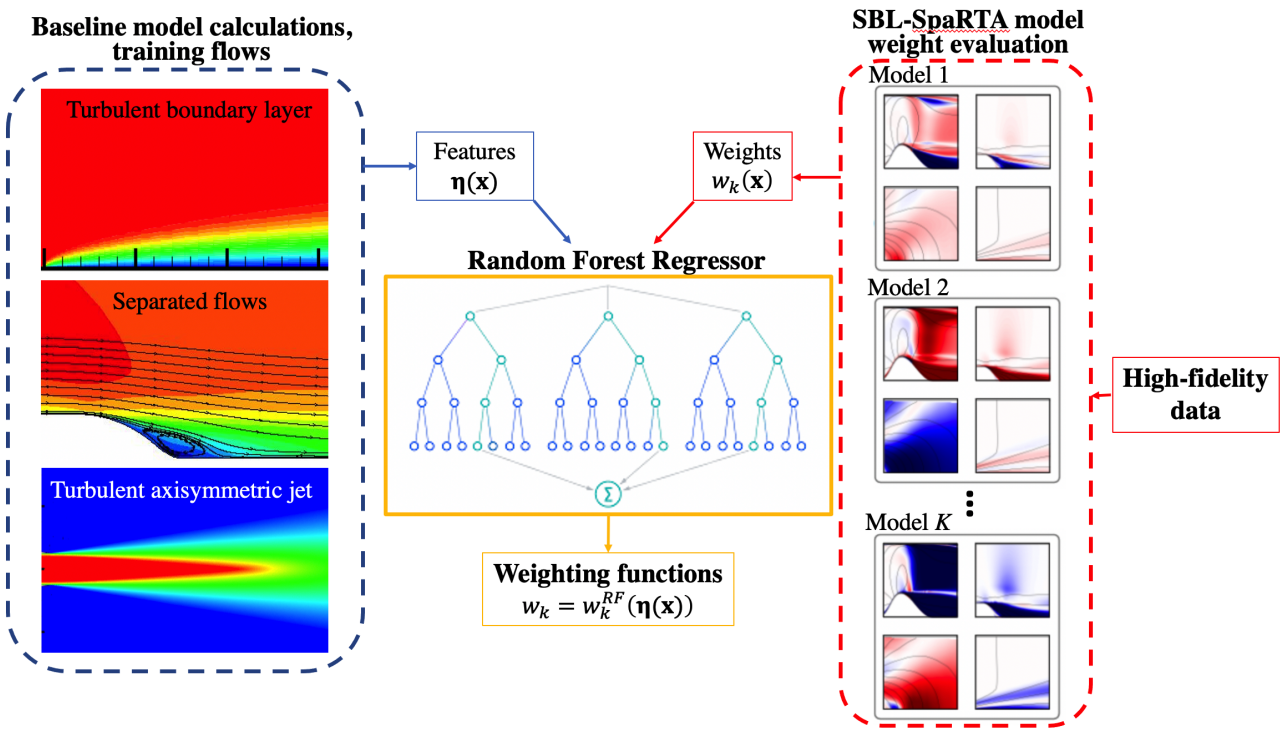


Figure 5.5: Workflow of X-MA training. The baseline k-ω SST model is used to evaluate flow features for a set of training flows including flat plates with various pressure gradients, separated flows and a jet flow (left part); K SBL-SpaRTA model solutions for the training cases are compared with high-fidelity data (right part) to evaluate the gating function (5.9) and the model weights (5.8). The features and the corresponding weights are used to train Random Forests Regressors that map the local flow features into model weights.

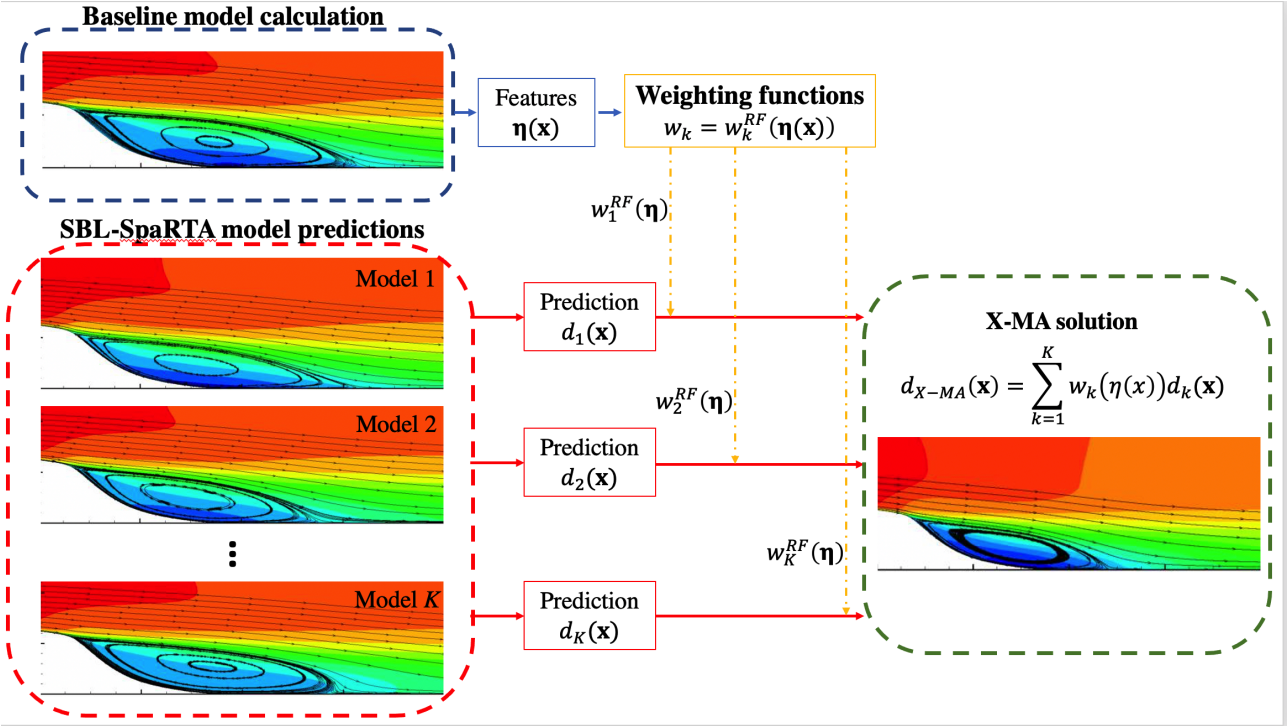


Figure 5.6: Workflow for X-MA predictions. The baseline $k-\omega$ SST model is used to evaluate flow features for a new flow, and K SBL-SpaRTA candidate models are used to generate predictions of the same flow. The features are used to interrogate the Random Forests Regressors for the weighting functions, and the weights are finally used to aggregate the candidate models solutions into the X-MA prediction.

5.2.2 Complete X-MA learning process

In the following is described the complete X-MA learning process:

1. A set $\mathcal{K}$ of K SBL-SpaRTA corrective models are considered.

$$\mathcal{K} = \left\{\mathbf{M}^{(ZPG)}, \mathbf{M}^{(CHAN)}, \mathbf{M}^{(APG)}, \mathbf{M}^{(SEP)}, \mathbf{M}^{(ANSJ)}\right\}$$

2. A set $\mathcal{C}$ of C flows of various configurations are considered for which high-fidelity data are available at some points in the physical space: $\mathcal{C} = \{CHAN, APG, ANSJ, SEP\}$. Each dataset corresponding to a flow configuration $c \in \mathcal{C}$ is denoted by $\mathcal{D}_c$ ($\mathcal{D}_{CHAN}$, $\mathcal{D}_{ANSJ}$...). $\mathcal{D}_c = \{\mathbf{x}_i^{(c)}, \mathbf{QoI}\left(\mathbf{x}_i^{(c)}\right)\}_{i=1}^{N_c}$ where $\mathbf{QoI}\left(\mathbf{x}_i^{(c)}\right)$ is a raw vector of QoI evaluated at $\mathbf{x}_i^{(c)}$, and N_c the number of data points in $\mathcal{D}_c$. In this study, horizontal velocity will be the only QoI

used for model weights construction so that **QoI** = (U). Let $\mathcal{D}$ be the dataset bringing together the C datasets of flow cases: $\mathcal{D} = \{\mathcal{D}_c\}_{c\in\mathcal{C}}$. Each $\mathcal{D}_c$ is splitted into two subsets $\mathcal{D}_c^{(1)}$ and $\mathcal{D}_c^{(2)}$. We denote $\mathcal{D}^{(1)} = \left\{\mathcal{D}_c^{(1)}\right\}_{c\in\mathcal{C}}$ and $\mathcal{D}^{(2)} = \left\{\mathcal{D}_c^{(2)}\right\}_{c\in\mathcal{C}}$.

3. For each $c \in \mathcal{C}$:

   (a) K RANS computations, corresponding to the K corrective models, are performed using a grid mesh comprising, among others, the $\mathbf{x}_i^{(c)}$s of $\mathcal{D}_c^{(1)}$.

   (b) A set of vectors $\boldsymbol{\eta}_i^{(c)}$ of features are computed from the baseline $k-\omega$ SST and for each $\mathbf{x}_i^{(c)}$ of $\mathcal{D}_c^{(1)}$.

4. For every $2\sigma_w^2 \in \mathcal{E} = \{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ and QoI $\in$ **QoI**:

   (a) Exact weights are computed for each point in $\mathcal{D}_c^{(1)}$ and for ever $c \in \mathcal{C}$ by using Equations (5.8) and (5.9) (the QoI is denoted δ in these equations). The set of exact weights is denoted $\mathcal{W}_{2\sigma_w^2,c}^{(\text{QoI})}$.

   (b) Accounting for the C weights sets $\left\{\mathcal{W}_{(2\sigma_w^2),c}^{(\text{QoI})}\right\}_{c\in\mathcal{C}}$, and the C features sets $\left\{\boldsymbol{\eta}_i^{(c)}\right\}_{c\in\mathcal{C}}$, the parameters of a global regression $\mathcal{R}_{(2\sigma_w^2)}^{(QoI)}$ are estimated.

   (c) Based on the regression $\mathcal{R}_{(2\sigma_w^2)^2}^{(\text{QoI})}$, an aggregated X-MA solution is computed on a grid including the $\left\{\mathbf{x}_i^{(c)}\right\}_{c\in\mathcal{C}} \subset \mathcal{D}^{(2)}$.

   (d) The discrepancy between the aggregated X-MA solution and the high-fidelity data is estimated over $\mathcal{D}^{(2)}$, by using the improvement metric $Imp_{QoI}(\%)$ over the baseline k-ω SST model relative to several QoI on a set of data points D:

$$Imp_{QoI}(\%) = \left[1 - \frac{\sum_{\mathbf{x}\in D}\Big(QoI(\mathbf{x}) - QoI^{HF}(\mathbf{x})\Big)^2}{\sum_{\mathbf{x}\in D}\Big(QoI^{baseline}(\mathbf{x}) - QoI^{HF}(\mathbf{x})\Big)^2}\right] \times 100 \tag{5.13}$$

5. Based on the improvement measures of 4d, a "best" set of models' weights $\mathcal{W}_{(2\sigma_w^2)^*}^{(\text{QoI})}$ is selected for every QoI $\in$ **QoI**:

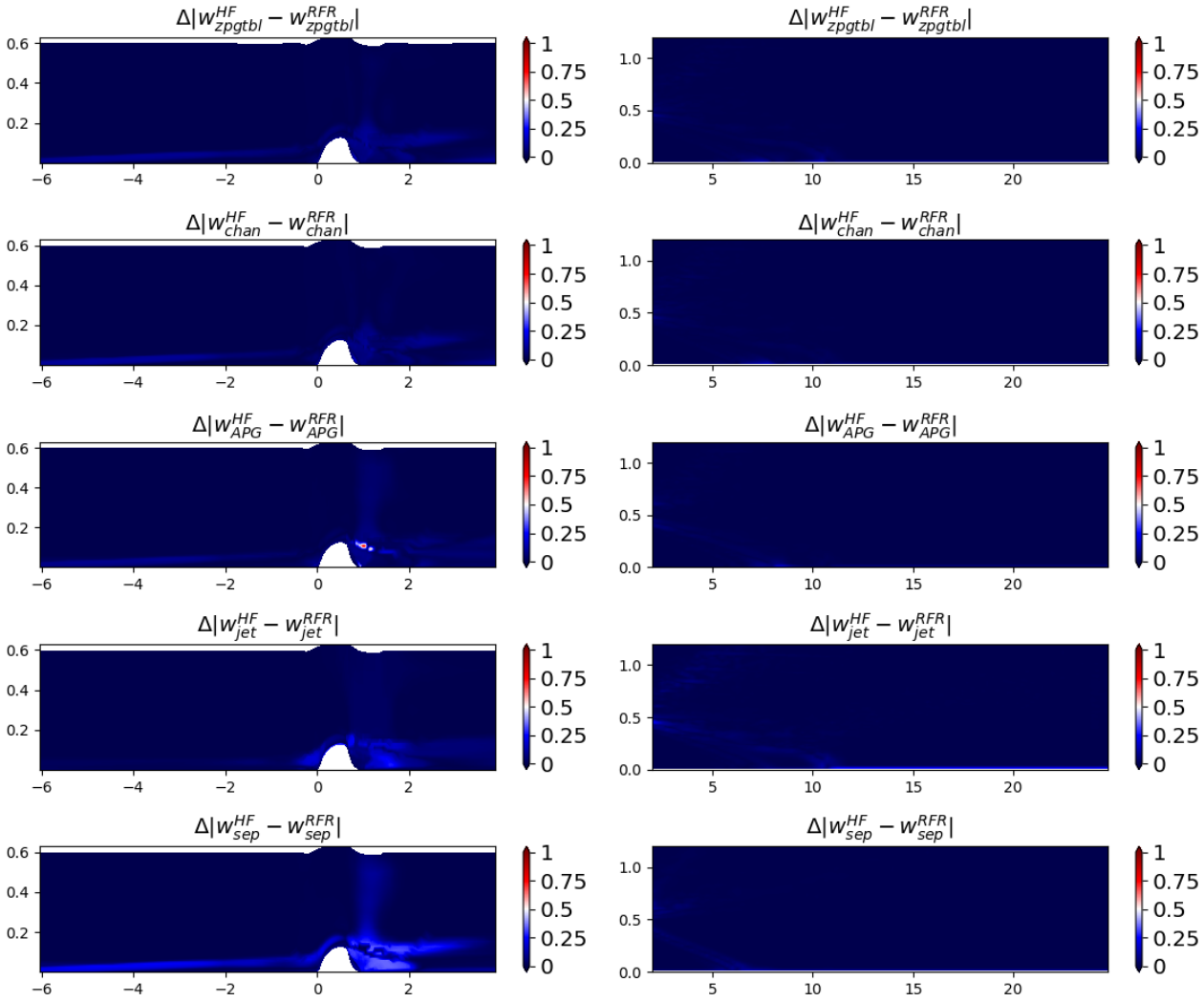
$$\sigma_w^*\Big(\text{QoI}\Big) = \underset{2\sigma_w^2\in\mathcal{E}}{\arg\max}\Big(Imp_{\text{QoI}}(\%)\Big) \tag{5.14}$$

6. A new flow is predicted from the CFD solver, in conjunction with the X-MA method and the regressor $\mathcal{R}^{(\mathrm{QoI})}_{(2\sigma^2_w)^*}$.

Preliminary tests showed that this configuration yielded a very good training. The validation score R^2, defined by

$$R^2 = 1 - \frac{\sum_i (\overline{w}(\mathbf{x}_i) - w(\mathbf{x}_i))^2}{\sum_i (\overline{w}(\mathbf{x}_i) - \mathbb{E}(\overline{w}(\mathbf{x}_i))^2}, \quad \begin{cases} \overline{w}(\mathbf{x}_i): \text{ computed from high-fidelity data} \\ w(\mathbf{x}_i): \text{ RFR predicted} \end{cases}$$

is greater than 0.97 for both $\mathcal{D}_c^{(1)}$ (training error) and $\mathcal{D}_c^{(2)}$ (generalization error) $\forall c \in \mathcal{C}$. Regarding unseen scenarios, R^2 is greater than 0.95, for ASJ and 2DZP, and greater than 0.85, for 2DWMH. Absolute discrepancies between high-fidelity data and RFR predictions, reported in Figures 5.7a and 5.7b, show very low values.



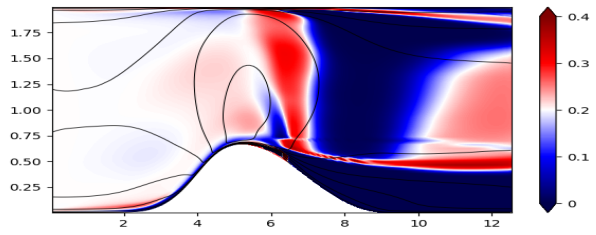
(a) 2DWMH test case.

(b) ASJ test case.

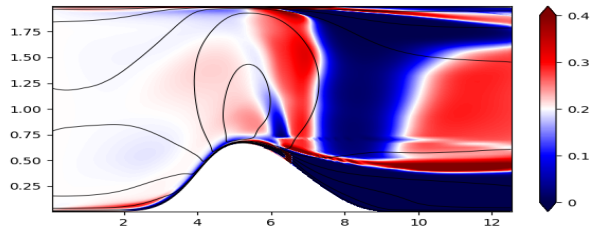
Figure 5.7: $\Delta|w^{RFR} - w^{HF}|$.

On the other hand, it is worth noting that, using the methodology, relevant SBL-SpaRTA models are assigned the right weight in different regions of the domain. For instance, in Figures 5.8 and 5.9, are plotted the colormaps of the weights w_k computed from high-fidelity data for two training cases (namely, CD and ANSJ) and the five SBL-SpaRTA models. In the same figures we also report the iso-contours of the longitudinal velocity of the baseline model, to illustrate how the models are scored in different flow regions. In the CD case (Figure 5.8), all

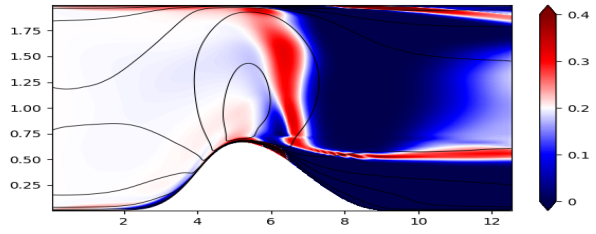
models are scored equally upstream of the throat, except at the beginning of the convergent, where the ANSJ model is downgraded compared to the others. On the other hand, in the separated region downstream of the throat, the SEP model is assigned a much higher weight than the other ones. The ANSJ model exhibits the worst score almost everywhere. For the ANSJ case (Figure 5.9), the models have essentially equal weights in the potential cone region, which is insensitive to the turbulence model. Immediately downstream of this region, the ZPG and APG models exhibit relatively high performance scores. As expected, the ANSJ model is assigned the highest score in the far jet region. On the contrary, the SEP model exhibits a very low score almost everywhere. Overall, all of these results underscore a significant outcome: customized models exhibit a clear regional performance, mainly directed by the change in the underlying physics. To explore additional maps of model weights for various training flow cases exhibiting similar patterns and interpretations, readers are encouraged to refer to the Appendix C.1.



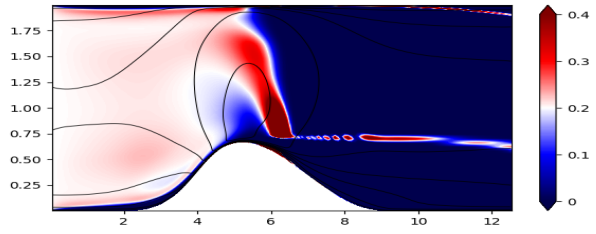
(a) w_{ZPG}



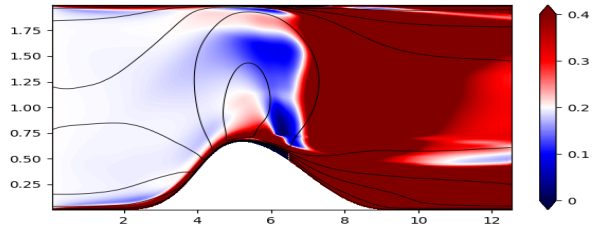
(b) w_{CHAN}



(c) w_{APG}

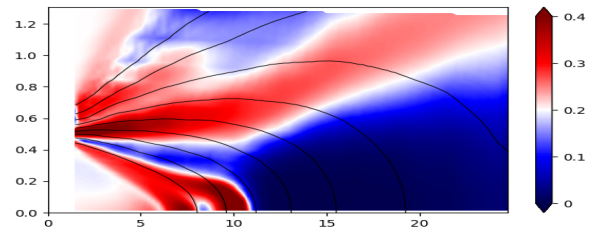


(d) w_{ANSJ}

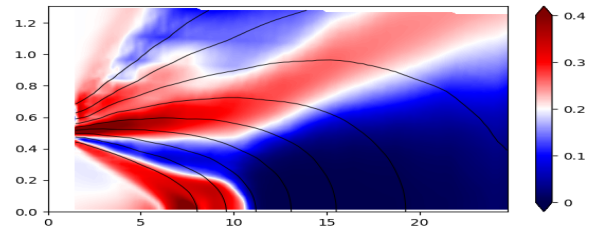


(e) w_{SEP}

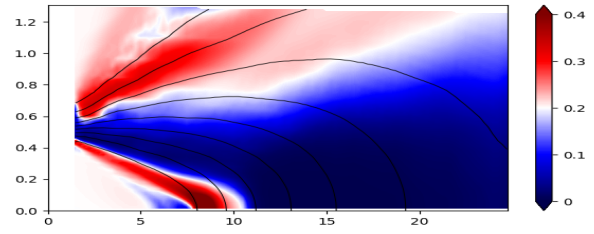
Figure 5.8: Colormaps of exact optimal model weights for the CD flow (various SBL-SpaRTA) and iso-contours of the longitudinal velocity (baseline model).



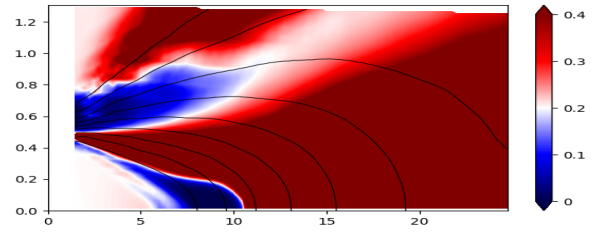
(a) w_{ZPG}



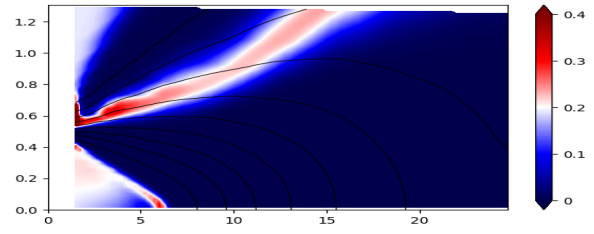
(b) w_{CHAN}



(c) w_{APG}



(d) w_{ANSJ}



(e) w_{SEP}

Figure 5.9: Colormaps of exact optimal model weights for the ANSJ flow (various SBL-SpaRTA) and iso-contours of the longitudinal velocity (baseline model).

5.3 Model aggregation results

In the following, we first apply the X-MA approach to two flows included in the training sets used to learn the SBL-SpaRTA corrections or the X-MA weighting functions. Then, we evaluate X-MA for generalization and compute an aggregated prediction for three unseen flows selected from those proposed in the NASA turbulence modeling testing challenge. [115]. In all cases, the X-MA results are obtained as follows:

1. The five models of Table 5.2 are used to make stochastic predictions of flow at stake, *i.e.* to estimate the mean $\mathbb{E}(d_k)$ and the variance $Var(d_k)$ of any output flow quantity.

2. At each mesh point, the baseline model solution (identical to the CHAN solution) is used to compute the features of Table 5.3

3. The features are fed to the trained RF to obtain the weighting functions w_k

4. The mean and variance of the X-MA aggregated solution are then computed.

5.3.1 Application of X-MA to flows in the training set

The X-MA is first applied to two flows among those used to train the weighting functions. We select one of the canonical fully-developed channel flows (CHAN) and one of the separated flows (CD). CHAN data were also used to train one of the SBL-SpaRTA models, which occurred to be identical to the baseline model (zero corrections). For CD, the model learned on the set of SEP flows in Table 5.1 improved the results over the baseline (CHAN model), but did not predict the small separation bubble in the divergent, while all other models largely overestimated the bubble size (see the results reported in Figure 5.4a).

In Figure 5.10 we plot the expectancy of the X-MA velocity profile along with the reference DNS data and baseline model. In the picture, the grey-shaded area represents the *convex accessible region*, *i.e.* the envelope of solutions given by the five component models. We also reported error bars, corresponding to $\pm 3\sqrt{Var_{\text{X-MA}}}$. The solution is in good agreement

with the DNS, and it slightly outperforms the baseline k-ω SST model in the log layer. The weights attributed to the various models are reported in the lower panel of the same figure. In the viscous sublayer all models are assigned equal weights, since all models predict the linear solution. In the log layer, the ANSJ model is downgraded, whereas the ZPG and APG models are assigned similar weights because they exhibit a similar performance (see Figure 5.2a). The SEP model is eventually assigned an increased weight in the defect layer, but its contribution remains small overall. Despite the large spread of the accessible area, the solution variance is rather small. This is due to the fact that 1) most models (except ANSJ) predict similar solutions with small posterior variance due to the residual uncertainties in model parameters and 2) the outlier model ANSJ is affected zero weight in the region where it strongly deviates from the other models.

Figure 5.11 displays the expectancy and error bars of the friction coefficient distribution for the CD case. For this case, the SEP model is assigned the highest weight (close to 1 in most regions), but the ANSJ model takes over in the throat region, where the SEP model is overly dissipative. Of note, the APG model is assigned a slightly higher weight than the other models in the divergent, *i.e.* in the adverse pressure gradient region, but its contribution remains similar to the ZPG and CHAN models overall. This confirms that the ZPG, CHAN and APG models behave rather similarly to each other and the baseline k-ω SST. The automatic selection of the locally best performing models by means of the weighting functions allows to capture the recirculation bubble, which was missed or overestimated by the component models. In most regions, the error bars are small because a single model tends to prevail. The X-MA prediction couldn't reach the high-fidelity values of C_f directly after reattachment since the convex accessible envelop of the solutions do not comprise these data points.

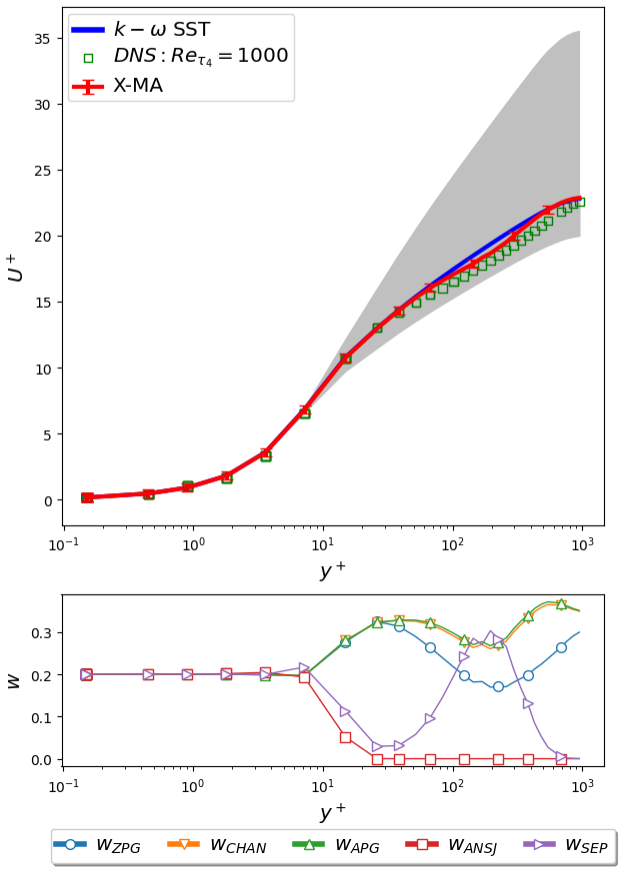


Figure 5.10: X-MA prediction of the velocity profile for the turbulent channel flow at $Re_\tau = 1000$. The grey shade represents the accessible region.

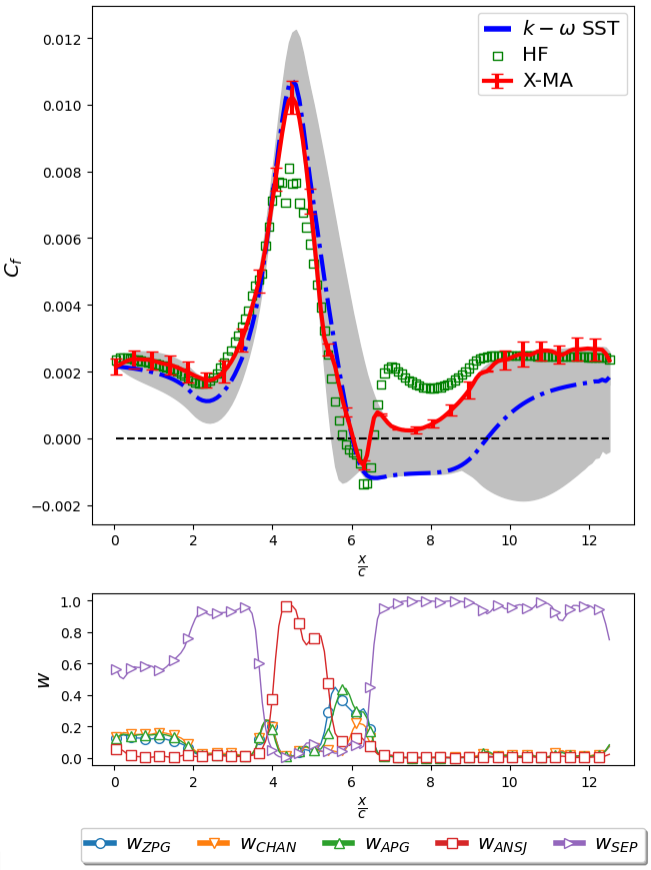


Figure 5.11: X-MA prediction of the skin friction distribution along the bottom wall for the converging-diverging (CD) channel flow. The grey shade represents the accessible region.

5.3.2 X-MA prediction of unseen flows

Next, we assess the X-MA for flow cases that were not used for training the SBL-SpaRTA models or the RF weighting functions. For that purpose, we selected three test cases of increasing difficulty from the NASA 2022 Symposium on Turbulence modeling Collaborative Testing Challenge [115], discussed in the following. More details about the setup and the computational

grids can be found in Chapter 3.

Turbulent flat plate (2DZP)

We consider the turbulent flat plate test case from `https://turbmodels.larc.nasa.gov/flatplate.html`. In Figure 5.12 (left panel) we report the velocity profile at the streamwise location $x = 0.97$ where $Re_\theta \approx 10000$. The Figure shows that the X-MA expectancy is in good agreement with both the baseline k-ω SST prediction and with Coles' mean velocity profile [116, 117]. In the Figure we also report the accessible region (grey shade) and error bars corresponding to ± 3 standard variations of the X-MA prediction. Such intervals are much narrower than the accessible zone because the outlier models (such as the ANSJ model) are assigned low weights, which penalizes their contribution to the total variance. Models trained on similar cases, *e.g.* ZPG, CHAN, and APG, exhibit comparable weighting functions (shown in the bottom part) . The SEP model, specifically tailored for separated flows, is assigned locally a slightly higher weight in the external part of the boundary layer, as previously observed for the channel flow case. Finally, the ANSJ is assigned low weight in most of the flow.

The right panel of Figure 5.12 displays the skin friction coefficient distribution along the plate wall and the corresponding weighting functions. The skin friction was not used for training the weights, but it is tightly related to the velocity profiles. As a consequence, we want to see if the weighting functions trained on the velocity are still meaningful. Again, the X-MA prediction is in good agreement with the baseline calculation, which in turn agrees well with the turbulent correlation (equation 6-121) from [118]. In this case the models are ranked according to the values that the weighting functions take at the wall location, where all models are almost equally weighted, with the ANSJ model being slightly below. Note that the goal of X-MA is not to select a single "best" model in each flow region, but to determine an optimal combination of the component models that captures the data. In this case, some models overpredict the skin friction, while others underpredict it. The RF captures this behavior and assigns approximately equal weights after excluding the outlier, so that on average X-MA predicts the correct value. Despite a large spread in model predictions, the ANSJ model is assigned again a low enough

weight to limit its contribution to the X-MA average and variance, which reduces the predictive uncertainty.

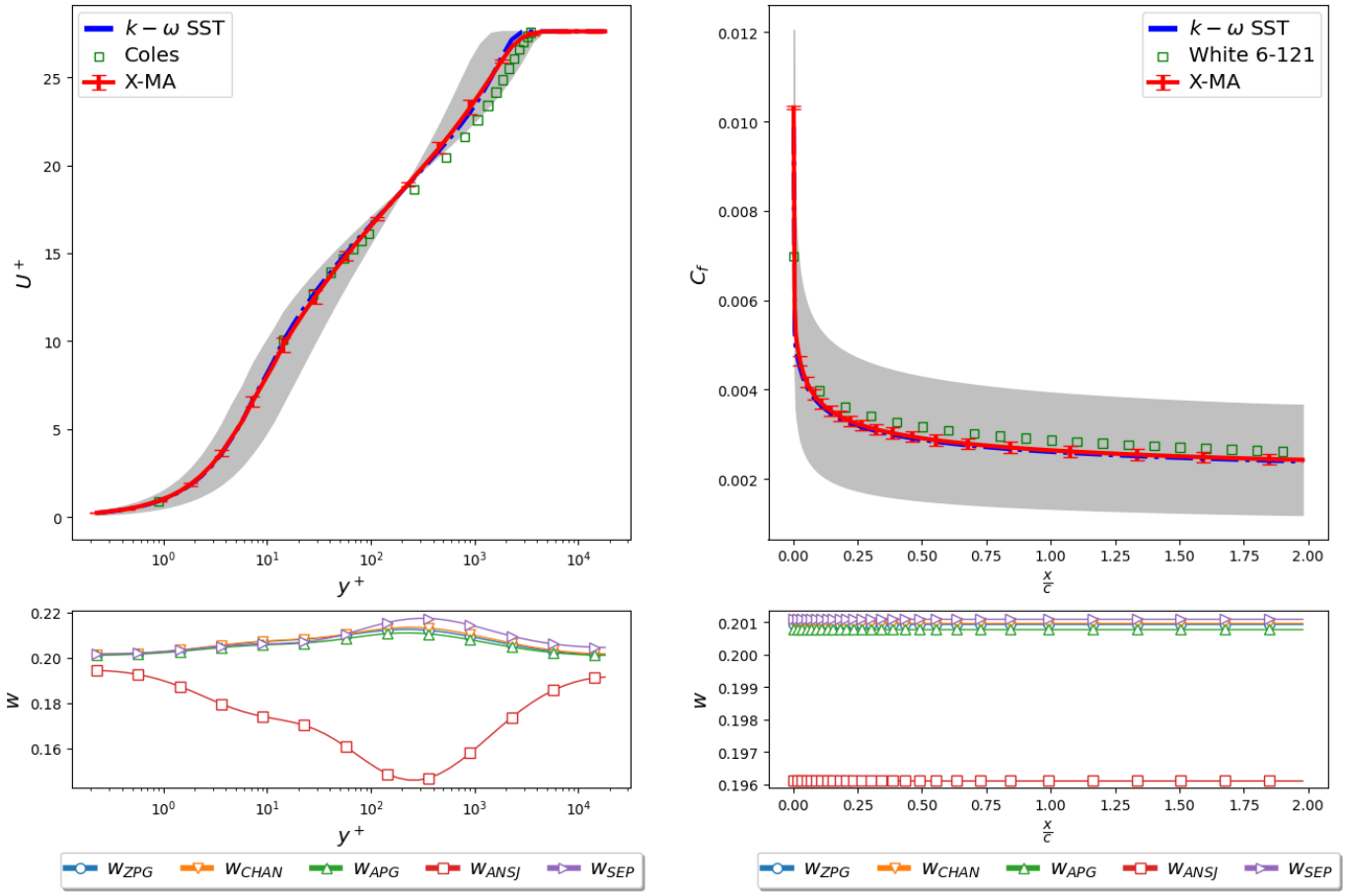


Figure 5.12: X-MA prediction of the velocity profile at $x = 0.97$ (top left) and of the skin friction distribution along the wall (top right) for the NASA turbulent flat plate flow case (2DZP). The bottom panels show the corresponding weighting function distributions. The grey shade represents the accessible region. Error bars correspond to ± 3 standard deviations.

Axisymmetric Subsonic Jet (ASJ)

The second test case is the NASA Axisymmetric subsonic jet. Figure 5.13 displays the distribution of the horizontal velocity along the jet axis. The X-MA expected solution shows an excellent agreement with the reference experimental data, and it represents a significant

improvement over the baseline. In the potential region, all component models are assigned approximately equal weights, except for the SEP model, which is slightly penalized. In the early jet region there is no clear winner and the mixture smoothly switches from one model to another, until only the ANSJ model emerges in the far jet. The velocity profiles reported in Figure 5.14 also match very well the reference data, except for the last profile, which is not included in the X-MA accessible region. In this case, the X-MA prediction lies along the boundary closest to the data, and its ± 3 standard deviations uncertainty bars encompass the reference data. Interestingly, the discrepancies among component models tend to increase when moving downstream. This showcases that at the farest stations, the ANSJ model performs better than the others yet exhibits a discrepancy with respect to the reference. The large uncertainty bars in the regions of dominance of w_{ANSJ} proves also that this region of the flow is highly sensitive to this models' calibration.

Wall-Mounted Hump (2DWMH)

The left panel of Figure 5.15 shows the pressure coefficient distribution along the wall. The X-MA captures very well the high-fidelity data, some discrepancies being visible within the separated region (corresponding to the pressure plateau located approximately between the abscissas 0.75 and 1). The predictions clearly outperform the baseline model in the diverging part. Upstream of the hump, the models designed for flat plates are assigned equal weights (reported in the bottom panel), the SEP model emerges as the highest weighted model throughout the flow, whereas the ANSJ model is assigned a low weight. The X-MA uncertainty bars are small upstream (except at pressure extrema) and they become larger in the separated region, highlighting the high sensitivity of this region to the injection of eddy-viscosity performed by the SEP model. In the right panel we report the skin friction distribution for the same case. X-MA captures rather well the reference, showing a great improvement over the baseline and capturing the reattachment location rather well. The weights are the same as for the pressure coefficient, since they only depend on the local flow features and not on the QoI to be predicted. The component models solutions for the C_f are widespread over a large accessible

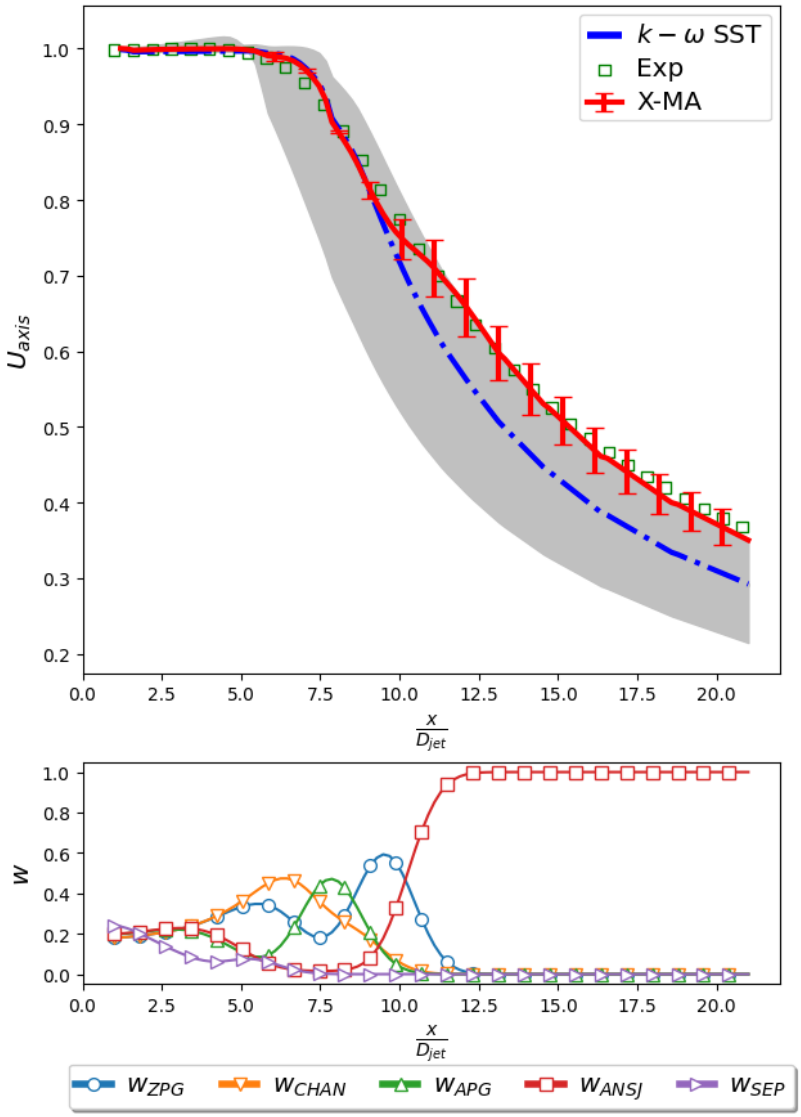


Figure 5.13: Distribution of the streamwise velocity along symmetry axis for the Axisymmetric Subsonic Jet (ASJ) case. The grey shade represents the accessible region. Error bars correspond to ±3 standard deviations.

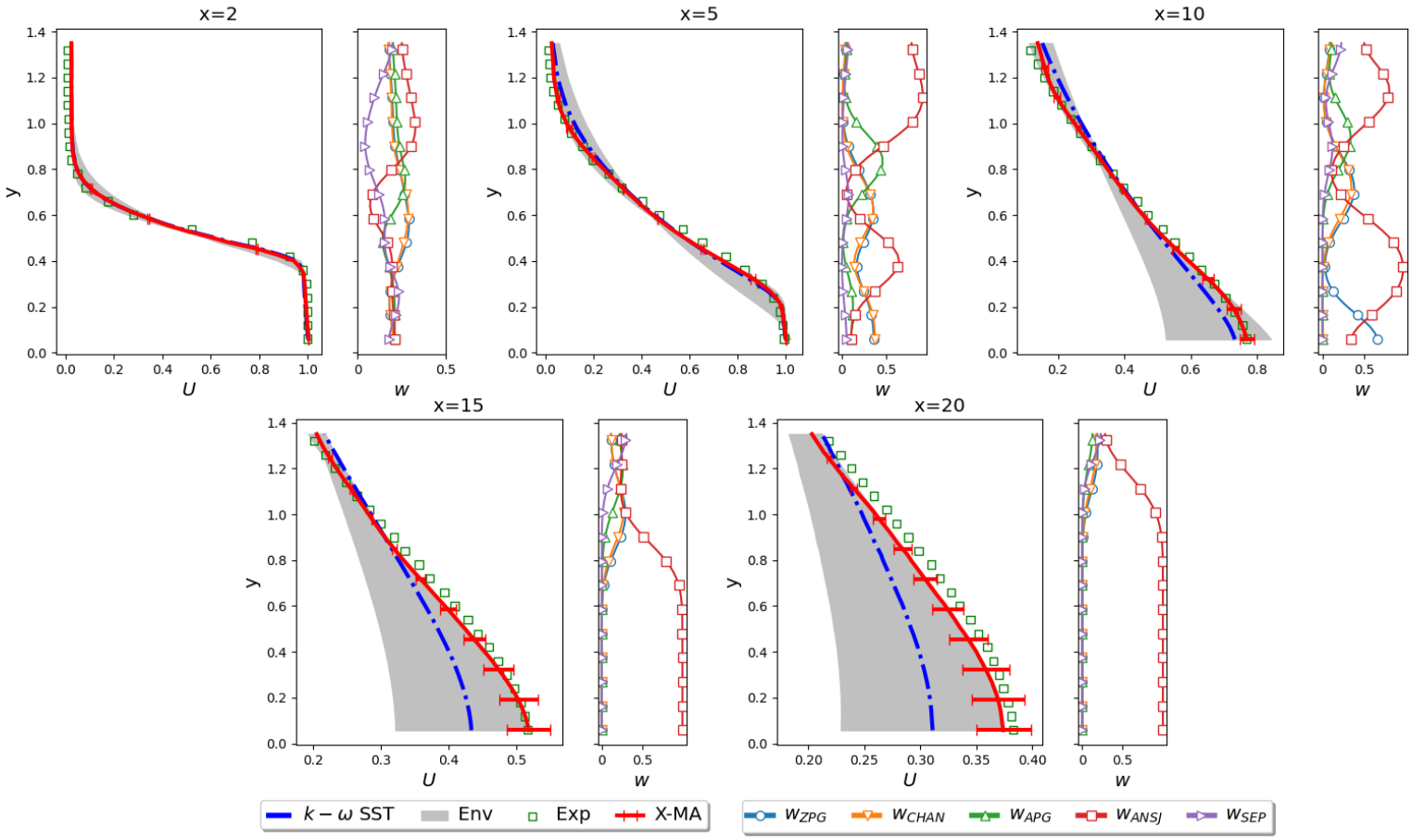


Figure 5.14: Profiles of the streamwise velocity U at various horizontal locations for the Axisymmetric Subsonic Jet (ASJ) case.

area, but again the error bars are rather small because only one of the models is assigned a high weight, while the unsuitable model here, the ANSJ model, which is responsible for this significant discrepancy in the accessible area, is evidently rated poorly, and the three models (CHAN, ZPG and APG) predict similar solutions. Unfortunately, the bars do not always fully encompass the reference solution, but they are very closely aligned and show clear improvement compared to the baseline. In Figures 5.16 and 5.17, we can observe a good agreement between the aggregated X-MA solutions for both horizontal velocity and Reynolds shear stress and the high-fidelity data when compared to the baseline model. This agreement is particularly present in the separated region ($0.8 \leq x \leq 1.2$), where the velocity profiles closely match to the experimental data, in contrast to the baseline model, which fails to exhibit attached velocity profiles. Concerning the Reynolds shear stress profiles, the X-MA prediction closely approaches the higher values observed in the high-fidelity data, surpassing again the performance of the baseline model.

5.3.3 Summary of the results and discussion

To provide an overall picture of the performance of the proposed X-MA methodology on unseen flow cases, the improvement metric (5.13) is displayed in Table 5.4 for the three prediction cases and with various models, namely, the individual customized models and the aggregated X-MA prediction. The results show that the customized models perform remarkably well for flows similar to those they have been trained on. However, their performance significantly deteriorates for significantly different flows. Conversely, the X-MA prediction consistently outperforms the baseline k-ω SST model for all cases. In certain cases, the X-MA prediction even surpasses the performance of the optimal customized model. This shows that the X-MA prediction effectively captures the combined effects of the different customized models used in the mixture by enhancing the prediction locally where the customized model predictions may not be optimal.

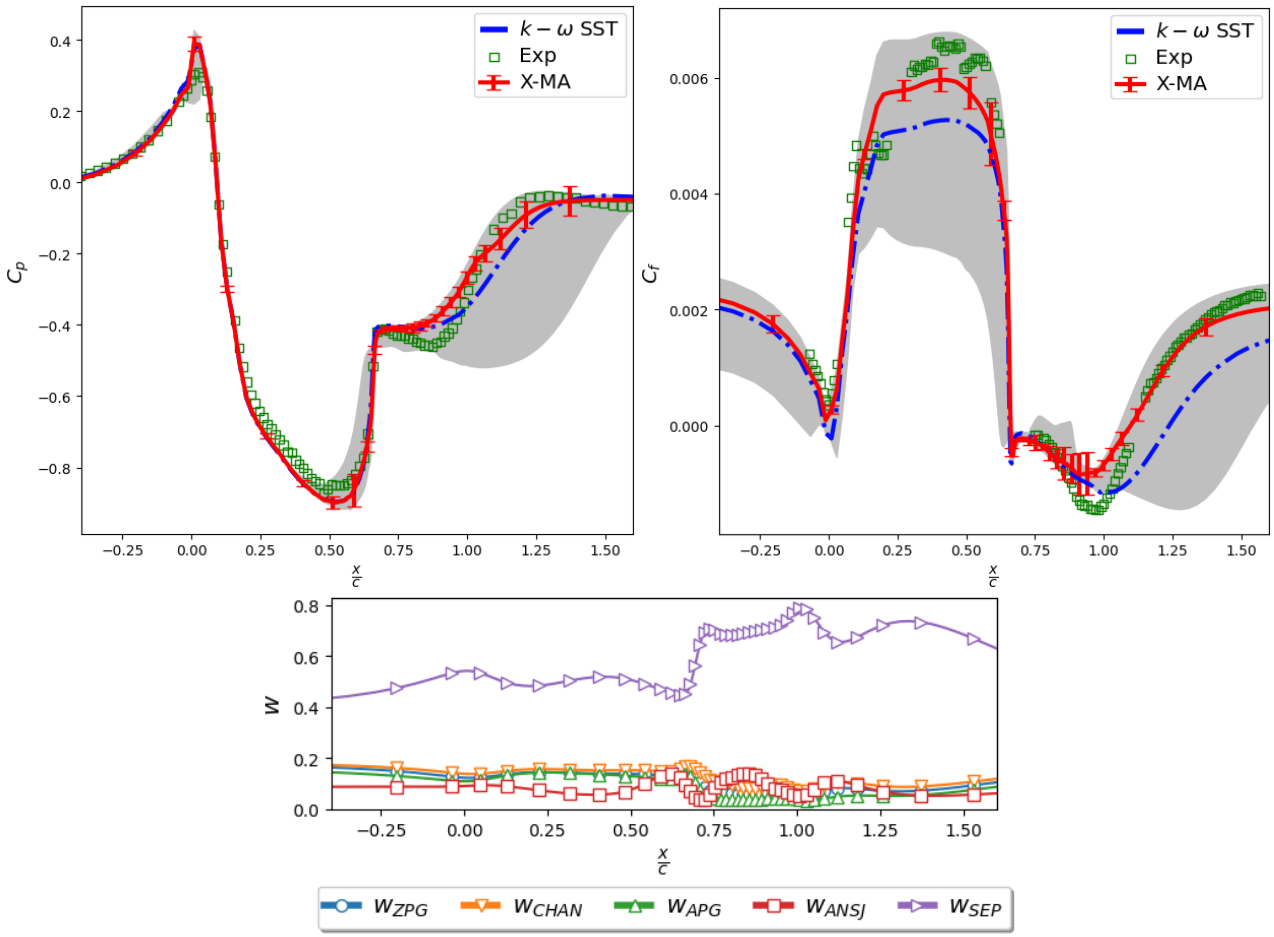


Figure 5.15: Distribution of the pressure coefficient (top left) and skin friction coefficient (top right) along the wall for the NASA wall-mounted hump case (2DWMH). The weighting function distributions are reported in the bottom panel. The grey shade represents the accessible region. Error bars correspond to ±3 standard deviations.

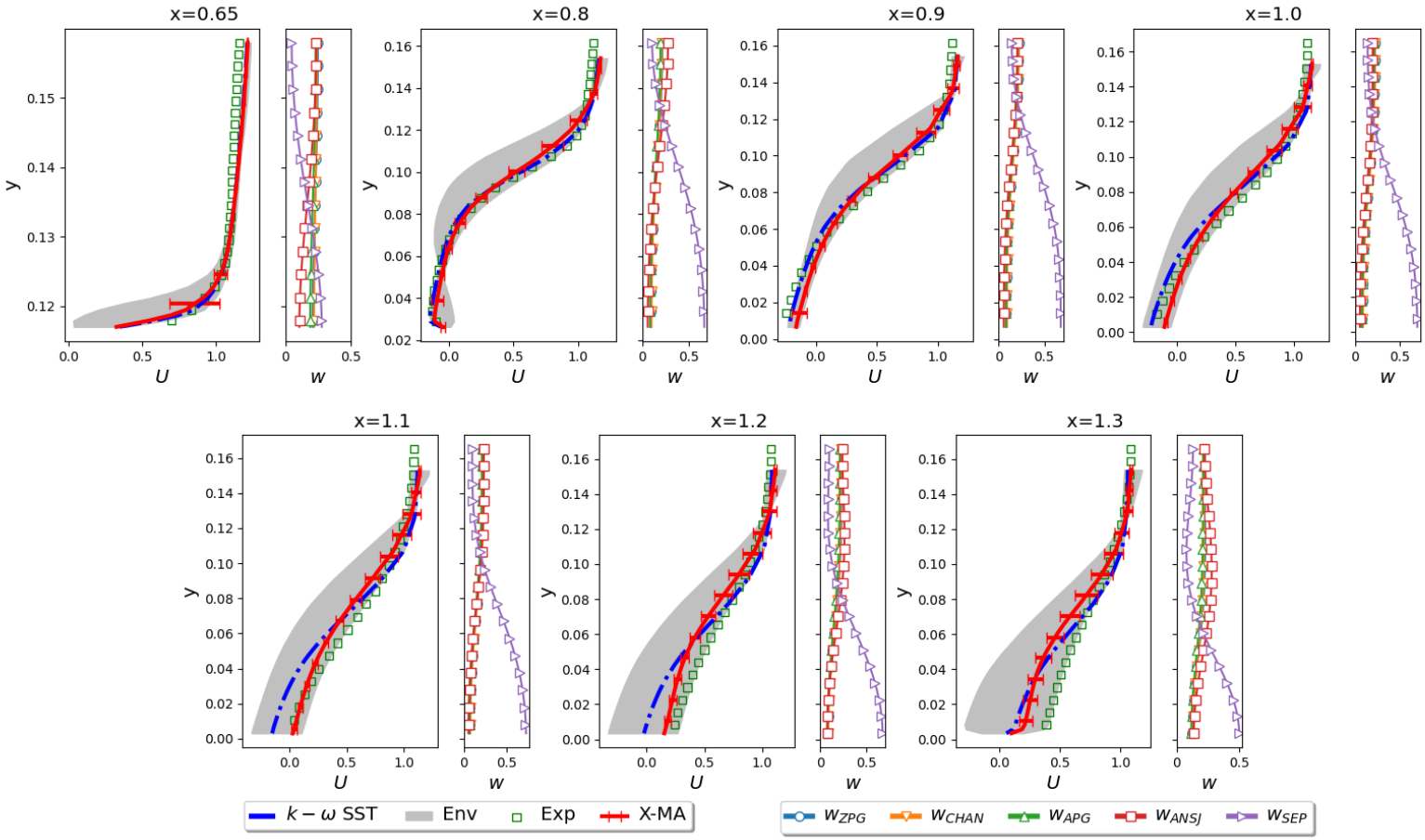


Figure 5.16: Profiles of the streamwise velocity U at various horizontal locations for the Wall-Mounted Hump case.

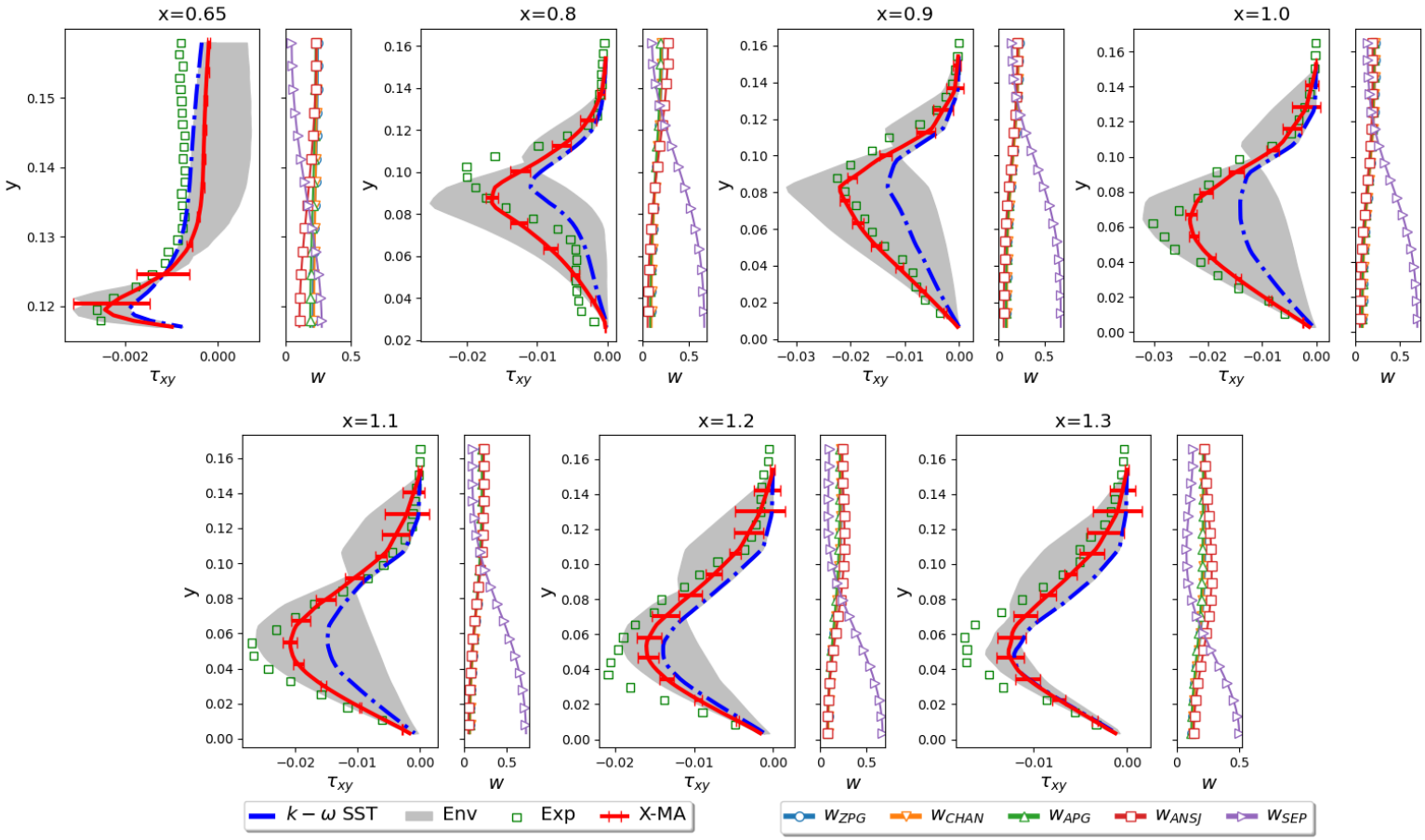


Figure 5.17: Profiles of Reynolds shear stress τ_{xy} at various horizontal locations for the Wall-Mounted Hump case.

case	QoI	X-MA	$M^{(ZPG)}$	$M^{(CHAN)}$	$M^{(APG)}$	$M^{(ANSJ)}$	$M^{(SEP)}$
2DZP	U	**7.0**	6.6	0	5.5	-927.1	-241.0
	C_f	**10.6**	6.2	0	4.2	-3580.0	-1868.0
ASJ	U	**79.6**	8.5	0	-13.6	72.0	-535.9
	τ_{xy}	**13.3**	5.6	0	-21.3	51.6	-595.1
2DWMH	U	**76.0**	-18.9	0	-70.6	-1265.8	65.4
	τ_{xy}	**74.1**	-3.8	0	37.0	-152.4	60.9
	C_p	**17.9**	-6.7	0	-36.0	-1098.0	-21.5
	C_f	**64.0**	-18.2	0	-9.7	-819.2	47.6

Table 5.4: Improvements in (%) wrt the baseline k-ω SST for the test cases.

5.4 Conclusions

In this chapter, we presented a machine-learning methodology for aggregating data-driven turbulence models trained for narrow classes of flows and thus providing predictions of more general unseen flows. The model aggregation also provides an estimate of predictive uncertainty.

First, the customized turbulence models are trained on several flow cases encompassing selected types of physical phenomena and operating conditions, following the SBL-SpaRTA framework presented in details in Chapter 4. Then, in order to make predictions of more general flows including features of the various training flow classes, we propose a Mixture-of-Experts approach, named space-dependent model aggregation (X-MA) consisting in building a local convex linear combination of the solutions predicted by the set of learned turbulence models by means of weighting functions that depend on a vector of well-chosen flow features and reflecting the local plausibility of every model.

Results prove first that the customized models perform well for flows within or close to those in the training set, but extrapolate badly to very different flows, because of opposite correction requirements. A good example is given by the jet flow and the separated flows: for the jet, the correction tends to reduce the model eddy viscosity, while for the separated flows terms contributing to increase turbulent dissipation are needed. For flat plate and channel flows (including flat plate flows with adverse pressure gradients), the learned model corrections produce no or little changes with respect to the baseline model. Moreover, postdiction and

predictions, resp. both on training and test cases, clearly demonstrate that the appropriate models are activated in their respective zones of expertise, leading to significantly improved aggregated solution across various quantities of interest. Importantly, the aggregation approach does not compromise the accuracy of predictions for canonical flows, where the baseline models already perform well. In addition, the model mixture provides an estimate of the predictive variance, *i.e.* a measure of model uncertainty: in regions of large discrepancies of the individual solutions, the X-MA variance becomes larger, thus warning the user about the reliability of the computed prediction. The cost of the uncertainties calculations can be greatly reduced by using a sparse uncertainty quantification method for selecting and computing a reduced number of the PC expansion coefficients, such as the one available in the *equadratures* package [113] used in this study.

Despite the promising results, the external model aggregation approach does not constitute a turbulence model, but rather an uncertainty quantification method. The non-intrusive X-MA method consists in post-processing individual model predictions, and the aggregated prediction is not a solution of the conservation equations. This is why in the next chapter we investigate an intrusive version of X-MA, consisting in generating a blended turbulence model that is then propagated through the mean flow equations.

Chapter 6

Intrusive space-dependent aggregation of data-driven turbulence models

Contents

The methodology proposed in the previous chapter showed promise for merging the predictions of different data-driven models, or "experts", each one trained on a different class of flows. The approach, inspired by model aggregation techniques used in so-called ensemble machine learning, ends up producing an estimate of the expected (average) solution, weighted according to the local model performance scores. However, such an averaged prediction, however, is not necessarily consistent with the conservation equations. This is acceptable in the context of uncertainty quantification, but it may be a problem for applications where the exact conservation of some physical quantities (*e.g.* mass and energy) is mandatory. In addition, the RFR approximation of the model weights functions is not always smooth, requiring the application of a filter.

For all of these reasons, this chapter explores an alternative approach, referred to as "in-

trusive X-MA", which uses the model weights to construct a blended SBL-SpaRTA model that combines various customized SBL-SpaRTA corrections of the Reynolds stress tensor.

6.1 Intrusive X-MA

Let K be a set of SBL-SpaRTA models. In contrast to the non-intrusive version described in Chapter 5, in the case of the intrusive method, mixing occurs directly at the level of Reynolds tensor construction. Therefore, the equations solved by the solver are:

$$\begin{cases} \dfrac{\partial U_i}{\partial x_i} = 0 \\ U_j \dfrac{\partial U_i}{\partial x_j} = -\dfrac{1}{\rho}\dfrac{\partial P}{\partial x_i} + \dfrac{\partial}{\partial x_j}\left(\nu \dfrac{\partial U_j}{\partial x_J} - \boxed{\tau_{ij}}\right) \\ \dfrac{\partial k}{\partial t} + U_j \dfrac{\partial k}{\partial x_j} = \boxed{P_k} + \boxed{R} - \beta^* k\omega + \dfrac{\partial}{\partial x_j}\left((\nu + \sigma_k \nu_t)\dfrac{\partial k}{\partial x_j}\right) \\ \dfrac{\partial \omega}{\partial t} + U_j \dfrac{\partial \omega}{\partial x_j} = \dfrac{\gamma}{\nu_t}(\boxed{P_k} + \boxed{R}) - \beta\omega^2 + \dfrac{\partial}{\partial x_j}\left((\nu + \sigma_\omega \nu_t)\dfrac{\partial \omega}{\partial x_j}\right) \end{cases}$$

with

$$\begin{cases} \tau_{ij} = 2k\left(\dfrac{1}{3}\delta_{ij} + b_{ij}^0 + \displaystyle\sum_{k=1}^{K} w_k(\boldsymbol{\eta}(\mathbf{x})) b_{ij}^{\Delta^{(k)}}\right) \\ P_k = \min\left(2\nu_t S^2 - 2k\left(\displaystyle\sum_{k=1}^{K} w_k(\boldsymbol{\eta}(\mathbf{x})) b_{ij}^{\Delta^{(k)}}\right)\dfrac{\partial U_i}{\partial x_j}, 10\beta^*\omega k\right) \\ R = 2k\left(\displaystyle\sum_{k=1}^{K} w_k(\boldsymbol{\eta}(\mathbf{x})) b_{ij}^{R^{(k)}}\right)\dfrac{\partial U_i}{\partial x_j} \end{cases}$$

where the $b_{ij}^{\Delta^{(k)}}$ and the $b_{ij}^{R^{(k)}}$ are the corrections for the k^{th} SBL-SpaRTA model. The construction of weight functions is also performed similarly to the non-intrusive method :

1. in a first step, the exact values of weight functions are calculated at certain points within the domain where high-fidelity data are available by using Equations (5.8) and (5.9)

rewritten below:

$$\begin{cases} w_k(\delta^{(k)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w) = \dfrac{g_k(\delta^{(k)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w)}{\sum_{l=1}^{K} g_l(\delta^{(l)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w)} \\ g_k\Big(\delta^{(k)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w\Big) = \exp\left(-\dfrac{\Big(\delta^{(k)}(\mathbf{x}) - \bar{\delta}(\mathbf{x})\Big)^T . \Big(\delta^{(k)}(\mathbf{x}) - \bar{\delta}(\mathbf{x})\Big)}{\sqrt{Var(\bar{\delta})} \quad \times \quad 2\sigma_w^2} \right) \end{cases}$$

2. in the next step, an algorithm of regression is utilized to learn the relationship

$$\underbrace{\boldsymbol{\eta}(\mathbf{x}) = \Big(\eta_1(\mathbf{x}), ..., \eta_{11}(\mathbf{x})\Big)}_{\text{local flow features}} \xrightarrow[\mathcal{W}]{ML} \underbrace{\Big(w_1\Big(\delta^{(1)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w\Big), ..., w_K\Big(\delta^{(K)}(\mathbf{x}); \bar{\delta}(\mathbf{x}); \sigma_w\Big)\Big)}_{\text{local models weights}} \quad (6.1)$$

based on the values computed in the first stage, $\boldsymbol{\eta}$ still being the set of features described in Section 5.2.1.

First, let's remember that the features at each point in the flow domain are computed using the baseline $k-\omega$ SST model and are consistent with those used in the previous chapter. Similarly, both the local flow features and the model weights are computed for the training flow cases presented in Table 6.1:

Training cases	Description	Source
CHAN	DNS of turbulent channel flows	[90]
	$180 \leq Re_\tau \leq 5000$	[112]
ANSJ	PIV of near sonic axisymmetric jet	[95]
SEP	LES of Periodic Hills (PH) at $Re = 10595$	[99]
	DNS of converging-diverging channel (CD) at $Re = 13600$	[97]
	LES of curved backward facing step (CBFS) at $Re = 13700$	[98]

Table 6.1: List of data used to train the ML regressor of model weights.

Next, the coefficient σ_w still needs to be determined in the equations used to calculate the weights.

Regarding the regression step, although the RFR used in Chapter 5 to represent the weighting functions has proven effective in a non-intrusive setting, it is intrinsically non-smooth which

could lead to numerical problems when the blended model is integrated into the CFD solver in the intrusive setting. For this purpose, we look for an alternative regressor that can provide an accurate and smooth representation of the weighting functions. Preliminary tests showed that both linear and polynomial regression methods produce poor training scores when attempting to map the model weights to the space of physical flow features. This suggests that the relationship between the two is highly non-linear. Consequently, we seek a regressor capable of handling these complex, non-linear patterns while delivering smooth and reliable predictions, making Gaussian Process Regressors (GPR)[119] a good candidate for our modeling approach. GPR is a non-parametric and probabilistic ML model used for regression tasks. It excels in modeling complex and non-linear relationships between input and output variables, with the flexibility to adapt to various patterns. Key to GPR is the choice of kernel functions, which define data point similarities. Initial tests demonstrated that when experimenting with various kernel functions, the radial basis function (RBF) kernels consistently produced the smoothest and most accurate regression results. Given the importance of these characteristics in our chosen regressor, we have opted for the RBF kernels. The model hyperparameters, including noise levels and smoothness, are learned from the data. To train the GPR, the total number of training points was limited to approximately $N = 3000$ to manage the training cost of the GPR, which involves inverting an $N \times N$ matrix. It is worth noting that this dataset size is smaller than what is used for training the RFR in the previous chapter. We spaced the data points to prevent redundancy and, in areas where no turbulence is expected, as far away from the boundary layer, we used even wider space between points to save computational resources.

Finally, the resulting turbulence model correction consists of all the posterior probability distributions for the chosen models' parameters and their associated deterministic tensor and weighting terms, making it a stochastic model. This stochastic correction is propagated into the flow solver using once again the sparse PCE (Polynomial Chaos Expansion) [113].

In Chapter 5, we show that, for the 2D cases under consideration, the most influential mod-

els among the five considered as building blocks are the $\mathbf{M}^{(CHAN)}$ model, which is identical to the baseline $k-\omega$ SST, the separated model $\mathbf{M}^{(SEP)}$, and the jet model $\mathbf{M}^{(ANSJ)}$. As a consequence, and in order to simplify the intrusive setting, in this chapter we retain only these three models.

The complete intrusive X-MA process is detailed in Section 6.1.1.

6.1.1 Complete intrusive X-MA process

As with non-intrusive X-MA, the description of the complete intrusive process includes, on the one hand, the search for an optimal σ_w and, on the other, the optimal regression parameters. In practice, the optimal σ_w is sought from the set $\mathcal{E} = \{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ using a grid search method. The optimization criterion is based on the comparison of a certain QoI between high-fidelity data and intrusive X-MA. For better readability, weights computed from a given QoI are denoted w_{QoI} (weight calculation implies σ_w). Horizontal velocity U and Reynolds shear stress τ_{xy} are the QoI considered in this study for weights construction. In contrast to the unified hyperparameter approach in the non-intrusive X-MA, further numerical tests have demonstrated that better results can be achieved by choosing different hyperparameters for each training flow dataset. The complete process is as follows:

1. A set $\mathcal{K}$ of K SBL-SpaRTA corrective models are considered. $\mathcal{K} = \{$baseline $k-\omega$ SST, $\mathbf{M}^{(SEP)}, \mathbf{M}^{(ANSJ)}\}$.

2. A set $\mathcal{C}$ of C flows of various configurations are considered for which high-fidelity data are available at some points in the physical space. $\mathcal{C} = \{CHAN, ANSJ, CD, CBFS, PH\}$. Each dataset corresponding to a flow configuration $c \in \mathcal{C}$ is denoted by $\mathcal{D}_c$ ($\mathcal{D}_{CHAN}$, $\mathcal{D}_{ANSJ}$...). $\mathcal{D}_c = \{\mathbf{x}_i^{(c)}, \mathbf{QoI}\left(\mathbf{x}_i^{(c)}\right)\}_{i=1}^{N_c}$ where $\mathbf{QoI}\left(\mathbf{x}_i^{(c)}\right)$ is a raw vector of QoI evaluated at $\mathbf{x}_i^{(c)}$, and N_c the number of data points in $\mathcal{D}_c$. The horizontal velocity and the Reynolds shear stress are the QoI considered in the study so that $\mathbf{QoI} = (U, \tau_{xy})$. Each $\mathcal{D}_c$ is splitted into two subsets $\mathcal{D}_c^{(1)}$ and $\mathcal{D}_c^{(2)}$.

3. For each $c \in \mathcal{C}$:

   (a) K RANS computations, corresponding to the K corrective models, are performed using a grid mesh comprising, among others, the $\mathbf{x}_i^{(c)}$s of $\mathcal{D}_c^{(1)}$.

   (b) A set of vectors $\boldsymbol{\eta}_i^{(c)}$ of features are computed from the baseline $k-\omega$ SST and for each $\mathbf{x}_i^{(c)}$ of $\mathcal{D}_c^{(1)}$.

   (c) For each $2\sigma_w^2 \in \mathcal{E}$ and each QoI $\in$ **QoI** component :

      i. Exact weights are computed for each point in $\mathcal{D}_c^{(1)}$ by using Equations (5.8) and (5.9) (the QoI is denoted δ in these equations). The set of exact weights is denoted $\mathcal{W}_{2\sigma_w^2,c}^{(\text{QoI})}$.

      ii. The parameters of the regression $\mathcal{R}_{2\sigma_w^2,c}^{(\text{QoI})}$ are computed by means of $\boldsymbol{\eta}_i^{(c)}$ and $\mathcal{W}_{2\sigma_w^2,c}^{(\text{QoI})}$ (Equation 6.1).

      iii. Based on the regression $\mathcal{R}_{2\sigma_w^2,c}^{(\text{QoI})}$, an aggregated X-MA solution is computed on a grid including the $\mathbf{x}_i^{(c)}$.

      iv. The discrepancy between the aggregated X-MA solution and the high-fidelity data is estimated over $\mathcal{D}_c^{(2)}$, by using the improvement measure $Imp_{QoI}(\%)$ (Equation (5.13)). The improvement metric is evaluated using the same QoI employed to construct the model weights.

   (d) Based on the improvement measures of 3c, a "best" set of model weights $\mathcal{W}_{(2\sigma_w^2)^*,c}^{(\text{QoI})}$ is selected for every QoI $\in$ **QoI**:

$$\sigma_w^*\Big(\text{QoI}, \mathcal{D}_c^{(2)}\Big) = \underset{2\sigma_w^2 \in \mathcal{E}}{\arg\max}\Big(Imp_{\text{QoI}}(\%)\Big) \tag{6.2}$$

   (e) To select an optimal weights set (there is one set per **QoI** component resulting from step 3d), the improvement metric is applied to the quantity of interest q not used to construct model weights:

$$\mathcal{W}_{(2\sigma_w^2)^*,c}^{(\text{QoI})^*} = \underset{q \in \mathbf{QoI}-\{\text{QoI}\}}{\arg\max}\Big(Imp_q(\%)\Big) \tag{6.3}$$

4. Accounting for the C weights sets $\mathcal{W}_{(2\sigma_w^2)^*,c}^{(\text{QoI})^*}$ and the C features sets $\boldsymbol{\eta}_i^{(c)}$, the parameters of a latest global regression $\mathcal{R}$ are estimated (Equation 6.1).

5. A new flow is predicted from the CFD solver, in conjunction with the X-MA method and the regressor $\mathcal{R}$.

To ensure a fair comparison between the intrusive and non-intrusive X-MA, the same process is employed to select the optimal set of weights and train the GPR accordingly for both paradigms. The distinction in their application lies in the calculation of the "aggregated X-MA solution", used to compute the improvement metric in steps 3(c)iv, 3d and 3e:

- For the non-intrusive X-MA, the aggregated X-MA prediction of a QoI from is obtained as the linear combination of the K RANS solutions provided by the building-block models, via the weighting functions (Eq. (5.11)).

- For the intrusive X-MA, the weighting functions are used to determine the X-MA blended correction b^{Δ} and R. The latter is then propagated through the CFD solver to predict the QoI.

Of note, both the intrusive and the non-intrusive X-MA approaches yielded the same weighting functions, which are trained externally prior to the prediction step.

Based on numerical tests, the choice of the horizontal velocity U, as the QoI used for weight calculation, leads to more accurate results than any other QoI - more details can be found in Appendix D where different kind of data are used to train the model weights. In Table 6.2 we summarize the improvements achieved over the baseline model for the training cases using different values of the hyperparameter σ_w. In the same table we also report results obtained by using the single component models.

case	$2\sigma_w^2(U)^*$	QoI	Non-intrusive X-MA	Intrusive X-MA	$\mathbf{M}^{(CHAN)}$	$\mathbf{M}^{(ANSJ)}$	$\mathbf{M}^{(SEP)}$
CHAN	1	U	91.6	78.9	0	-15654	-319.3
		τ_{xy}	48.1	-186.5	0	-90.6	-247
ANSJ	10^{-4}	U	80.6	53.1	0	78.7	-260.7
		τ_{xy}	57.6	51.9	0	78.7	-334.3
CD	10^{-2}	U	73.0	68.3	0	-642.3	31.3
		τ_{xy}	24.8	20.5	0	-198.3	31.1
CBFS	10^{-3}	U	85.2	78.3	0	-393.2	93.6
		τ_{xy}	78.1	72.7	0	-84.7	83.5
PH	10^{-4}	U	86.3	78.7	0	-132.8	83.3
		τ_{xy}	20.4	36.1	0	-70.4	27.6

Table 6.2: Improvements in (%) wrt the baseline k-ω SST on training cases using the optimal w_U

While $\mathbf{M}^{(SEP)}$ and $\mathbf{M}^{(ANSJ)}$ models outperform the baseline model ($\mathbf{M}^{(CHAN)}$) solely in their respective training flows, the non-intrusive X-MA consistently surpasses the baseline model in all training cases. The intrusive X-MA also outperforms the baseline in all training cases, but is relatively less accurate than the non-intrusive method. The reasons for this slight decrease in performance of intrusive X-MA compared to non-intrusive X-MA will be discussed in Section 6.2.

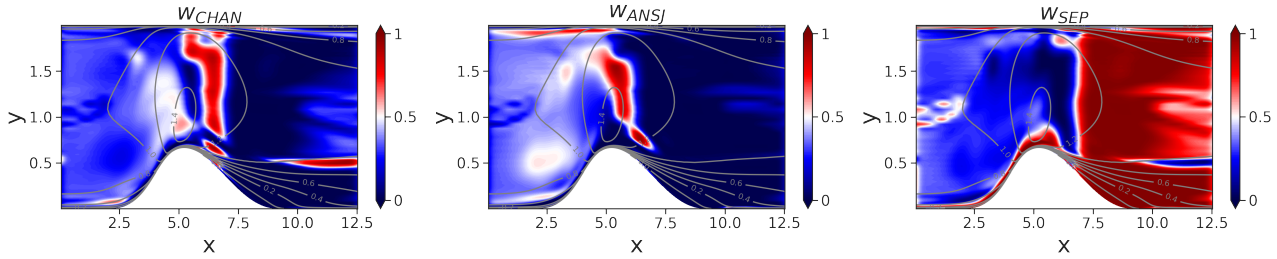
In addition to the improvement metric, the absolute error between the weights calculated from the high-fidelity data and those calculated by regression was plotted for the different training and test cases. The results show that the regression makes very few errors with only minor exceptions possibly related to the GPR architecture, data informativeness, or features mismatches in test cases. These results are available in Appendices D.3.1 and D.3.2.

Finally, the resulting set of model weights for both training and test flow cases (Tables 6.1 and 6.3), respectively, can be found in Figures 6.1 and 6.2.

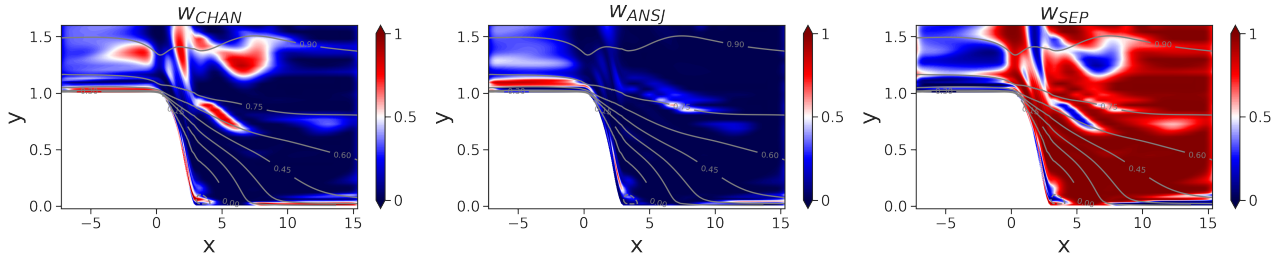
Test cases	Description
2DZP	2D Zero Pressure Gradient Flat Plate Validation Case
ASJ	Axisymmetric Subsonic Jet
2DWMH	2D NASA Wall-Mounted Hump Separated Flow Validation Case

Table 6.3: List of test flow cases.

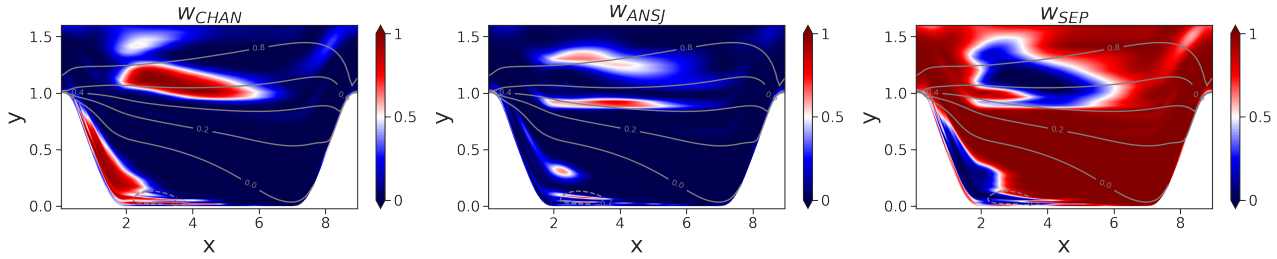
In Figure 6.1, separated regions are primarily characterized by high weights of $\mathbf{M}^{(SEP)}$. The $k-\omega$ SST model is notably prominent in the dead water region for separated cases and near the flat plate boundary layer in the CBFS case, as anticipated. In contrast, $\mathbf{M}^{(ANSJ)}$ dominates the far jet region in the ANSJ flow case, while the $k-\omega$ SST model receives high weights in the region between the near jet and far jet. The spatial distribution of model weights reaffirms the localized advantage of customized models. While these models are designed for specific cases, their benefits are only applicable in certain regions. This reinforces the hypothesis that there is a need to aggregate these models. In the test cases, we focus on 2DWMH and ASJ (Figure 6.2). In the 2DWMH case, $\mathbf{M}^{(SEP)}$ is prominent in the recirculation region, while downstream, the $k-\omega$ SST model takes precedence in the established attached boundary layer. $\mathbf{M}^{(ANSJ)}$'s importance is reduced, except near the hump, where strong pressure gradients occur. In the ASJ case, we observe a weight distribution similar to that of the ANSJ case: $\mathbf{M}^{(ANSJ)}$ is dominant in the far jet region, the $k-\omega$ SST model in the region between the far and near jet and in the outer region of the free shear layer, and $\mathbf{M}^{(SEP)}$ weights is overall negligible.



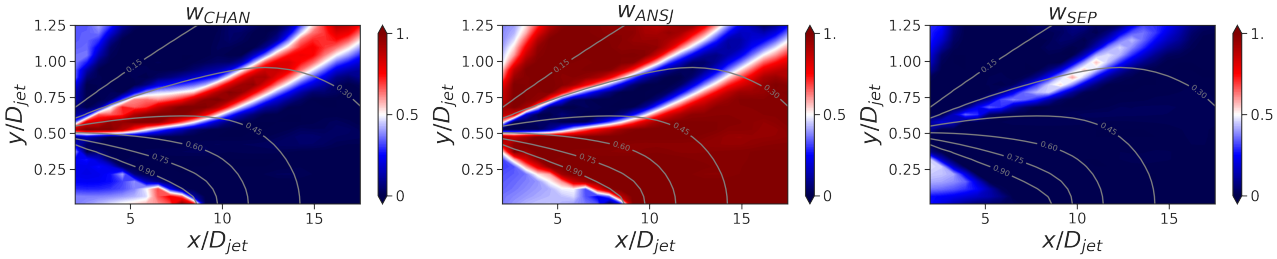
(a) Converging diverging channel.



(b) Curved backward facing step.

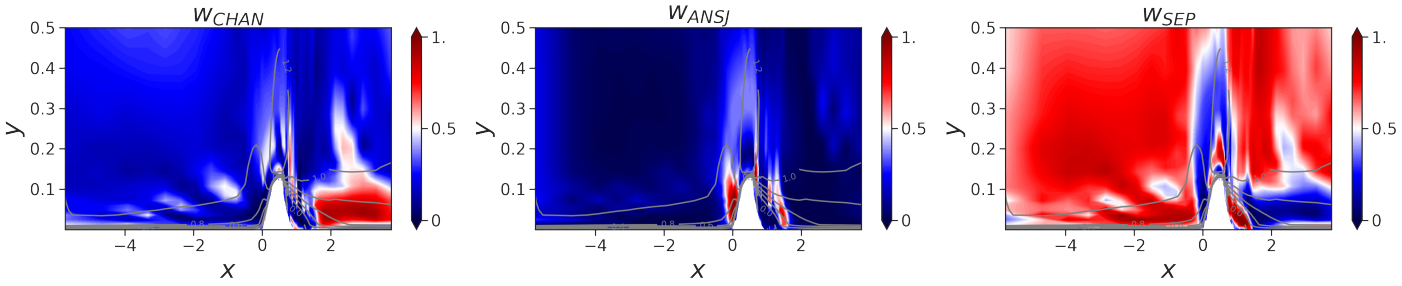


(c) Periodic hills.

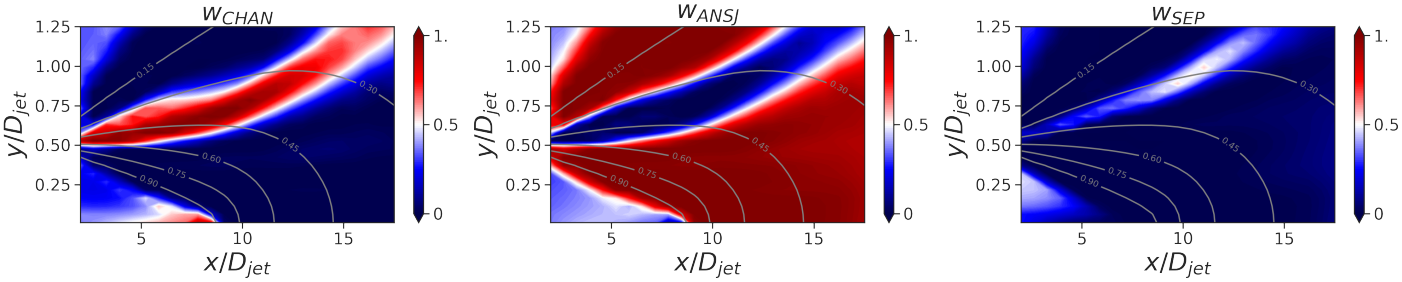


(d) Axisymmetric near-sonic jet.

Figure 6.1: Optimized model weights for the training flow cases.



(a) Wall-mounted hump.



(b) Axisymmetric subsonic jet.

Figure 6.2: Optimized model weights for the test flow cases.

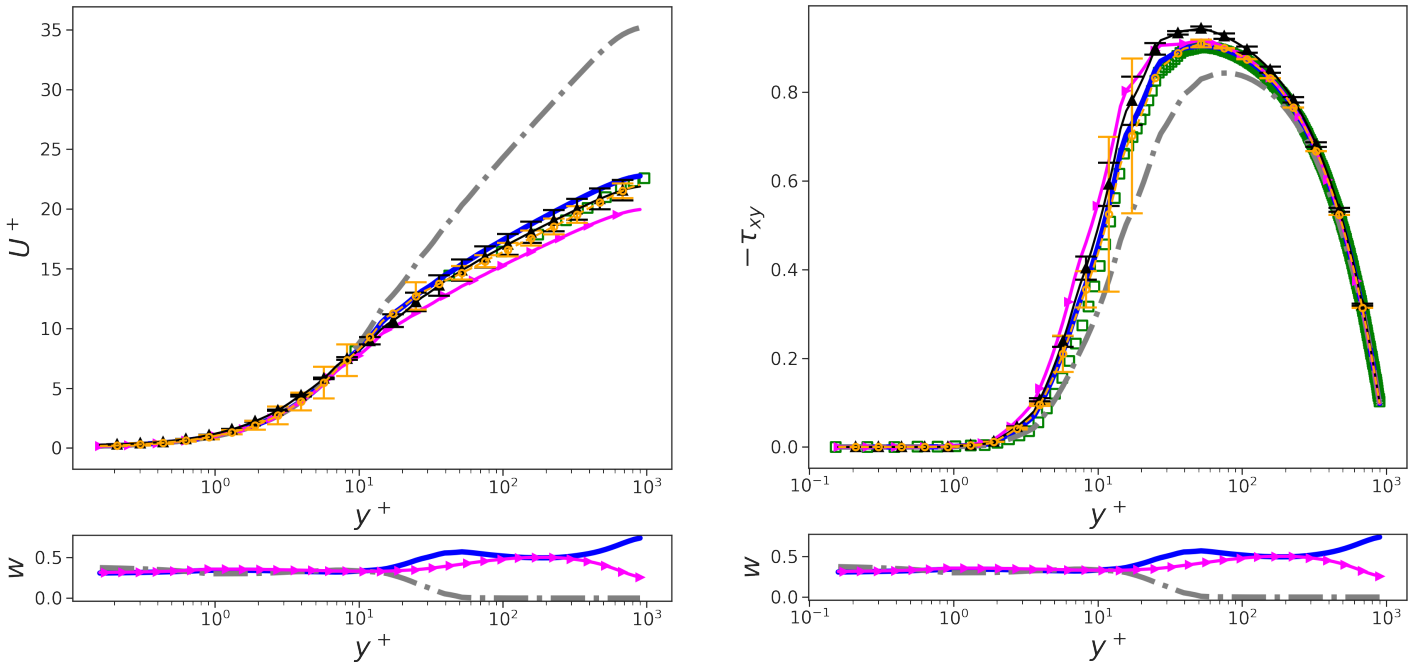
6.2 Comparison of the intrusive and non-intrusive X-MA

In this section, we evaluate the predictive performance for both intrusive and non-intrusive X-MA. We begin with a presentation of the training results as represented by a series of plots of various QoI in the training flow cases. Following this, we shift our focus to a meticulous analysis of the predictions on test flow cases, serving as an evaluation for the generalization capabilities of both intrusive and non-intrusive X-MA paradigms in unseen scenarios. These test plots rigorously assess the effectiveness of the paradigms under the choice of horizontal velocity to construct model weights as well as the use of Gaussian Process Regressors (GPR) as regression algorithm when confronted with previously unseen data.

6.2.1 X-MA results for flows in the training set

Turbulent channel flow

We first consider the results for a turbulent channel flow at $Re_\tau = 1000$, illustrated in Figure 6.3a. This figure first highlights that both intrusive and non-intrusive predictions match well with the high-fidelity data, slightly surpassing the baseline $k-\omega$ SST model in the log region. Within the viscous sublayer of the boundary layer, all the models are assigned similar weights, due to the reduced turbulent stresses. However, a noticeable enhancement is observed in the logarithmic zone of the boundary layer. This improvement can be attributed to the combination of $\mathbf{M}^{(SEP)}$ and the baseline $k-\omega$ SST in the logarithmic region, which leads to a more accurate slope in the logarithmic portion of the velocity profile. Further away from the logarithmic zone, in Figure 6.3a, a shift is noticeable as the baseline $k-\omega$ SST begins to dominate over $\mathbf{M}^{(SEP)}$. On the other hand, both $\mathbf{M}^{(SEP)}$ and $\mathbf{M}^{(ANSJ)}$ are inaccurate in modeling the velocity profiles. In particular, the performance of $\mathbf{M}^{(ANSJ)}$ is particularly poor, and this is effectively reflected in its corresponding local weight, starting from the exit of the viscous sublayer. In Figure 6.3b, the non-intrusive X-MA prediction fits better the high-fidelity data, while the intrusive one exhibits a little discrepancy, primarily evident in the log region.



(a) U^+. (b) τ^+_{xy}.

Figure 6.3: U^+ vs. y^+ and τ^+_{xy} vs. y^+ and the corresponding model weights w for the periodic channel flow case at $Re_\tau = 1000$. $\mathbf{M}^{(CHAN)}$ (—); $\mathbf{M}^{(ANSJ)}$ (-·-·-); $\mathbf{M}^{(SEP)}$ (—►—); High-fidelity data (□); Non-intrusive X-MA (-◇-); Intrusive X-MA (—▲—).

Axisymmetric near-sonic jet flow

In Figure 6.4 we show the profile of the streamwise velocity along the axisymmetric axis, alongside the distributions of weights assigned to the candidate models. The baseline model ($\mathbf{M}^{(CHAN)}$) mispredicts the velocity profiles along the x-axis. $\mathbf{M}^{(SEP)}$ performs even worse, because it tends to increase the eddy viscosity production. On the contrary, the $\mathbf{M}^{(ANSJ)}$ model captures well the high-fidelity data in the far jet region $8D_{jet}$ to $10D_{jet}$. The non-intrusive X-MA captures very well the high-fidelity solution, which is located within the convex envelope of the component model solutions. Specifically, the non-intrusive solution captures well the transition between the near jet region (from 0 to $8D_{jet}$) and the far jet region (from 10 to $18Djet$), thanks to the smooth transition from one component model to another. The intrusive X-MA approach provides a rather accurate solution, surpassing the baseline $k-\omega$ SST model. However, it appears to be less accurate than the non-intrusive X-MA. We attribute this slight degradation in performance to transport effects: the extra eddy viscosity introduced by $\mathbf{M}^{(SEP)}$ in the inlet is transported downstream, reducing the jet spreading rate. A similar effect is also observed in Figure 6.5a for the horizontal velocity profiles at various stations.

The Reynolds shear stresses at the same locations, are shown in Figure 6.5b. The overall performance of X-MA remains good, with predictions that outperform the baseline model.

Separated flow cases

Finally, we use X-MA to predict one of the separated flows used for training, namely, the converging-diverging (CD) channel. In Figure 6.6 we report the friction coefficient distribution at the bottom wall, the horizontal velocity profiles, and the Reynolds shear stresses along the horizontal $x-$axis at various stations. Several comments are in order. First, in Figure 6.6a, we observe distinct behaviors from the three candidate models. the baseline $k-\omega$ SST predicts a relatively large recirculation bubble following the throat of the converging-diverging section, whereas $\mathbf{M}^{(ANSJ)}$ predicts an even larger separation bubble. In contrast, $\mathbf{M}^{(SEP)}$ improves the results overall, but predicts no separation bubble, the high-fidelity separation bubble being extremely short. Although based on only three models and a different weight

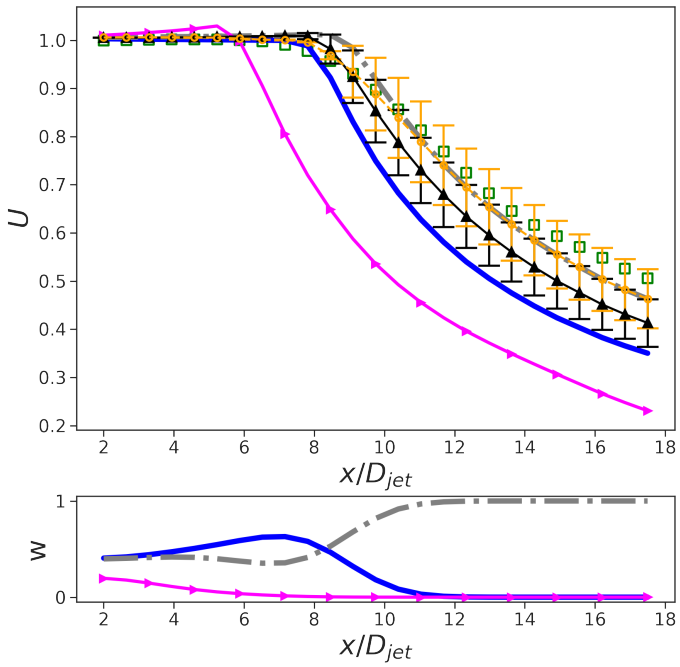
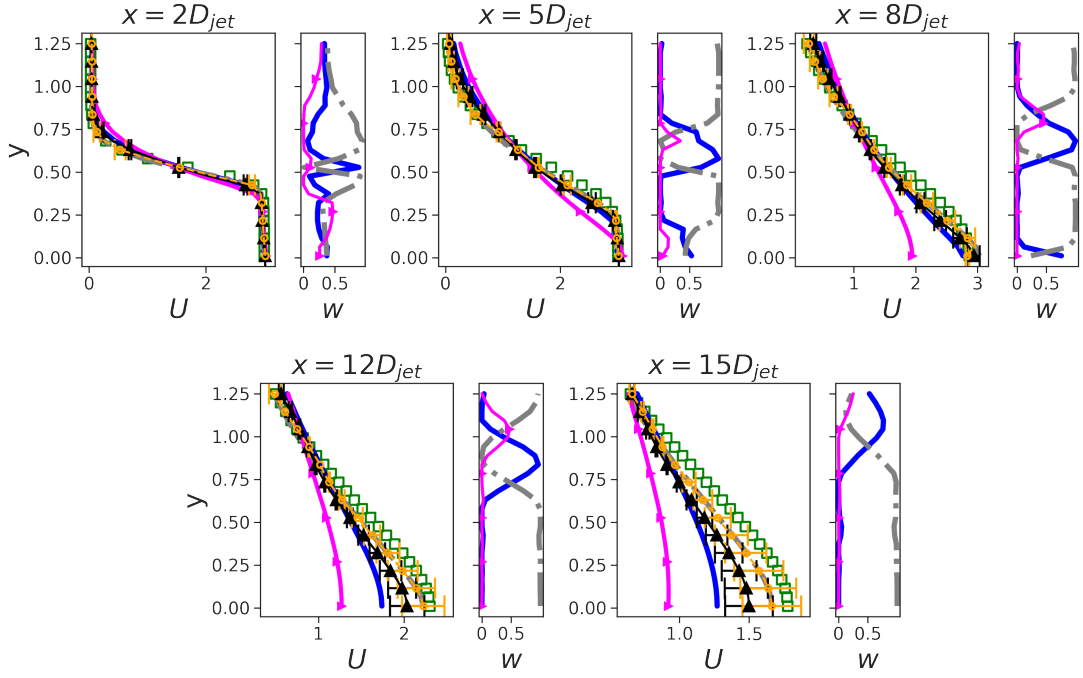
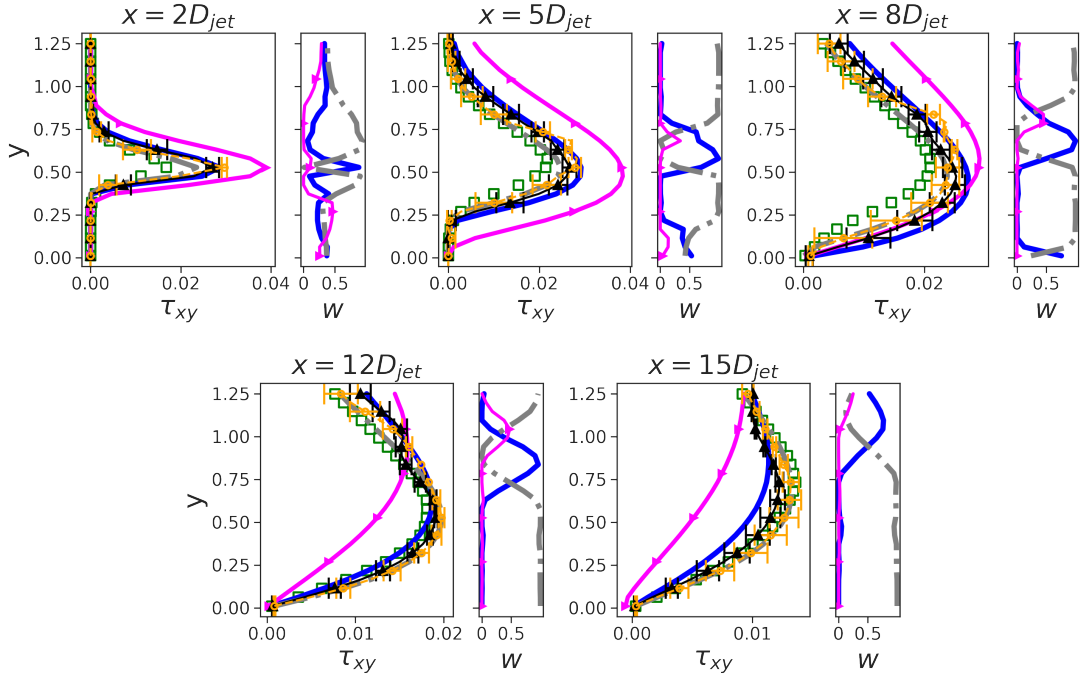


Figure 6.4: U vs. x/D_{jet} along the horizontal axis and the corresponding model weights w for the ANSJ flow case. Baseline $k-\omega$ SST (—); $\mathbf{M}^{(ANSJ)}$ (-·-); $\mathbf{M}^{(SEP)}$ (—►); High-fidelity data (□); Non-intrusive X-MA (-⧗-); Intrusive X-MA: (—▲—).



(a) U.

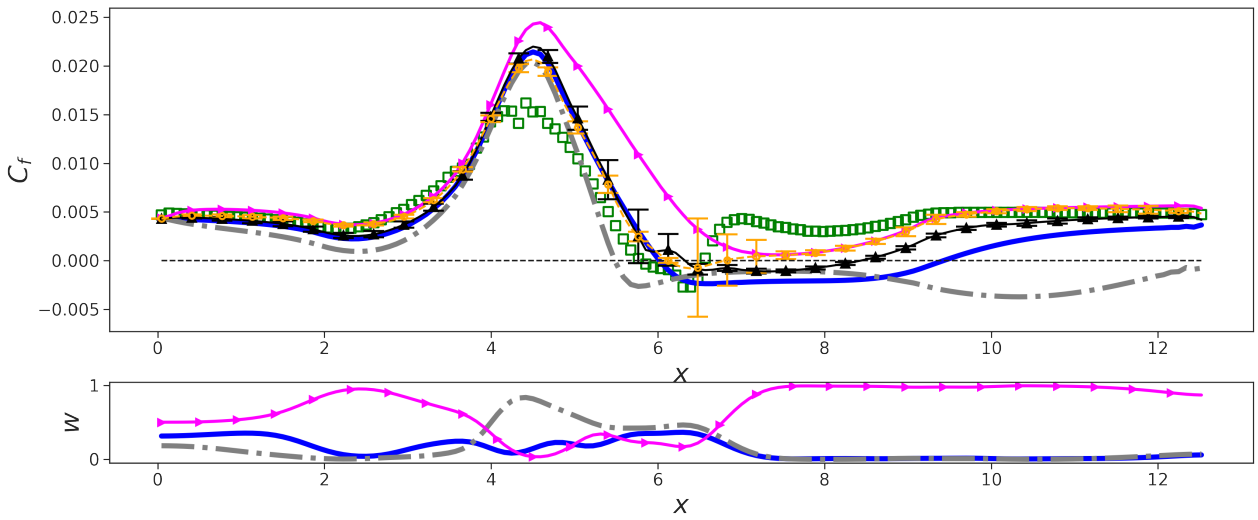


(b) τ_{xy}.

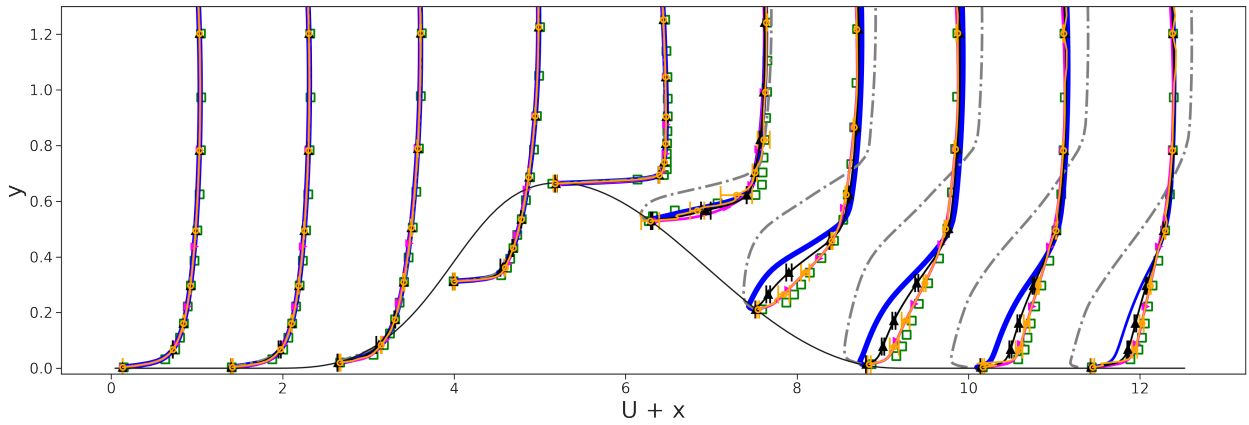
Figure 6.5: Horizontal velocity U and Reynolds shear stresses τ_{xy} at various x/D_{jet} positions for the ANSJ flow case. Baseline $k-\omega$ SST (—); $\mathbf{M}^{(ANSJ)}$ (···); $\mathbf{M}^{(SEP)}$ (→); High-fidelity data (□); Non-intrusive X-MA (-⊕-); Intrusive X-MA (—⊕—).

regressor compared to Figure 5.11, the non-intrusive X-MA does an excellent job of optimally combining the component models. It assigns higher weight to $\mathbf{M}^{(ANSJ)}$ before the onset of separation, then transitions to $\mathbf{M}^{(SEP)}$ immediately after separation. As a result, the separation bubble is very well captured. Downstream of the reattachment point, the non-intrusive X-MA solution follows the $\mathbf{M}^{(SEP)}$ prediction of C_f, which is the closest to high-fidelity data. The intrusive X-MA solution also performs consistently better than individual models, but it is less accurate than the non-intrusive approach. The blended model yields a recirculation bubble of reduced size, separating slightly downstream and reattaching downstream of the high-fidelity position, though still smaller in comparison to the baseline model. We also notice that, while the effect of $\mathbf{M}^{(ANSJ)}$ before the separation point is immediate in the non-intrusive X-MA, this effect or contribution seems to experience a spatial delay in the intrusive paradigm. The effective influence of $\mathbf{M}^{(ANSJ)}$ is also faintly perceived downstream, where the recirculation bubble tends to delay reattachment compared to a classical friction coefficient prediction of the category $\mathbf{M}^{(SEP)}$. Far downstream from the reattachment, the intrusive X-MA solution approaches the non-intrusive X-MA and the high-fidelity data.

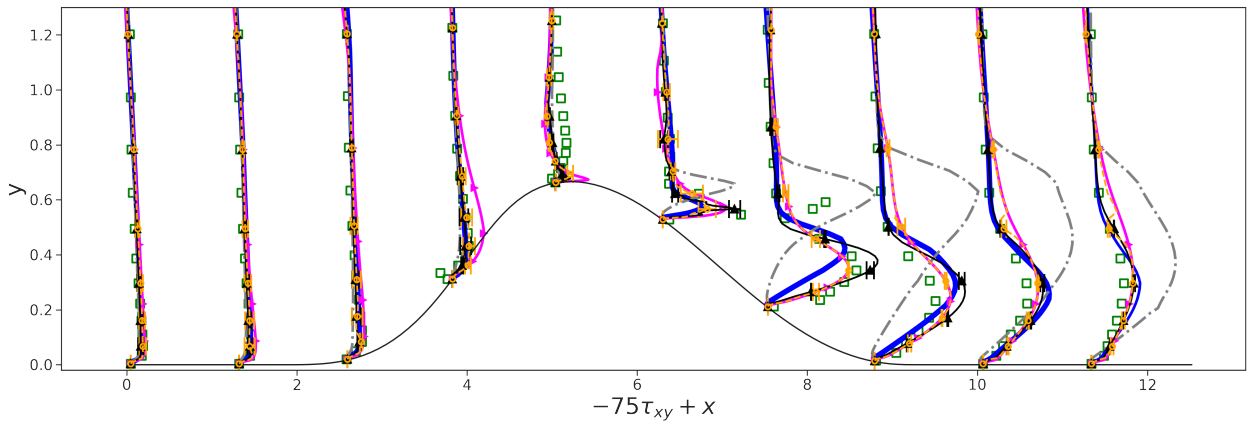
In Figures 6.6b and 6.6c, we present the horizontal velocity and Reynolds shear stress profiles. Here, the non-intrusive X-MA prediction matches remarkably well the high-fidelity data and $\mathbf{M}^{(SEP)}$. The intrusive X-MA prediction is also very good, but again it does not reach the accuracy of intrusive X-MA, due to transport effects. The non-intrusive X-MA prediction of Reynolds shear stress aligns well with both the high-fidelity data and $\mathbf{M}^{(SEP)}$ prediction. The overall performance of the non-intrusive X-MA approach remains very good compared to the baseline. However, in the intrusive X-MA approach, it is not possible to control exactly the effect of the corrections, because the effect of models applied at one point in the flow contaminate other points, especially downstream. For the second and third training cases, the reader can refer to the Appendix D Section D.1, where a similar detailed analysis is presented.



(a) C_f.



(b) $U + x$.



(c) $-75\tau_{xy} + x$.

Figure 6.6: Horizontal velocity U and Reynolds shear stresses τ_{xy} at various x positions for the CD flow case. Baseline $k-\omega$ SST (—); $\mathbf{M}^{(ANSJ)}$ (-·-); $\mathbf{M}^{(SEP)}$ (—►); High-fidelity data (□); Non-intrusive X-MA (-⧗-); Intrusive X-MA (—⧗—).

6.2.2 Prediction of unseen flows

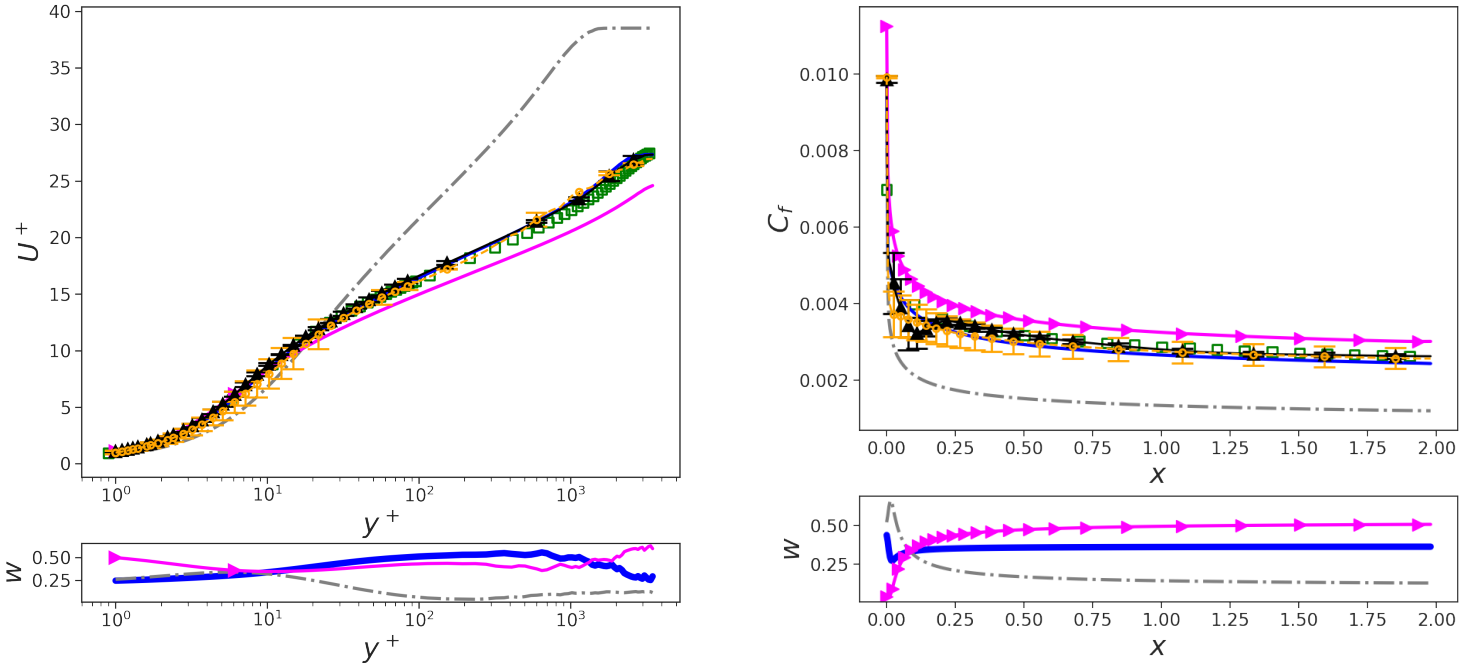
2D Flat Plate

In Figure 6.7a, we present the profiles of U^+ along y^+ at $x = 0.97$, accompanied by the local model weights. Clearly, both intrusive and non-intrusive X-MA predictions exhibit excellent alignment with the high-fidelity data. In the viscous sublayer, the baseline $k - \omega$ SST and $\mathbf{M}^{(ANSJ)}$ are assigned equal weights, while $\mathbf{M}^{(SEP)}$ is locally given double weight. Nevertheless, this adjustment does not compromise the overall results. In the logarithmic and outer regions, a mixture of both $\mathbf{M}^{(SEP)}$ and the baseline model contributes is used.

Figure 6.7b shows the friction coefficient for the same 2DZP case. Once again, both intrusive and non-intrusive X-MA predictions closely match the levels of high-fidelity values. However, in the initial section of the flat plate, several models are activated, with $\mathbf{M}^{(ANSJ)}$ being assigned the highest weight. The interpretation of the weights applied in this region is not straightforward. One possible explanation for this behavior could be the presence of a localized high-pressure gradient near the onset of the boundary layer at $x = 0$, which closely resembles the conditions that trigger $\mathbf{M}^{(ANSJ)}$. Further downstream, a perfect agreement with high-fidelity data for both non-intrusive and intrusive X-MA. This strong concordance underscores the robustness of the X-MA approach in reproducing canonical flows.

Axisymmetric subsonic jet

The second test case is the axisymmetric subsonic jet flow (ASJ). Given its resemblance to the ANSJ training case, we anticipate a similar behavior in this scenario. As depicted in Figure 6.8, the $\mathbf{M}^{(ANSJ)}$ model captures the high-fidelity data very well only in the far jet region. The behaviors of the baseline $k - \omega$ SST and $\mathbf{M}^{(SEP)}$ models are similar to those observed for the ANSJ case. The non-intrusive X-MA prediction in the region between $6D_{jet}$ and $12D_{jet}$ follows mainly the one of the baseline model and the $\mathbf{M}^{(ANSJ)}$, leading to a decrease of velocity along $x-$axis compared to high-fidelity data. However, we recall that when using non-intrusive X-MA with 5 candidate SBL-SpaRTA models (Chapter 5), the predictions in this



(a) U^+.

(b) C_f.

Figure 6.7: U^+ vs. y^+ and C_f vs. x for the 2DZP flow case. Baseline $k-\omega$ SST (—); $\mathbf{M}^{(ANSJ)}$ (···); $\mathbf{M}^{(SEP)}$ (—▸); High-fidelity data (□); Non-intrusive X-MA (-○-); Intrusive X-MA (—▲—).

intermediate regions show significant improvement, thanks to the contributions of many models (Figure 5.13). This raises the question of determining the ideal number of candidate models to employ in order to attain optimal accuracy without incurring excessive computational costs in terms of the number of simulations to be run. In addition, Figure 6.2b reveals that the GPR-predicted weights in the intermediate region between the near and far jet regions do not correspond to the optimal weights based on the reference solution, which may also explain the decreased accuracy in this case compared to the results of Chapter 5 (Figures 5.13 and 5.14). Moreover, the intrusive X-MA solution shows a certain latency in its response to the baseline model local weights because of transport of the correction, resulting in a closer alignment to the $\mathbf{M}^{(ANSJ)}$ model in this intermediate region. Between the ANSJ training case and the ASJ test case, a notable difference is the Mach number, which is approximately reduced by a factor of 5. Despite this substantial change, the X-MA approach continues to yield satisfactory results. Notably, the correct spreading of the jet in the far jet region is maintained at a satisfactory level, significantly outperforming the predictions obtained from the baseline $k-\omega$ SST model. The consistency in performance improvements is further confirmed by the data shown in Figure 6.9, where the horizontal velocity and Reynolds shear stresses predicted by both the intrusive and non-intrusive X-MA methods remain closely aligned with high-fidelity data.

2D Wall-mounted hump

Moving on to the NASA 2D Wall-Mounted Hump, this specific case serves as a representative example of separated flow scenarios. It operates at a significantly higher Reynolds number (80×10^6) when compared to the training flow cases that feature relatively moderate Reynolds numbers (around 10^4). Therefore, this test case entails an extrapolation in both geometry and Reynolds number.

The pressure and friction coefficients are displayed in Figure 6.10, along with the model weights across the wall. The regions preceding separation are predominantly influenced by the baseline model, while the $\mathbf{M}^{(SEP)}$ and $\mathbf{M}^{(ANSJ)}$ models are assigned similar moderate weights. In this case, this benefits to the intrusive X-MA prediction, where the locally received turbulent

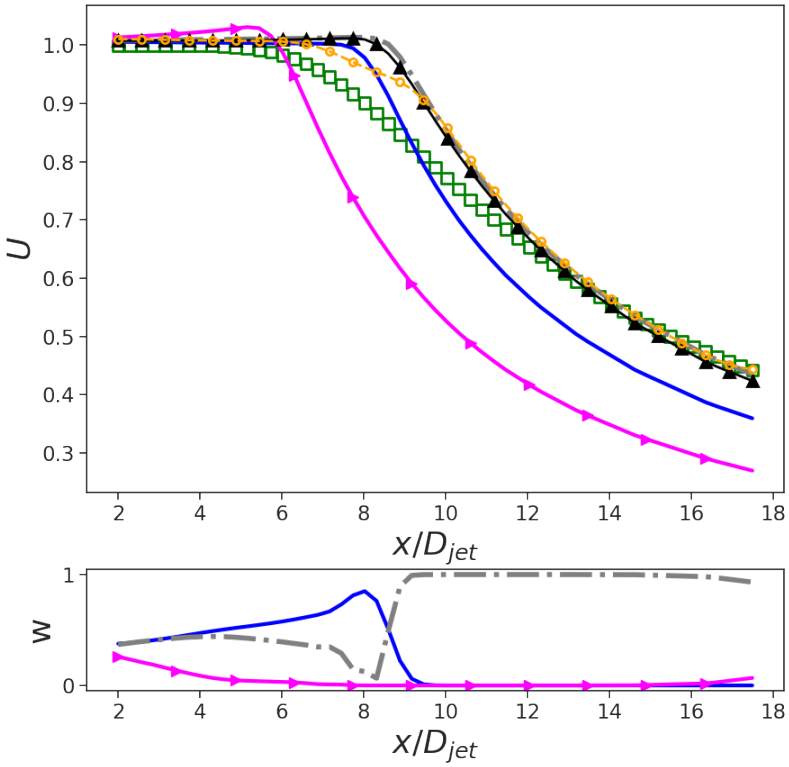
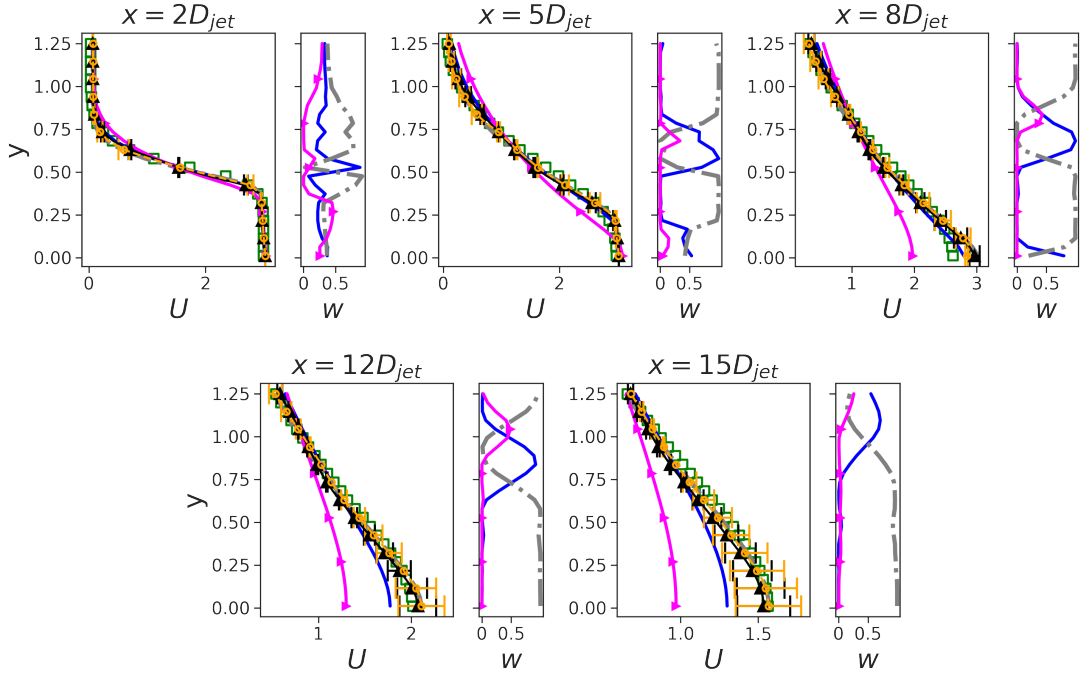
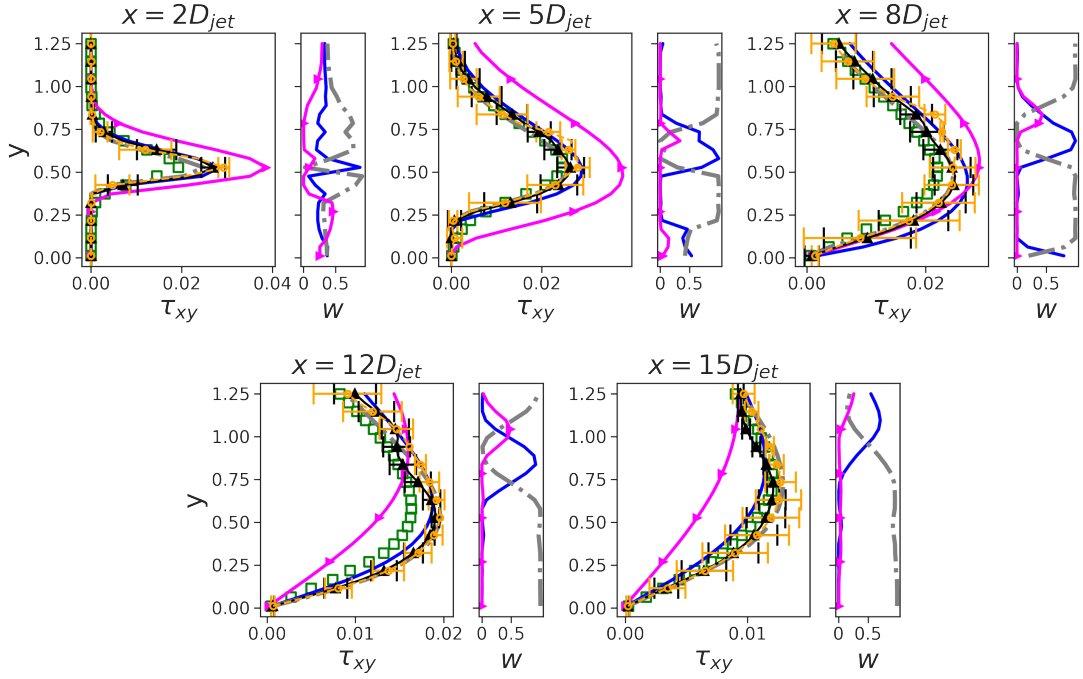


Figure 6.8: U vs. x/D_{jet} along the axisymmetric horizontal axis and the corresponding model weights w for the ASJ flow case. Baseline $k-\omega$ SST (—); $\mathbf{M}^{(ANSJ)}$ (-·-·-); $\mathbf{M}^{(SEP)}$ (—►—); High-fidelity data (□); Non-intrusive X-MA (-⊕-); Intrusive X-MA (—▲—).

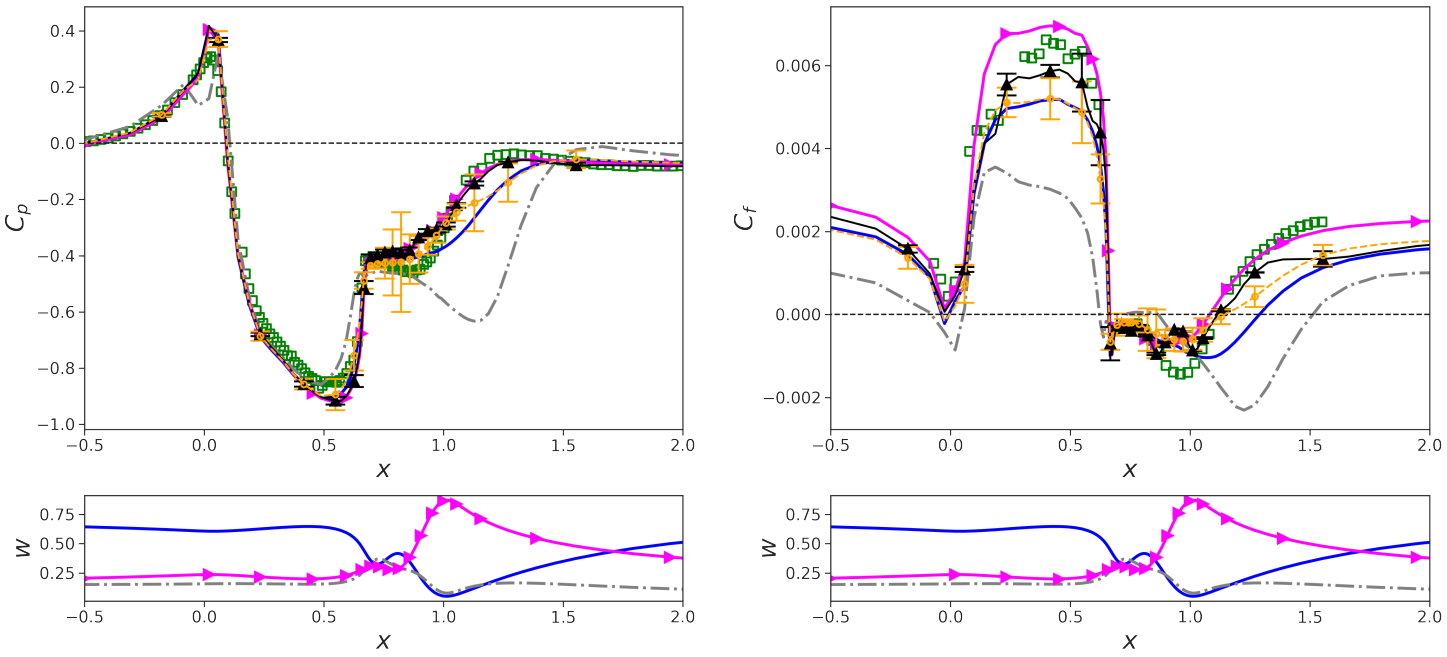


(a) U.



(b) τ_{xy}.

Figure 6.9: Horizontal velocity U and Reynolds shear stresses τ_{xy} at various x/D_{jet} positions for the ASJ flow case. Baseline $k-\omega$ SST (—); $\mathbf{M}^{(ANSJ)}$ (-·-·-); $\mathbf{M}^{(SEP)}$ (—►—); High-fidelity data (□); Non-intrusive X-MA (-⊕-); Intrusive X-MA (—▲—).



(a) C_p. (b) C_f.

Figure 6.10: Pressure coefficient C_p and friction coefficient C_f along x axis for the WMH flow case. Baseline $k-\omega$ SST (—); $\mathbf{M}^{(ANSJ)}$ (···); $\mathbf{M}^{(SEP)}$ (—▸); High-fidelity data (□); Non-intrusive X-MA (-⊥-); Intrusive X-MA (—⊥—).

kinetic energy production is improved.

Beyond the reattachment point, the $\mathbf{M}^{(SEP)}$ correction becomes predominant. Consequently, both intrusive and non-intrusive X-MA predictions of skin friction accurately depict the size and location of the recirculation bubble. Moving further downstream, the baseline model regains its influence, while the impact of $\mathbf{M}^{(SEP)}$ diminishes at the expense of the baseline model. This decrease in $\mathbf{M}^{(SEP)}$'s contribution is not immediately reflected in the intrusive X-MA predictions, primarily due to transport effects.

The profiles of horizontal velocity and Reynolds shear stress at various stations along the hump are presented in Figure 6.11. Figure 6.11a shows that both the non-intrusive and intrusive X-MA predictions closely mirror the high-fidelity velocity profiles. In Figure 6.11b, both X-

MA paradigms exhibit a slightly reduced performance in comparison to $\mathbf{M}^{(SEP)}$ alone but still clearly outperform the baseline model. Overall, these observations provide confidence in the X-MA ability to enhance predictive accuracy for unseen flow cases.

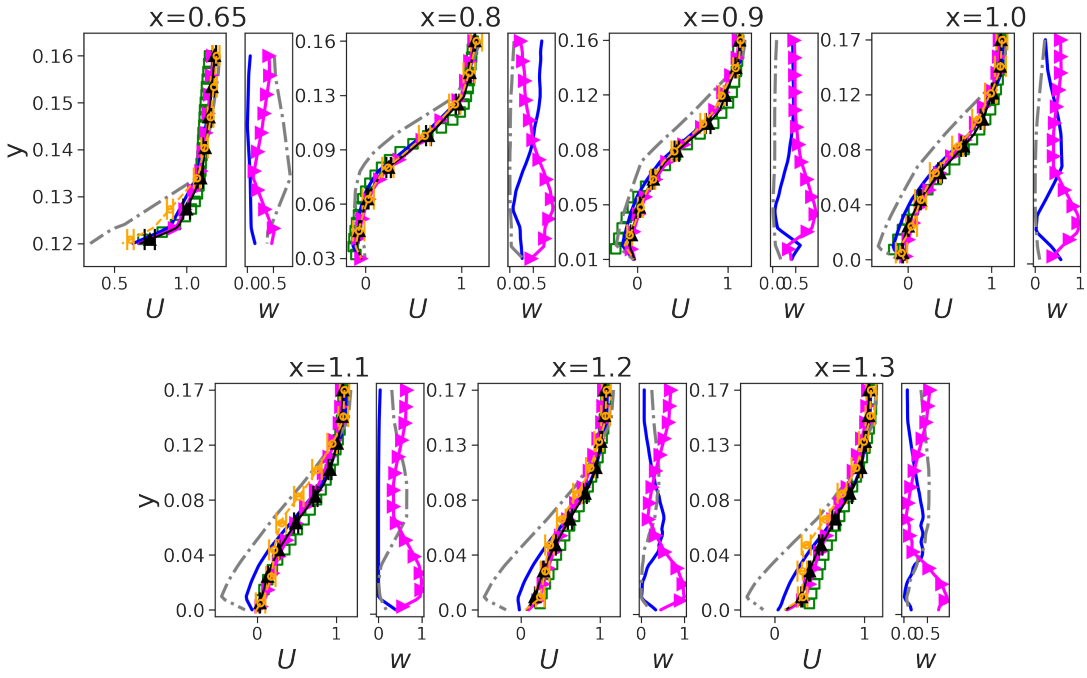
To conclude this analysis, the performance of both intrusive and non-intrusive paradigms on the test cases is quantitatively assessed in Table 6.4 using the $Imp(\%)$ metric. Both non-intrusive and intrusive approaches improve the baseline model. The advantage of one over the other is not very clear-cut. The intrusive X-MA consistently shows a notable level of improvement across all unseen test cases, which positions it as a promising candidate for future developments.

case	QoI	Non-intrusive X-MA	Intrusive X-MA	$\mathbf{M}^{(CHAN)}$	$\mathbf{M}^{(ANSJ)}$	$\mathbf{M}^{(SEP)}$
2DZP	U^+	0.3	3.6	0	-3304.1	-117.2
	C_f	5.7	9.7	0	-469.2	-21.2
ASJ	U	71.5	56.5	0	62.7	-650.7
	τ_{xy}	32.4	55.6	0	56.2	-562.9
2DWMH	U	8.1	45.4	0	-251.3	46.7
	τ_{xy}	50.6	45.8	0	-38.9	75.8
	C_f	27.0	64.2	0	-441.2	67.6
	C_p	29.1	13.4	0	-498.2	16.2

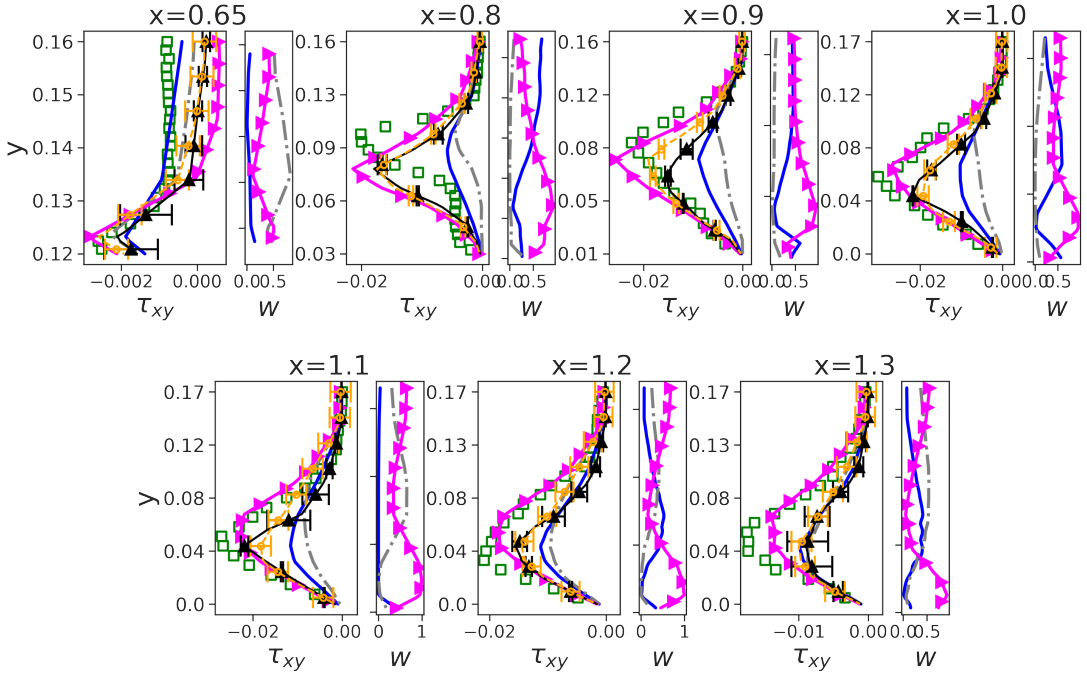
Table 6.4: Improvements in (%) wrt the baseline k-ω SST on test cases

6.3 Conclusions

In this chapter, we introduced an alternative, intrusive, formulation of X-MA consisting in mixing different data-driven model corrections within the flow solver, i.e. in using data-driven weighting functions to build a blended SBL-SpaRTA model. The intrusive approach was systematically compared to the intrusive X-MA methodology of Chapter 5. Both non-intrusive and intrusive X-MA were able to improve over the baseline $k-\omega$ SST, and were sometimes locally sometimes better than the customized model prediction. The intrusive X-MA could not achieve the same accuracy as the intrusive one because of transport of the local corrections to downstream locations in the flow. However, the intrusive X-MA is cheaper than



(a) U.



(b) τ_{xy}.

Figure 6.11: Horizontal velocity U and Reynolds shear stresses τ_{xy} at various x positions for the WMH flow case. Baseline $k-\omega$ SST (—); $\mathbf{M}^{(ANSJ)}$ (-·-); $\mathbf{M}^{(SEP)}$ (—▸); High-fidelity data (□); Non-intrusive X-MA (-⊕-); Intrusive X-MA (—▲—).

the non-intrusive one, because it requires less RANS simulations (a single RANS is needed if the uncertainty in the model parameters is neglected). Additionally, the intrusive X-MA satisfies the governing equations, contrary to the non-intrusive one. As a consequence it provides smooth solutions, while the external model aggregation can sometimes originate wiggles or stepcase solutions.

In summary, both intrusive and non-intrusive X-MA methodologies yield satisfactory results for the considered set of flows. The intrusive X-MA is probably more easily acceptable for the community of RANS users that are not interested in uncertainty quantification.

Chapter 7

Conclusions and perspectives

Contents

7.1 Summary

This PhD thesis aims to advance the current state-of-the-art in RANS turbulence modeling using machine learning. The first part of the thesis explores the Sparse Bayesian Learning (SBL) algorithm as an effective tool for sparse Bayesian regression and cost-effective uncertainty quantification. The application of this algorithm leads to the development of sparse stochastic EARSM-type closures, referred to as SBL-SpaRTA models. A systematic training procedure is detailed and applied to generate models for a subset of turbulent separated flows where the conventional $k-\omega$ SST model has limitations in predicting recirculating bubble dimensions. The resulting models are interpretable and Galilean frame invariant. In addition, the stochastic nature of these models replaces traditional deterministic coefficients with Gaussian probability distributions, with mean and variance inferred from pre-processed high-fidelity data. This approach also allows uncertainty quantification for different QoI and can be used to perform sensitivity analysis for each of the correction terms.

A drawback of the customized SBL-SpaRTA models is that although such models are very

effective in improving the prediction of recirculation zones, they also modify upstream and downstream regions where the conventional $k-\omega$ SST model typically performs well. Furthermore, the SBL-SpaRTA models show suboptimal performance when dealing with flow scenarios that do not belong to the class of flows used for training, indicating a lack of generalization.

In response to these limitations, the SBL-SpaRTA framework was used to train models on a variety of turbulent flow cases, including turbulent flat plates with varying pressure gradients, turbulent separated flows, and jet flows. Using the same methodology described in the initial contribution, this training phase led to the generation of sparse and interpretable SBL-SpaRTA-type corrections. These corrections allowed us to gain a deeper understanding of the shortcomings of the baseline $k-\omega$ SST in different flow categories, and to pinpoint areas where improvements were necessary. In addition, we were able to evaluate the additional value provided by our customized corrections.

Then, building on the Mixture-of-Experts concept, we proceeded to learn local weighting functions for each model based on its performance in predicting velocity data. The weighting functions reflect local regions of better performance for each model under consideration. However, weighting functions that rely solely on spatial coordinates cannot be generalized to another flow. To overcome this limitation, we introduced a method to link model weights to local flow physics. This was done by training Random Forest Regressors on the local model weights, using locally computed physical flow features derived from literature and domain knowledge. The weighting functions were then used to aggregate the predictions of different data-driven models according to a "non-intrusive" procedure called X-MA. The X-MA prediction consistently improved over the baseline model across the flow domain, both for training flow cases and for unseen flow scenarios. However, the non-intrusive X-MA itself is not a turbulence model, but rather an uncertainty quantification methodology applied as post-processing to a set of competing models.

The last chapter focused on the development of an intrusive X-MA procedure, which corresponds to the generation of a blended turbulence model by applying a weighted combination of data-driven corrections to the baseline model based on a set of local flow features. The blended

model is constructed as a spatial convex linear combination of the previously developed SBL-SpaRTA models. To enable a comparative analysis between the intrusive and non-intrusive methods, we introduce a systematic procedure for the optimal selection of hyperparameters governing the model weights and the identification of the most appropriate QoI for their formulation. In intrusive X-MA, although data-driven corrections are applied locally, their effects are transported by convection and diffusion, causing delays in the application/removal of a given correction in a given flow region. This does not happen in non-intrusive X-MA, where the results of different component models are statically blended a posteriori. As a result, intrusive X-MA is slightly less accurate than non-intrusive X-MA. As a counterpart, the intrusive X-MA results are a solution of the governing equations, which is not true for the non-intrusive solution, the latter corresponding more to an uncertainty quantification method than to a turbulence model. In addition, intrusive X-MA provides smoother solutions and is less computationally demanding, resulting in a single RANS simulation if the uncertainties in the model parameters are neglected.

7.2 Perspectives

The present research work offers multiple perspectives for future studies.

The SBL-SpaRTA models are trained *a priori*. This requires full-field high-fidelity data generated through the k-corrective frozen procedure. This in turn requires high-fidelity data of velocity and Reynolds stress over the entire computational domain, which is not always possible. As a follow-up to the present study, we plan to develop a CFD-in-the-loop training procedure (in the spirit of [71]), accelerated by ML surrogates of the CFD solver response. Such procedure is more costly, but i) it avoids feature mismatch problems between the training and prediction settings; ii) suppresses the need for full-field data and sets the stage for learning models directly from sparse (e.g. experimental) observations.

Another path for improvement is to go beyond Pope's Reynolds stress representation. Pope's theory assumes that the nonlinear anisotropy depends entirely on the velocity gradient tensor.

However, this assumption implies local turbulence equilibrium, which is not satisfied in situations characterized by rapid distortions within the flow. An avenue for progress could be to extend the SBL-SpaRTA framework to a much more comprehensive representation, relying on the sparsity-promoting SBL procedure to select the most relevant features.

Finally, the SBL-SpaRTA needs to be extended and assessed for 3D flows. Note that the deterministic SpaRTA approach has already been successfully applied to 3D flows in [64].

The model aggregation techniques investigated in this study have shown promise in improving the generalizability of data-driven RANS models. However, the weighting functions rely on a specific, heuristic choice of a set of local flow features, here extracted from previous work by Ling and Templeton [52]. More efficient feature choices are possible, and future work should focus on feature engineering, normalization, and selection.

In addition, the choice of training data was empirical and dictated by common sense. A more systematic study of data selection that maximizes the amount of information injected into the learning algorithm is worthy of future work. For example, here we trained component SBL-SpaRTA models on arbitrarily chosen model "classes". However, such flows may contain several concurrent physical processes (equilibrium boundary layers, wakes, pressure gradient regions, separated flow regions, 3D regions...). The customized model then tends to correct the baseline model "on average" for all such processes. Using modern clustering techniques, the different physical processes can be extracted from each flow dataset. Data corresponding to the same cluster in different flows could be aggregated and used to train a "process-specific" data-driven correction. This should lead to more accurate results than the current simpler approach, provided an appropriate clustering algorithm can be found.

Finally, the weighting functions used in this study were trained a priori and the input features were based on the baseline model. Again, model-consistent training of the blending functions would allow the updated feature to be taken into account and lead to more accurate results.

As a main take-away message, we hope to have convinced the reader that training and aggregating models tailored to specific tasks based on local flow features shows promise for

consistently improving RANS predictions over a range of flows, while keeping the individual data-driven contributions simple and interpretable. Further efforts in this direction may help to move towards generalizable data-driven RANS models.

Bibliography

[1] P. Spalart, "Strategies for turbulence modelling and simulations," *International Journal of Heat and Fluid Flow*, vol. 21, pp. 252–263, 2000.

[2] P. Spalart, "Philosophies and fallacies in turbulence modeling," *Progress in Aerospace Sciences*, vol. 74, pp. 1–15, 2015.

[3] P. A. Durbin, "Some recent developments in turbulence closure modeling," *Annual Review of Fluid Mechanics*, vol. 50, pp. 77–103, 2018.

[4] S. B. Pope, *Turbulent flows*. Cambridge university press, 2000.

[5] D. C. Wilcox, *Turbulence modeling for CFD*, 3rd ed. DCW Industries, 2006.

[6] J. L. Lumley, "Toward a turbulent constitutive relation," *Journal of Fluid Mechanics*, vol. 41, no. 2, pp. 413–434, 1970.

[7] A. N. Kolmogorov, "Equations of turbulent motion in an incompressible fluid," in *Dokl. Akad. Nauk SSSR*, vol. 30, 1941, pp. 299–303.

[8] M. Saiy, "An experimental and computational investigation of turbulence in plane two-stream mixing layers with various levels of turbulence," 1974.

[9] D. B. Spalding, *Mathematical models of turbulent transport processes*. Imperial College of Science and Technology, Department of Mechanical Engineering, 1978.

[10] D. C. Wilcox, "Reassessment of the scale-determining equation for advanced turbulence models," *AIAA journal*, vol. 26, no. 11, pp. 1299–1310, 1988.

[11] C. G. Speziale, R. Abid, and E. C. Anderson, "Critical evaluation of two-equation models for near-wall turbulence," *AIAA journal*, vol. 30, no. 2, pp. 324–331, 1992.

[12] F. R. Menter, "Improved two-equation k-omega turbulence models for aerodynamic flows," Tech. Rep., 1992.

[13] P. Spalart and S. Allmaras, "A one-equation turbulence model for aerodynamic flows," in *30th aerospace sciences meeting and exhibit*, 1992, p. 439.

[14] F. G. Schmitt, "About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity," *Comptes Rendus Mécanique*, vol. 335, pp. 617–627, 2007.

[15] P. Spalart and M. Shur, "On the sensitization of simple turbulence models to rotation and curvature," *Aerospace Science and Technology*, vol. 1, pp. 297–302, 1997.

[16] C. Speziale, "On nonlinear k-l and k-ϵ models of turbulence," *Journal of Fluid Mechanics*, vol. 178, pp. 459–475, 1987.

[17] P. Durbin, "Near-wall turbulence closure modelling without damping functions," *Theoretical and Computational Fluid Dynamics*, vol. 3, pp. 1–13, 1991.

[18] W. Rodi, "A new algebraic relation for calculating the Reynolds stresses," *Gesellschaft Angewandte Mathematik und Mechanik Workshop Paris France*, vol. 56, p. 219, Mar. 1976.

[19] S. B. Pope, "A more general effective-viscosity hypothesis," *Journal of Fluid Mechanics*, vol. 72, no. 2, pp. 331–340, 1975.

[20] T. B. Gatski and C. G. Speziale, "On explicit algebraic stress models for complex turbulent flows," *Journal of Fluid Mechanics*, vol. 254, pp. 59–78, 1993.

[21] S. Wallin and A. V. Johansson, "An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows," *Journal of Fluid Mechanics*, vol. 403, pp. 89–132, 2000.

[22] C. Speziale, "A review of Reynolds Stress models for Turbulent shear flows," NASA, Tech. Rep. NASA-CR-195054, 1995.

[23] F. Menter, A. Garbaruk, and Y. Egorov, "Explicit algebraic reynolds stress models for anisotropic wall-bounded flows," *Progress in flight physics*, vol. 3, pp. 89–104, 2012.

[24] H. Xiao and P. Cinnella, "Quantification of model uncertainty in RANS simulations: A review," *Progress in Aerospace Sciences*, vol. 108, pp. 1–31, 2019.

[25] K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," *Annual Review of Fluid Mechanics*, vol. 51, pp. 357–377, 2019.

[26] K. Duraisamy, "Perspectives on machine learning-augmented Reynolds-Averaged and Large Eddy Simulation models of turbulence," *Physical Review Fluids*, vol. 6, p. 050504, 2021.

[27] R. D. Sandberg and Y. Zhao, "Machine-learning for turbulence and heat-flux model development: A review of challenges associated with distinct physical phenomena and progress to date," *International Journal of Heat and Fluid Flow*, vol. 95, p. 108983, 2022.

[28] M. Emory, J. Larsson, and G. Iaccarino, "Modeling of structural uncertainties in Reynolds-Averaged Navier–Stokes closures," *Physics of Fluids*, vol. 25, no. 11, p. 110822, 2013.

[29] C. Gorlé and G. Iaccarino, "A framework for epistemic uncertainty quantification of turbulent scalar flux models for Reynolds-Averaged Navier-Stokes simulations," *Physics of Fluids*, vol. 25, no. 5, p. 055105, 2013.

[30] R. L. Thompson, A. A. Mishra, G. Iaccarino, W. Edeling, and L. Sampaio, "Eigenvector perturbation methodology for uncertainty quantification of turbulence models," *Physical Review Fluids*, vol. 4, no. 4, p. 044603, 2019.

[31] P. D. A. Platteeuw, G. J. A. Loeven, and H. Bijl, "Uncertainty quantification applied to the k–ϵ model of turbulence using the probabilistic collocation method," in *10th AIAA Non-Deterministic Approaches Conference*, 2008, paper no.: 2008-2150.

[32] S. H. Cheung, T. A. Oliver, E. E. Prudencio, S. Prudhomme, and R. D. Moser, "Bayesian uncertainty analysis with applications to turbulence modeling," *Reliability Engineering & System Safety*, vol. 96, no. 9, pp. 1137–1149, 2011.

[33] W. N. Edeling, P. Cinnella, R. P. Dwight, and H. Bijl, "Bayesian estimates of parameter variability in the k–ε turbulence model," *Journal of Computational Physics*, vol. 258, pp. 73–94, 2014.

[34] L. Margheri, M. Meldi, M. Salvetti, and P. Sagaut, "Epistemic uncertainties in RANS model free coefficients," *Computers & Fluids*, vol. 102, pp. 315–335, 2014.

[35] S. V. Poroseva, M. Y. Hussaini, and S. L. Woodruff, "Improving the predictive capability of turbulence models using evidence theory," *AIAA Journal*, vol. 44, no. 6, pp. 1220–1228, 2006.

[36] W. N. Edeling, P. Cinnella, and R. P. Dwight, "Predictive RANS simulations via Bayesian model-scenario averaging," *Journal of Computational Physics*, vol. 275, pp. 65–91, 2014.

[37] W. N. Edeling, M. Schmelzer, R. P. Dwight, and P. Cinnella, "Bayesian predictions of Reynolds-Averaged Navier-Stokes uncertainties using maximum a posteriori estimates," *AIAA Journal*, vol. 56, no. 5, pp. 2018–2029, 2018.

[38] M. de Zordo-Banliat, X. Merle, G. Dergham, and P. Cinnella, "Bayesian model-scenario averaged predictions of compressor cascade flows under uncertain turbulence models," *Computers & Fluids*, vol. 201, p. 104473, 2020.

[39] M. de Zordo-Banliat, X. Merle, G. Dergham, and P. Cinnella, "Estimates of turbulence modeling uncertainties in NACA65 cascade flow predictions by Bayesian model-scenario

averaging," *International Journal of Numerical Methods for Heat & Fluid Flow*, no. 4, pp. 1398–1414, 2020.

[40] B. Tracey, K. Duraisamy, and J. Alonso, "Application of supervised learning to quantify uncertainties in turbulence and combustion modeling," in *51st AIAA Aerospace Sciences Meeting*, 2013, Dallas, TX, paper 2013-0259.

[41] E. J. Parish and K. Duraisamy, "A paradigm for data-driven predictive modeling using field inversion and machine learning," *Journal of Computational Physics*, vol. 305, pp. 758–774, 2016.

[42] A. Ferrero, A. Iollo, and F. Larocca, "Field inversion for data-augmented RANS modelling in turbomachinery flows," *Computers & Fluids*, vol. 201, p. 104474, 2020.

[43] P. S. Volpiani, M. Meyer, L. Franceschini, J. Dandois, F. Renac, E. Martin, O. Marquet, and D. Sipp, "Machine learning-augmented turbulence modeling for rans simulations of massively separated flows," *Physical Review Fluids*, vol. 6, no. 6, p. 064607, 2021.

[44] P. S. Volpiani, R. F. Bernardini, and L. Franceschini, "Neural network-based eddy-viscosity correction for rans simulations of flows over bi-dimensional bumps," *International Journal of Heat and Fluid Flow*, vol. 97, p. 109034, 2022.

[45] K. Duraisamy, "Perspectives on machine learning-augmented reynolds-averaged and large eddy simulation models of turbulence," *Physical Review Fluids*, vol. 6, no. 5, p. 050504, 2021.

[46] V. Srivastava and K. Duraisamy, "Generalizable physics-constrained modeling using learning and inference assisted by feature-space engineering," *Physical Review Fluids*, vol. 6, no. 12, p. 124602, 2021.

[47] C. L. Rumsey, G. N. Coleman, and L. Wang, "In search of data-driven improvements to rans models applied to separated flows," in *AIAA Scitech 2022 Forum*, 2022, p. 0937.

[48] H. Xiao, J.-L. Wu, J.-X. Wang, R. Sun, and C. Roy, "Quantifying and reducing model-form uncertainties in Reynolds-Averaged Navier-Stokes simulations: A data-driven, physics-informed Bayesian approach," *Journal of Computational Physics*, vol. 324, pp. 115–136, 2016.

[49] J.-L. Wu, J.-X. Wang, and H. Xiao, "A Bayesian calibration–prediction method for reducing model-form uncertainties with application in RANS simulations," *Flow, Turbulence and Combustion*, vol. 97, no. 3, pp. 761–786, 2016.

[50] J.-L. Wu, C. Michelén-Ströfer, and H. Xiao, "Physics-informed covariance kernel for model-form uncertainty quantification with application to turbulent flows," *Computers & Fluids*, vol. 193, p. 104292, 2019.

[51] W. N. Edeling, G. Iaccarino, and P. Cinnella, "Data-free and data-driven RANS predictions with quantified uncertainty," *Flow, Turbulence and Combustion*, vol. 100, no. 3, pp. 593–616, 2018.

[52] J. Ling and J. Templeton, "Evaluation of machine learning algorithms for prediction of regions of high Reynolds-Averaged Navier-Stokes uncertainty," *Physics of Fluids*, vol. 27, no. 8, p. 085103, 2015.

[53] J.-L. Wu, J.-X. Wang, H. Xiao, and J. Ling, "A priori assessment of prediction confidence for data-driven turbulence modeling," *Flow, Turbulence and Combustion*, vol. 99, no. 1, pp. 25–46, 2017.

[54] J.-L. Wu, H. Xiao, and E. Paterson, "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," *Physical Review Fluids*, vol. 3, no. 7, p. 074602, 2018.

[55] C. A. M. Ströfer and H. Xiao, "End-to-end differentiable learning of turbulence models from indirect observations," *Theoretical and Applied Mechanics Letters*, vol. 11, no. 4, p. 100280, 2021.

[56] X.-H. Zhou, J. Han, and H. Xiao, "Frame-independent vector-cloud neural network for nonlocal constitutive modeling on arbitrary grids," *Computer Methods in Applied Mechanics and Engineering*, vol. 388, p. 114211, 2022.

[57] M. L. Kaandorp and R. P. Dwight, "Data-driven modelling of the reynolds stress tensor using random forests with invariance," *Computers & Fluids*, vol. 202, p. 104497, 2020.

[58] C. Jiang, R. Vinuesa, R. Chen, J. Mi, S. Laima, and H. Li, "An interpretable framework of data-driven turbulence modeling using deep neural networks," *Physics of Fluids*, vol. 33, no. 5, p. 055133, 2021.

[59] X. He, J. Tan, G. Rigas, and M. Vahdati, "On the explainability of machine-learning-assisted turbulence modeling for transonic flows," *International Journal of Heat and Fluid Flow*, vol. 97, p. 109038, 2022.

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

[61] M. Schmelzer, R. P. Dwight, and P. Cinnella, "Discovery of algebraic Reynolds-stress models using sparse symbolic regression," *Flow, Turbulence and Combustion*, vol. 104, no. 2, pp. 579–603, 2020.

[62] S. Beetham and J. Capecelatro, "Formulating turbulence closures using sparse regression with embedded form invariance," *Physical Review Fluids*, vol. 5, no. 8, p. 084611, 2020.

[63] H. Mandler and B. Weigand, "A realizable and scale-consistent data-driven non-linear eddy viscosity modeling framework for arbitrary regression algorithms," *International Journal of Heat and Fluid Flow*, vol. 97, p. 109018, 2022.

[64] J. P. Huijing, R. P. Dwight, and M. Schmelzer, "Data-driven RANS closures for three-dimensional flows around bluff bodies," *Computers & Fluids*, vol. 225, p. 104997, 2021. [Online]. Available: https://doi.org/10.1016/j.compfluid.2021.104997

[65] S. Beetham, R. Fox, and J. Capecelatro, "Sparse identification of multiphase turbulence closures for coupled fluid–particle flows," *Journal of Fluid Mechanics*, vol. 914, p. A11, 2021.

[66] H. Li, Y. Zhao, J. Wang, and R. D. Sandberg, "Data-driven model development for Large-Eddy Simulation of turbulence using Gene-Expression Programming," *Physics of Fluids*, vol. 33, no. 12, p. 125127, 2021.

[67] J. Steiner, R. P. Dwight, and A. Viré, "Data-driven RANS closures for wind turbine wakes under neutral conditions," *Computers & Fluids*, vol. 233, p. 105213, 2022.

[68] M. Wang, C. Chen, and W. Liu, "Establish algebraic data-driven constitutive models for elastic solids with a tensorial sparse symbolic regression method and a hybrid feature selection technique," *Journal of the Mechanics and Physics of Solids*, vol. 159, p. 104742, 2022.

[69] X. Xu, F. Waschkowski, A. S. Ooi, and R. D. Sandberg, "Towards robust and accurate Reynolds-Averaged closures for natural convection via multi-objective CFD-driven machine learning," *International Journal of Heat and Mass Transfer*, vol. 187, p. 122557, 2022.

[70] Y. Zhao, H. D. Akolekar, J. Weatheritt, V. Michelassi, and R. D. Sandberg, "RANS turbulence model development using CFD-driven machine learning," *Journal of Computational Physics*, vol. 411, p. 109413, 2020.

[71] I. B. H. Saïdi, M. Schmelzer, P. Cinnella, and F. Grasso, "CFD-driven symbolic identification of algebraic reynolds-stress models," *Journal of Computational Physics*, vol. 457, p. 111037, 2022.

[72] X. He, F. Zhao, and M. Vahdati, "A turbo-oriented data-driven modification to the spalart–allmaras turbulence model," *Journal of Turbomachinery*, vol. 144, no. 12, p. 121007, 2022.

[73] F. Menter, R. Lechner, and A. Matyushenko, "Best practice: generalized k-ω two-equation turbulence model in ansys cfd (geko)," *ANSYS Germany GmbH*, 2019.

[74] R. Matai and P. Durbin, "Zonal eddy viscosity models based on machine learning," *Flow, Turbulence and Combustion*, vol. 103, pp. 93–109, 2019.

[75] J. Ho, N. Pepper, and T. Dodwell, "Probabilistic machine learning to improve generalisation of data-driven turbulence modelling," *arXiv preprint arXiv:2301.09443*, 2023.

[76] A. Lozano-Durán and H. J. Bae, "Machine learning building-block-flow wall model for large-eddy simulation," *Journal of Fluid Mechanics*, vol. 963, p. A35, 2023.

[77] M. deZordo Banliat, X. Merle, G. Dergham, and P. Cinnella, "Bayesian model-scenario averaged predictions of compressor cascade flows under uncertain turbulence models," *Computers & Fluids*, vol. 201, p. 104473, 2020.

[78] M. de Zordo-Banliat, X. Merle, G. Dergham, and P. Cinnella, "Estimates of turbulence modeling uncertainties in naca65 cascade flow predictions by bayesian model-scenario averaging," *International Journal of Numerical Methods for Heat & Fluid Flow*, 2022.

[79] D. Draper, "Assessment and propagation of model uncertainty," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 1, pp. 45–70, 1995.

[80] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: a tutorial (with comments by m. clyde, david draper and ei george, and a rejoinder by the authors," *Statistical science*, vol. 14, no. 4, pp. 382–417, 1999.

[81] G. Stoltz, "Agrégation séquentielle de prédicteurs: méthodologie générale et applications à la prévision de la qualité de l'air et à celle de la consommation électrique," *Journal de la Société française de Statistique*, vol. 151, no. 2, pp. 66–106, 2010.

[82] M. Devaine, P. Gaillard, Y. Goude, and G. Stoltz, "Forecasting electricity consumption by aggregating specialized experts: A review of the sequential aggregation of specialized

experts, with an application to slovakian and french country-wide one-day-ahead (half-) hourly predictions," *Machine Learning*, vol. 90, pp. 231–260, 2013.

[83] R. Deswarte, V. Gervais, G. Stoltz, and S. Da Veiga, "Sequential model aggregation for production forecasting," *Computational Geosciences*, vol. 23, pp. 1107–1124, 2019.

[84] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty Years of Mixture of Experts," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[85] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994.

[86] M. de Zordo-Banliat, G. Dergham, X. Merle, and P. Cinnella, "Space-dependent turbulence model aggregation using machine learning," *arXiv preprint arXiv:2301.09013*, 2023.

[87] B. E. Launder and B. I. Sharma, "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc," *Letters in heat and mass transfer*, vol. 1, no. 2, pp. 131–137, 1974.

[88] F. R. Menter, "Two-equation eddy-viscosity turbulence models for engineering applications," *AIAA journal*, vol. 32, no. 8, pp. 1598–1605, 1994.

[89] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in physics*, vol. 12, no. 6, pp. 620–631, 1998.

[90] R. D. Moser, J. Kim, and N. N. Mansour, "Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$," *Physics of fluids*, vol. 11, no. 4, pp. 943–945, 1999.

[91] M. Lee and R. D. Moser, "Direct numerical simulation of turbulent channel flow up to $Re_\tau = 5200$," *Journal of fluid mechanics*, vol. 774, pp. 395–415, 2015.

[92] J. Kim, P. Moin, and R. Moser, "Turbulence statistics in fully developed channel flow at low reynolds number," *Journal of fluid mechanics*, vol. 177, pp. 133–166, 1987.

[93] P. Schlatter and R. Örlü, "Assessment of direct numerical simulation data of turbulent boundary layers," *Journal of Fluid Mechanics*, vol. 659, pp. 116–126, 2010.

[94] A. Bobke, R. Vinuesa, R. Örlü, and P. Schlatter, "History effects and near equilibrium in adverse-pressure-gradient turbulent boundary layers," *Journal of Fluid Mechanics*, vol. 820, pp. 667–692, 2017.

[95] J. Bridges and M. Wernet, "Establishing consensus turbulence statistics for hot subsonic jets," in *16th AIAA/CEAS aeroacoustics conference*, 2010, p. 3751.

[96] J. Bridges and M. P. Wernet, "The nasa subsonic jet particle image velocimetry (piv) dataset," Tech. Rep., 2011.

[97] J.-P. Laval and M. Marquillie, "Direct numerical simulations of converging–diverging channel flow," in *Progress in wall turbulence: understanding and modeling.* Springer, 2011, pp. 203–209.

[98] Y. Bentaleb, S. Lardeau, and M. A. Leschziner, "Large-eddy simulation of turbulent boundary layer separation from a rounded step," *Journal of Turbulence*, no. 13, p. N4, 2012.

[99] M. Breuer, N. Peller, C. Rapp, and M. Manhart, "Flow over periodic hills–numerical and experimental study in a wide range of Reynolds numbers," *Computers & Fluids*, vol. 38, no. 2, pp. 433–457, 2009.

[100] D. Greenblatt, K. B. Paschal, C.-S. Yao, J. Harris, N. W. Schaeffler, and A. E. Washburn, "Experimental investigation of separation control part 1: baseline and steady suction," *AIAA journal*, vol. 44, no. 12, pp. 2820–2830, 2006.

[101] S. Cherroud, X. Merle, P. Cinnella, and X. Gloerfelt, "Sparse bayesian learning of explicit algebraic reynolds-stress models for turbulent separated flows," *International Journal of Heat and Fluid Flow*, vol. 98, p. 109047, 2022.

[102] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds-Averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics*, vol. 807, pp. 155–166, 2016.

[103] J.-L. Wu, H. Xiao, R. Sun, and Q. Wang, "RANS equations with explicit data-driven Reynolds stress closure can be ill-conditioned," *Journal of Fluid Mechanics*, vol. 869, pp. 553–586, 2019.

[104] M. Schmelzer, R. P. Dwight, and P. Cinnella, "Symbolic regression of algebraic stress-strain relation for RANS turbulence closure," in *6th ECCOMAS European Conference on Computational Mechanics: Solids, Structures and Coupled Problems, ECCM 2018 and 7th ECCOMAS European Conference on Computational Fluid Dynamics, ECFD 2018*. International Centre for Numerical Methods in Engineering, CIMNE, 2020, pp. 1789–1795.

[105] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning research*, vol. 1, pp. 211–244, 2001.

[106] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[107] D. MacKay, "Bayesian methods for backpropagation networks," in *Models of neural networks III*. Springer, 1996, pp. 211–254.

[108] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[109] S. Balakrishnan and D. Madigan, "Priors on the variance in sparse Bayesian learning: the demi-Bayesian lasso," *Frontiers of Statistical Decision Making and Bayesian Analysis: In Honor of James O. Berger*, pp. 346–359, 2010.

[110] J. O. Berger, "Bayesian analysis," in *Statistical Decision Theory and Bayesian Analysis*. Springer, 1985, pp. 118–307.

[111] P. Schlatter, R. Orlu, Q. Li, G. Brethouwer, A. V. Johansson, P. H. Alfredsson, and D. S. Henningson, "Progress in simulations of turbulent boundary layers," in *Seventh International Symposium on Turbulence and Shear Flow Phenomena*. Begel House Inc., 2011.

[112] S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," *Journal of Fluid Mechanics*, vol. 879, pp. 217–254, 2019.

[113] P. Seshadri and G. Parks, "Effective-Quadratures (EQ): Polynomials for computational engineering studies," *The Journal of Open Source Software*, vol. 2, pp. 166–166, 2017.

[114] S. Girimaji, "Machine learning, scale resolving simulations and the future of predictive computations of engineering flows: A perspective," in *2022 Symposium on Turbulence Modeling: Roadblocks, and the Potential for Machine Learning*, 2022.

[115] C. Rumsey and G. Coleman, "Nasa symposium on turbulence modeling: Roadblocks, and the potential for machine learning," Tech. Rep., 2022.

[116] D. Coles, "The law of the wake in the turbulent boundary layer," *Journal of Fluid Mechanics*, vol. 1, no. 2, pp. 191–226, 1956.

[117] Coles, "The turbulent boundary layer in a compressible fluid," *The Physics of Fluids*, vol. 7, no. 9, pp. 1403–1423, 1964.

[118] F. M. White, *Viscous fluid flow*. McGraw-Hill New York, 1974, vol. 3.

[119] C. E. Rasmussen, C. K. Williams *et al.*, *Gaussian processes for machine learning*. Springer, 2006, vol. 1.

[120] D. Xiu and G. E. Karniadakis, "The wiener–askey polynomial chaos for stochastic differential equations," *SIAM journal on scientific computing*, vol. 24, no. 2, pp. 619–644, 2002.

[121] R. C. Smith, *Uncertainty quantification: theory, implementation, and applications*. Siam, 2013, vol. 12.

[122] B. Sudret, "Global sensitivity analysis using polynomial chaos expansions," *Reliability engineering & system safety*, vol. 93, no. 7, pp. 964–979, 2008.

Appendix

Appendix A

Uncertainty quantification method

In this section, we describe the uncertainty quantification (UQ) method used in the *equadratures* library [113]. Let $f(\boldsymbol{\theta})$ be an expensive scalar-valued quantity of interest, where $\boldsymbol{\theta} = (\theta_1, ..., \theta_d)^T$ is a $d-$ dimensional vector of mutually independent random variables. Here, each parameter θ_i belongs to a domain $\Theta_i \in \mathbb{R}$. We usually consider the input domain to be a non-compact hypercube decomposed as a cartesian product of the form $\boldsymbol{\Theta} = \Theta^1 \times ... \times \Theta^d$.

Let $\rho_i(\theta_i)$ be a probability density function over the domain Θ^i. In making the assumption that $\boldsymbol{\theta}$ is a vector of independent random variables, the joint density ρ of all the probability distributions associated with $\boldsymbol{\theta}$ is given by:

$$\rho(\boldsymbol{\theta}) = \prod_{i=1}^{d} \rho_i(\theta_i) \tag{A.1}$$

.

As we intend to approximate f via a finite number of polynomials ϕ_i , we restrict indices i to lie in a finite multi-index set $\mathcal{I}$. Whilst considerable flexibility in specifying these multi-index sets exists, well-known $\mathcal{I}$ include tensor product index sets, total order index sets and hyperbolic spaces, as illustrated in Figure A.1. Each of these index sets in d dimensions is well-defined given a fixed $k \in \mathbb{N}$, which indicates the maximum polynomial degree associated to these sets:

- Tensor product index set $\mathcal{I}$ are characterized by:

$$\max_k i_k \leq k, \tag{A.2}$$

and have a cardinality equal to $(k+1)^d$.

- Total order index set $\mathcal{I}$ contain multi-indices satisfying:

$$\sum_{j=1}^{d} i_j \leq k, \tag{A.3}$$

with a total order index set $\mathcal{I}$ has cardinality

$$|\mathcal{I}| = \binom{k+d}{k}$$

.

- For a hyperbolic space index set:

$$\left(\sum_{j=1}^{d} i_j^q\right)^{\frac{1}{q}} \leq k, \tag{A.4}$$

where q is a user-defined constant that can be varied from 0.2 to 1.0. When this parameter is set to unity $q = 1$, the hyperbolic index space is equivalent to a total order index space. For values less than unity higher-order interactions terms are eliminated.

Of particular importance is the growth and interaction of the higher order terms. In total order and hyperbolic spaces, higher order interaction terms are significantly reduced. For many physical systems, these type of sparser basis have found utility as lower order interactions are often far more dominant.

Now, Assume that f is sufficiently smooth and continuous such that it can be approximated by a global polynomial p:

$$\begin{aligned} f(\boldsymbol{\theta}) &\approx p(\boldsymbol{\theta}) \\ &= \sum_{j=1}^{N} \xi_j \phi_{\mathbf{j}}(\boldsymbol{\theta}), \\ &= \mathbf{P}\xi \end{aligned} \tag{A.5}$$

defined as a weighted sum of N known basis polynomials, where

$$\phi_j(\boldsymbol{\theta}) = \prod_{k=1}^{d} \phi_{jk}^{(k)}\left(\theta_k\right), \qquad (j_1, ..., j_d) \in \mathbb{N}^d, \tag{A.6}$$

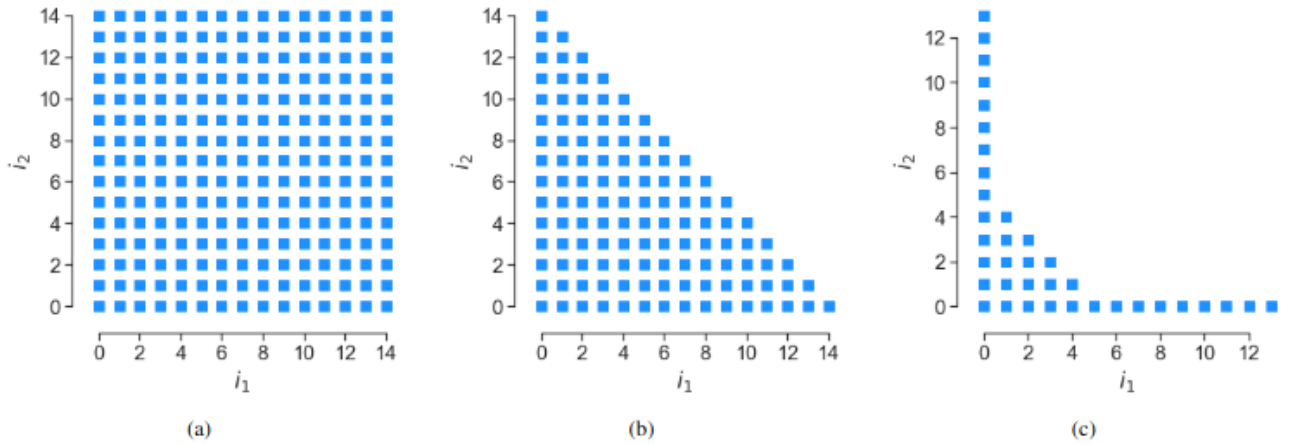


Figure A.1: Multi-indices for $d = 2$ with a maximum univariate degree of 14 for: (a) tensor order; (b): total order; (c): hyperbolic space.

and where $[\mathbf{P}]_{ij} = \phi_{\mathbf{j}}(\theta_i)$ for some discretized value $\theta_i \in \Theta_i$. Note that the polynomials ϕ_j defined in this way are mutually orthogonal in L^2 weighted by ρ. More specifically, we can state that these composite univariate polynomials must satisfy:

$$\int_{\times^{(k)}} \phi_g(\theta_k)\phi_h(\theta_i)\rho_k(\theta_k)d\theta_k = \delta_{gh}, \tag{A.7}$$

where δ_{gh} is the Kronecker delta; subscripts g and h denote polynomial orders. he above expression crystallizes the choice of the orthogonal polynomial family based on the choice of the weight function $\rho(\theta_k)$. For instance if $\rho(\theta_k)$ were the uniform distribution with $\Theta^{(k)} \in [a, b]$ then $\{\phi_j(\theta_k)\}$ would correspond to Legendre polynomials; for Gaussian weights one would use Hermite polynomials, and so on. Details about these weight-polynomial pairs can be found in [120].

These coefficients are defined to be:

$$\xi_j = \int_{\times} f(\boldsymbol{\theta})\phi_{\mathbf{j}}(\boldsymbol{\theta})\rho(\boldsymbol{\theta})d\boldsymbol{\theta} \tag{A.8}$$

which may be interpreted as the inner product of the function over the $j - th$ polynomial term. The overarching goal in this section will be the utilization of Gaussian quadrature for approximating Equation A.8 via:

$$\xi_j \approx \sum_{i=1}^{M} f(\chi_i)\phi_j(\chi_i)\omega_i \tag{A.9}$$

using quadrature points $\{\chi_i\}_{i=1}^{M} \in \mathcal{X}$, and a set of corresponding weights $\{\omega_i\}_{i=1}^{M}$, defined via:

$$\omega_i = \frac{\tilde{\omega}_i}{\sum_{i=1}^{M} \tilde{\omega}_i}, \quad \text{where} \quad \tilde{\omega}_i = \frac{1}{\sum_{j=1}^{N} \phi_j(\chi_i)} \tag{A.10}$$

To solve Equation A.9 for all j's, we assume access to input-output model evaluations of the form $\{\chi_i, f_j\}_{i=1}^{M}$.

In *equadratures*, coefficient computation strategies are passed as string input methods to the Polynomial class and include "least-squares", "compressed-sensing" and "relevance-vector-machine". While least-squares solutions place a restriction on the polynomial evaluation matrix ϕ as there must be at least as many rows as columns, we rather foxus on heuristics that permit bypassing this restriction. Assuming that the solution is sparse, *i.e.* with many zeros or near-zeros, it can be shown that heuristics based on L^1-minimization (LASSO) can be used. In practice, the use of L^1-minimization methods can give slow performance. The main bottleneck to the method is the determination of the unknown hyperparameter of the LASSO problem. Depending on the scale of the problem, it can take a wide range of values. Even when the output is normalized, several plausible values of the L^1 hyperparameter still need to be tested with trial-and-error. This in turn implies that multiple optimization problems need to be solved for one regression task. Below we describe an alternative approach for underdetermined sparse regression that obviates the need for hyperparameter searching with cross validation, namely relevance vector machines (RVMs).

The RVM is first proposed by Tipping [105] and introduced as a Bayesian method for compressed sensing in *equadratures* by implementing the Sparse Bayesian Learning algorithm. This method considers the task of coefficient computation in a probabilistic framework. The regression model is formulated using a generative process with Gaussian noise:

$$f(\boldsymbol{\theta}) \approx \sum_{j=1}^{N} \xi_j \phi_j(\boldsymbol{\theta}) + \epsilon = y \tag{A.11}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The hierarchical priors are constructed and inference is formulated in the same manner as in Section 4.2.

A.0.1 Computation of statistics

The *Statistics* class in *equadratures* computes moments, Sobol' indices and higher-order statistical metrics from the calculated coefficients. This builds upon some core ideas arising from polynomial chaos in the context of uncertainty quantification.

Moments

Computation of the mean and variance are readily straightforward when using pseudospectral approximations [121]. We can write the mean as:

$$\begin{aligned}\mathbb{E}(f(\boldsymbol{\theta})) &\approx \mathbb{E}(p(\boldsymbol{\theta})) \\ &= \mathbb{E}\left[\sum_{i=1}^{n} \xi_i \phi_{\mathbf{i}}(\boldsymbol{\theta})\right] \\ &= \mathbb{E}\left[\xi_1 \phi_{\mathbf{1}}(\boldsymbol{\theta})\right] + \underbrace{\mathbb{E}\left[\sum_{i=2}^{n} \xi_i \phi_{\mathbf{i}}(\boldsymbol{\theta})\right]}_{=0} \\ &= \xi_1\end{aligned} \tag{A.12}$$

In other words, the mean is simply given by the first coefficient of the expansion. Estimating the variance is also straightforward:

$$\begin{aligned}Var(f(\mathbf{x})) &\approx Var(p(\mathbf{x})) \\ &= \mathbb{E}\left[\left(\sum_{i=1}^{n} \xi_i \phi_{\mathbf{i}}(\boldsymbol{\theta}) - \xi_1\right)^2\right] \\ &= \mathbb{E}\left[\left(\sum_{i=2}^{n} \xi_i \phi_{\mathbf{i}}(\boldsymbol{\theta})\right)^2\right] \\ &= \sum_{i=2}^{n} \xi_i^2\end{aligned} \tag{A.13}$$

It is this ability to rapidly estimate statistical moments in the absence of additional sampling that makes these polynomial approximations so useful.

Global sensitivity analysis

Engineers are often interested in the answer to the question, "which of my model parameters are the most important?" This is the one of the objectives of global sensitivity analysis. It seeks

to apportion the uncertainty f according to the contribution of the inputs $\boldsymbol{\theta}$. Typically, in this context importance is characterized by the conditional variance. One relatively well-known strategy to quantify the importance of various inputs is through Sobol' indices (see page 323 in [121]), which can be readily approximated using orthogonal polynomial expansions [122]. As before let $\mathcal{I}$ be the multi-index set associated with a chosen polynomial basis used for approximating a function f. From Equation A.13, we can write the variance as:

$$Var(f(\mathbf{x})) = \sigma^2 = \sum_{i \in \mathcal{I}, i \neq 1} \xi_i^2 \tag{A.14}$$

Now Sobol' indices represent a fraction of the total variance that is attributed to each input variable (the first order Sobol' indices) or combinations thereof (higher order Sobol' indices). Let $\mathcal{I}_s$ be the set of multi-indices that depend only on the subset of variables $\mathbf{s} = \{i_1, ..., i_s\}$, *i.e.*,

$$\mathcal{I}_s = \left\{ i \in \mathbb{N}^d : l \in \mathbf{s} \Leftrightarrow i_l \neq 0 \right\}. \tag{A.15}$$

The first order partial variances σ_j^2 are then obtained by summing up the square of the coefficients in $\mathcal{I}_s$:

$$\sigma_j^2 = \sum_{i \in \mathcal{I}_j} \xi_i^2, \quad \mathcal{I}_j = \left\{ \mathbf{i} \in \mathbb{N}^d : i_j > 0 \right\} \tag{A.16}$$

and the higher order variances $\sigma^2_{i_1,...,i_s}$ can be written as:

$$\sigma_{\mathbf{s}}^2 = \sum_{\mathbf{i} \in \mathcal{I}_{\mathbf{s}}}, \quad \mathcal{I}_{\mathbf{s}} = \left\{ \mathbf{i} \in \mathbb{N}^d : l \in \mathbf{s} \Leftrightarrow i_l > 0 \right\} \tag{A.17}$$

The first and higher order Sobol' indices are then given by:

$$S_j = \frac{\sigma_j^2}{\sigma^2} \quad \text{and} \quad S_{\mathbf{s}} = \frac{\sigma_{\mathbf{s}}^2}{\sigma^2} \tag{A.18}$$

Appendix B

SBL-SpaRTA models for turbulent separated flows

This section reports the different models discovered for separated flows. According to the choice of the training flows and of the regularization hyperparameter lambda, various models are obtained, and submitted to cross-validation (see Chapter 4).

In the following, we provide the mathematical expressions of the various models, for various choices of λ.

- $\lambda = 10^2$:

$$\begin{cases} \mathbf{M}^{(1)}_{\mathbf{b}^\Delta} = & [(-0.496 \pm 0.0133) + (21.6 \pm 0.366) I_1 + (17.4 \pm 0.374) I_2] \mathbf{T}^{(1)} \\ & + [(7.52 \pm 0.0378) + (89 \pm 1.39) I_2] \mathbf{T}^{(2)} \\ & + [(2.78 \pm 0.0825)] \mathbf{T}^{(3)} \pm 0.00354 \\ \mathbf{M}^{(1)}_{\mathbf{b}^R} = & [(0.989 \pm 0.0153)] \mathbf{T}^{(1)} \pm 0.0157 \end{cases} \tag{B.1}$$

$$\begin{cases} \mathbf{M}^{(2)}_{\mathbf{b}^\Delta} = & [(-0.540 \pm 0.0138) + (22.8 \pm 0.376) I_1 + (17.2 \pm 0.395) I_2] \mathbf{T}^{(1)} \\ & + [(7.18 \pm 0.0370) + (-69.6 \pm 1.38) I_2] \mathbf{T}^{(2)} \\ & + [(2.82 \pm 0.0848)] \mathbf{T}^{(3)} \pm 0.00357 \\ \mathbf{M}^{(2)}_{\mathbf{b}^R} = & [(0.863 \pm 0.0274)] \mathbf{T}^{(1)} \pm 0.0381 \end{cases} \tag{B.2}$$

$$\begin{cases} \mathbf{M}^{(3)}_{\mathbf{b}^\Delta} = & [(-0.209 \pm 0.00837) + (0.938 \pm 0.221) I_1] \mathbf{T}^{(1)} \\ & + [(8.25 \pm 0.0473) + (-72.2 \pm 1.45) I_1 + (29.2 \pm 1.07) I_2] \mathbf{T}^{(2)} \\ & + [(5.08 \pm 0.0881)] \mathbf{T}^{(3)} \pm 0.00113 \\ \mathbf{M}^{(3)}_{\mathbf{b}^R} = & [(0.872 \pm 0.0322)] \mathbf{T}^{(1)} \pm 0.0358 \end{cases} \tag{B.3}$$

- $\lambda = 10^3$:

$$\begin{cases} \mathbf{M}^{(1)}_{\mathbf{b}^\Delta} = & [(-0.406 \pm 0.0123) + (16.9 \pm 0.339)I_1 + (15.2 \pm 0.341)I_2]\mathbf{T}^{(1)} \\ & +[(5.36 \pm 0.0190)]\mathbf{T}^{(2)} \\ & +[(2.52 \pm 0.0841)]\mathbf{T}^{(3)} \pm 0.00379 \\ \mathbf{M}^{(1)}_{\mathbf{b}^R} = & [(0.982 \pm 0.0152)]\mathbf{T}^{(1)} \pm 0.0157 \end{cases} \tag{B.4}$$

$$\begin{cases} \mathbf{M}^{(2)}_{\mathbf{b}^\Delta} = & [(-0.465 \pm 0.0125) + (18.3 \pm 0.345)I_1 + (14.5 \pm 0.348)I_2]\mathbf{T}^{(1)} \\ & +[(5.54 \pm 0.0194)]\mathbf{T}^{(2)} \\ & +[(2.57 \pm 0.0848)]\mathbf{T}^{(3)} \pm 0.00375 \\ \mathbf{M}^{(2)}_{\mathbf{b}^R} = & [(0.840 \pm 0.0270)]\mathbf{T}^{(1)} \pm 0.0381 \end{cases} \tag{B.5}$$

$$\begin{cases} \mathbf{M}^{(3)}_{\mathbf{b}^\Delta} = & [(-0.172 \pm 0.00544)]\mathbf{T}^{(1)} \\ & +[(5.39 \pm 0.0253) + (18.1 \pm 0.571)I_2]\mathbf{T}^{(2)} \\ & +[(4.91 \pm 0.0911)]\mathbf{T}^{(3)} \pm 0.00121 \\ \mathbf{M}^{(3)}_{\mathbf{b}^R} = & [(0.839 \pm 0.0316)]\mathbf{T}^{(1)} \pm 0.0358 \end{cases} \tag{B.6}$$

- $\lambda = 10^4$:

$$\begin{cases} \mathbf{M}^{(1)}_{\mathbf{b}^\Delta} = & [(-0.195 \pm 0.00498)]\mathbf{T}^{(1)} \\ & +[(5.29 \pm 0.0203)]\mathbf{T}^{(2)} \\ & +[(1.59 \pm 0.0715)]\mathbf{T}^{(3)} \pm 0.00408 \\ \mathbf{M}^{(1)}_{\mathbf{b}^R} = & [(0.959 \pm 0.0151)]\mathbf{T}^{(1)} \pm 0.0157 \end{cases} \tag{B.7}$$

$$\begin{cases} \mathbf{M}^{(2)}_{\mathbf{b}^\Delta} = & [(-0.217 \pm 0.00527) + (5.88 \pm 0.0637)I_1]\mathbf{T}^{(1)} \\ & +[(5.47 \pm 0.0208)]\mathbf{T}^{(2)} \\ & +[(1.62 \pm 0.0722)]\mathbf{T}^{(3)} \pm 0.00404 \\ \mathbf{M}^{(2)}_{\mathbf{b}^R} = & [(0.766 \pm 0.0258)]\mathbf{T}^{(1)} \pm 0.0381 \end{cases} \tag{B.8}$$

$$\begin{cases} \mathbf{M}^{(3)}_{\mathbf{b}^\Delta} = & [(-0.166 \pm 0.00552)]\mathbf{T}^{(1)} \\ & +[(4.75 \pm 0.0170)]\mathbf{T}^{(2)} \\ & +[(4.00 \pm 0.0841)]\mathbf{T}^{(3)} \pm 0.00124 \\ \mathbf{M}^{(3)}_{\mathbf{b}^R} = & [(0.737 \pm 0.0296)]\mathbf{T}^{(1)} \pm 0.358 \end{cases} \tag{B.9}$$

- $\lambda = 10^5$:

$$\begin{cases} \mathbf{M}^{(1)}_{\mathbf{b}^\Delta} = & [(5.09 \pm 0.0206)]\mathbf{T}^{(2)} + \pm 0.0042 \\ \mathbf{M}^{(1)}_{\mathbf{b}^R} = & [(0.887 \pm 0.0145)]\mathbf{T}^{(1)} \pm 0.0157 \end{cases} \tag{B.10}$$

$$\begin{cases} \mathbf{M}^{(2)}_{\mathbf{b}^\Delta} = & [(5.26 \pm 0.0211)]\mathbf{T}^{(2)} + \pm 0.00417 \\ \mathbf{M}^{(2)}_{\mathbf{b}^R} = & [(0.53 \pm 0.0216)]\mathbf{T}^{(1)} \pm 0.0383 \end{cases} \tag{B.11}$$

$$\begin{cases} \mathbf{M}^{(3)}_{\mathbf{b}^\Delta} = & [(4.62 \pm 0.0173)]\mathbf{T}^{(2)} + [(0.845 \pm 0.0399)]\mathbf{T}^{(3)} \pm 0.00128 \\ \mathbf{M}^{(3)}_{\mathbf{b}^R} = & [(0.407 \pm 0.0222)]\mathbf{T}^{(1)} \pm 0.0361 \end{cases} \tag{B.12}$$

- $\lambda = 2 \times 10^5$:

$$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(1)} = & \pm 0.00669 \\ \mathbf{M}_{\mathbf{b}^R}^{(1)} = & [(0.8433 \pm 0.0142)]\mathbf{T}^{(1)} \pm 0.0158 \end{cases} \tag{B.13}$$

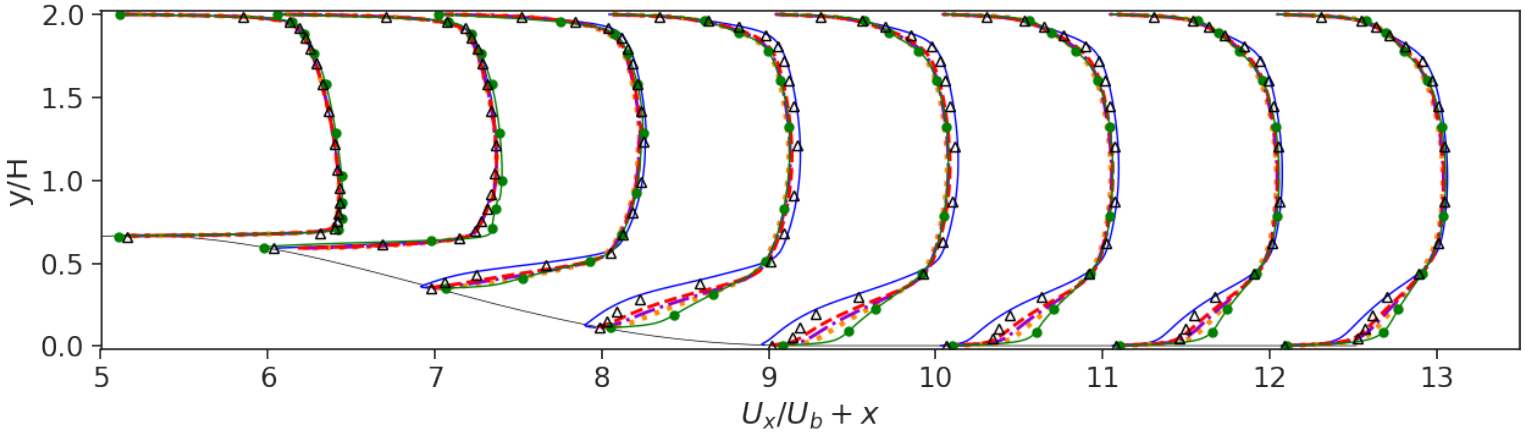
$$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(2)} = & \pm 0.00669 \\ \mathbf{M}_{\mathbf{b}^R}^{(2)} = & [(0.382 \pm 0.0184)]\mathbf{T}^{(1)} \pm 0.0385 \end{cases} \tag{B.14}$$

$$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(3)} = & \pm 0.00214 \\ \mathbf{M}_{\mathbf{b}^R}^{(3)} = & [(0.197 \pm 0.0156)]\mathbf{T}^{(1)} \pm 0.0364 \end{cases} \tag{B.15}$$

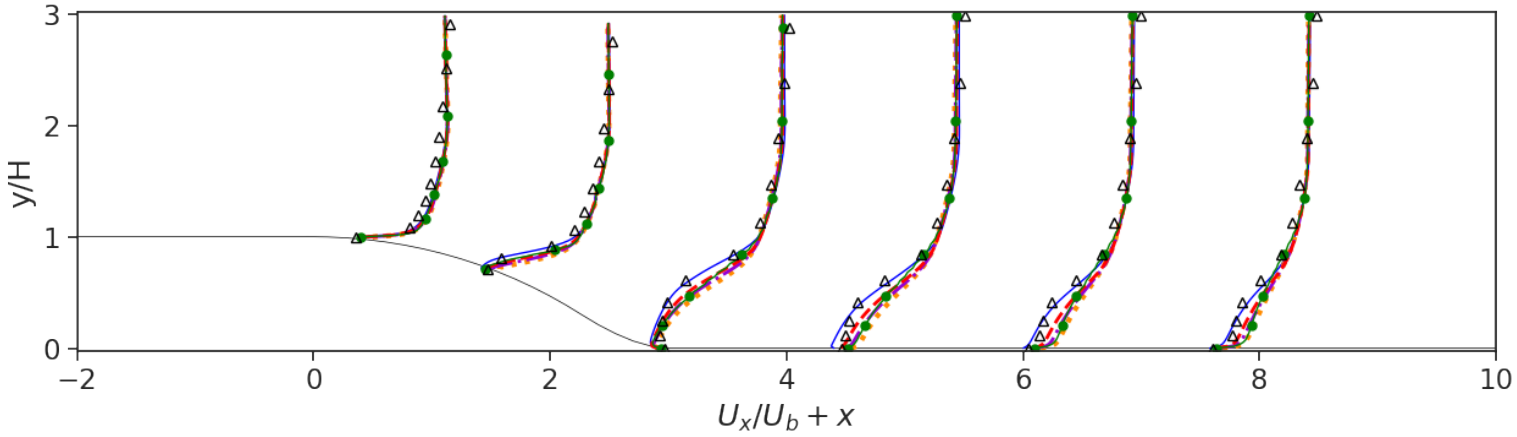
B.1 SBL models vs physics-based EARSM

In this Appendix we assess the best machine-learned model $\mathbf{M}^{(3)}$ against a physics-based EARSM model. More precisely, we consider the BSL-EARSM of Menter *et al.* [23], available through the *OpenFoam* software. This model is based on the EARSM formulation of Wallin and Johansson (WJ) [21] for the stress-strain relationship (derived from Pope's generalized eddy viscosity formulation), supplemented by the transport equations for the k-ω SST model to reduce its sensitivity to the free-stream conditions.

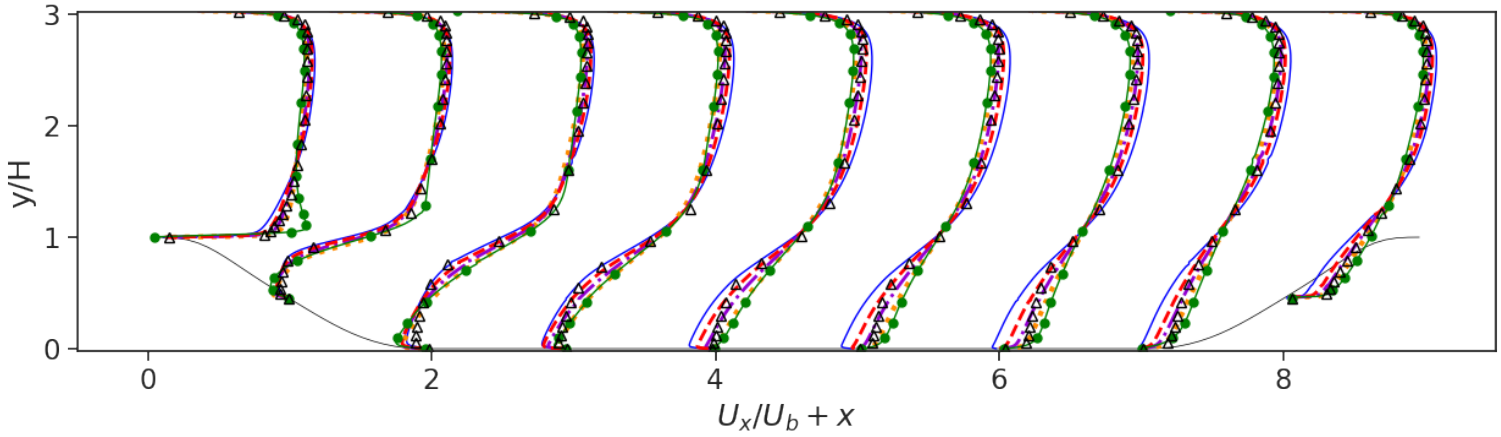
In Figures B.1, B.2 and B.3 we report selected numerical results for various QoI and flow cases. BSL-EARSM improves some QoI such as the velocity or the skin friction over the LEVM, but it is less accurate than the present models, and specifically $\mathbf{M}^{(3)}$. Additionally, it also fails in capturing the turbulent kinetic energy profiles accurately.



(a) Converging-diverging channel.

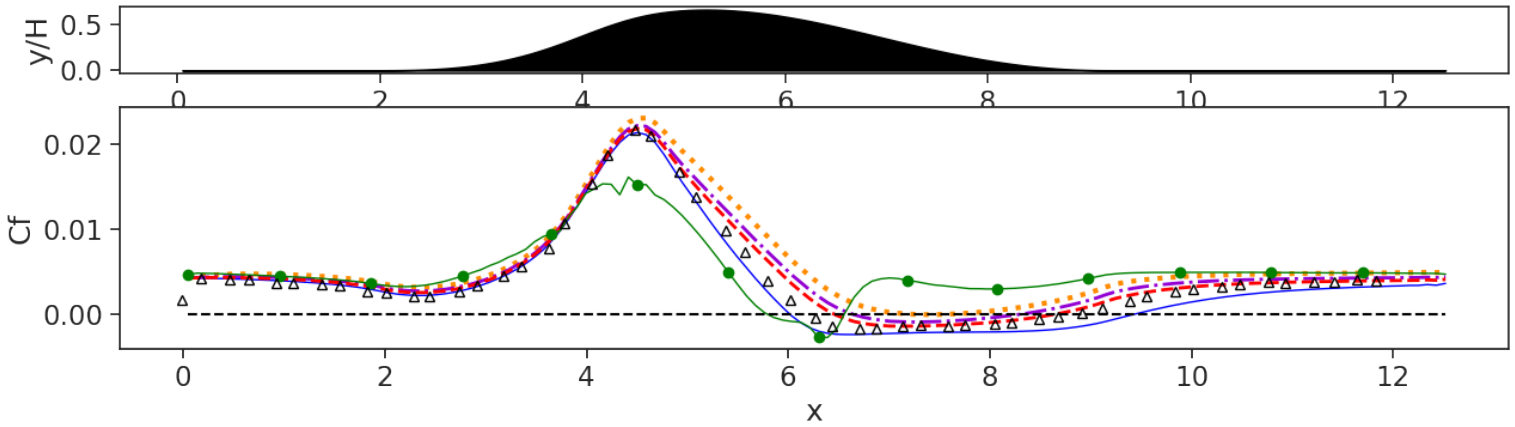


(b) Curved Backward-Facing Step.

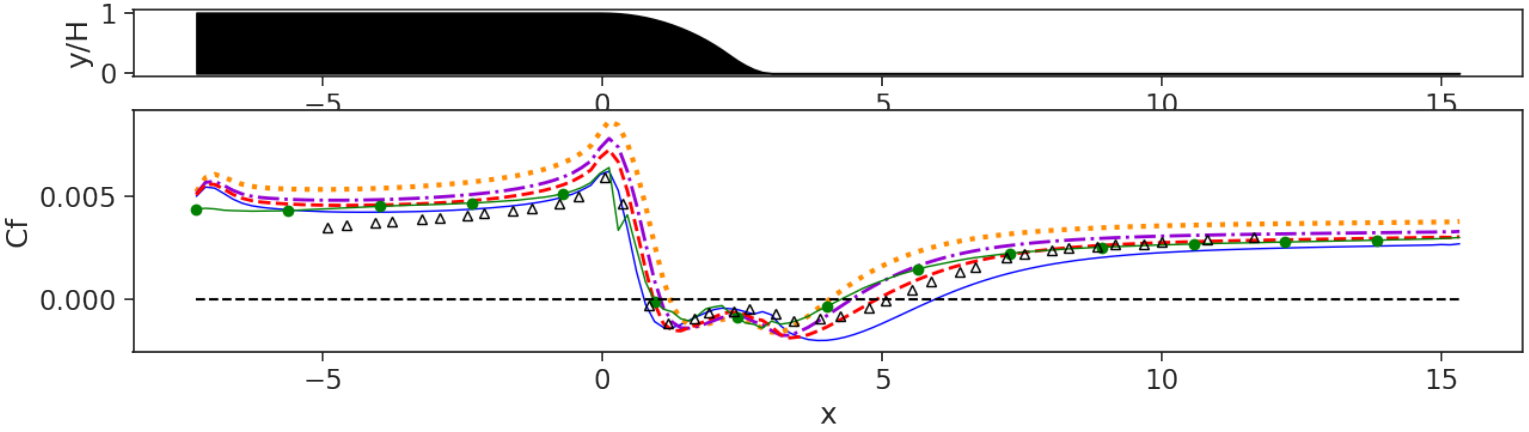


(c) Periodic Hills.

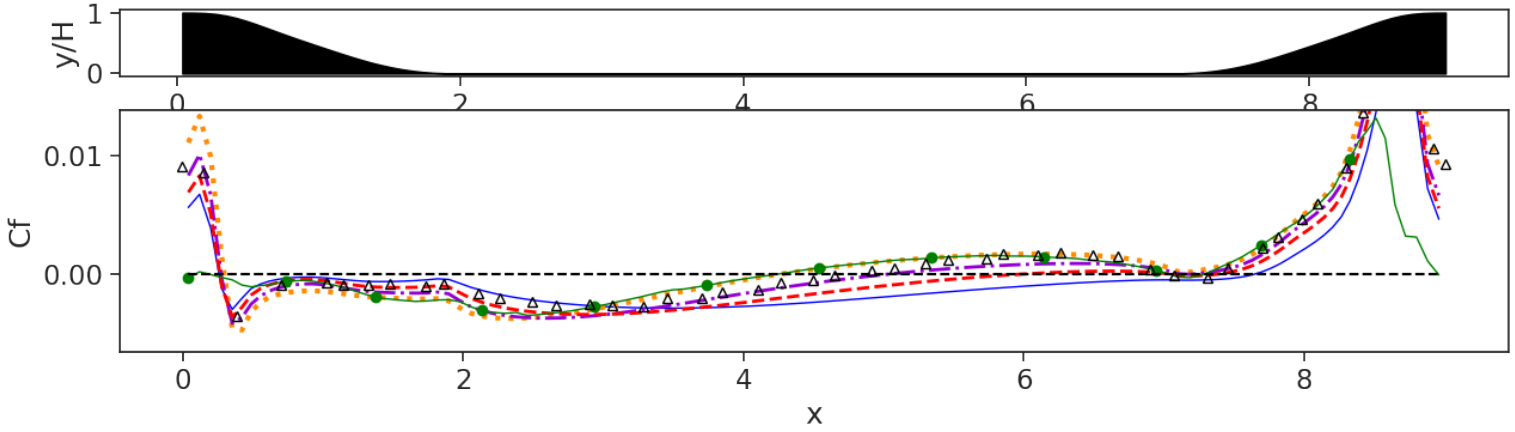
Figure B.1: Streamwise velocity profiles. Baseline $k-\omega$ SST (—), LES (—●—), $\mathbf{M}^{(1)}$ (·····), $\mathbf{M}^{(2)}$ (-·-·-) and $\mathbf{M}^{(3)}$ (- - -) compared to BSL-EARSM (△).



(a) Converging-diverging channel.

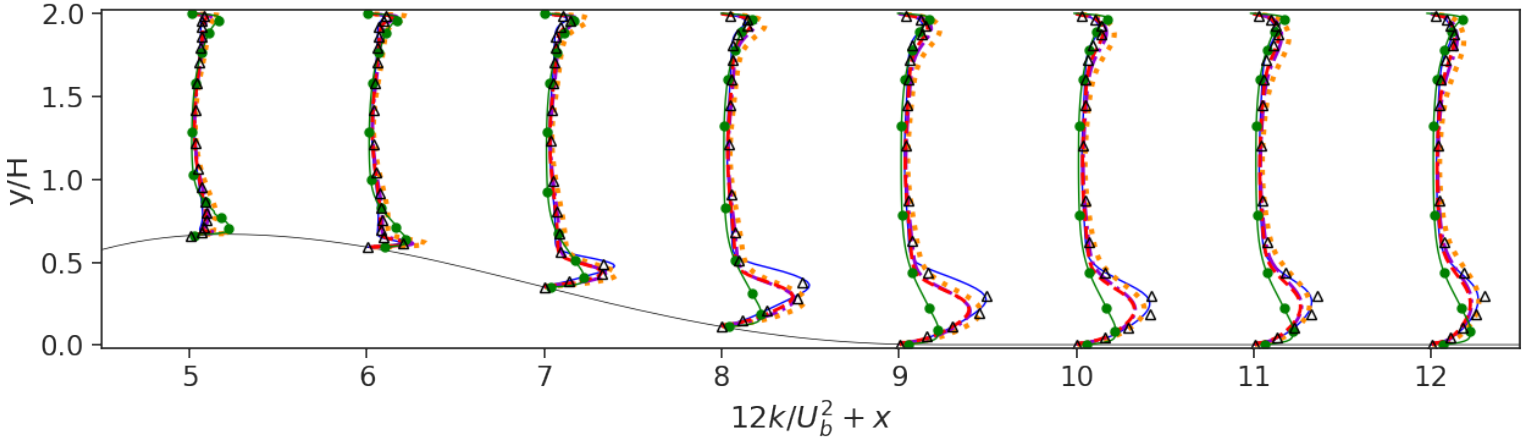


(b) Curved Backward-Facing Step.

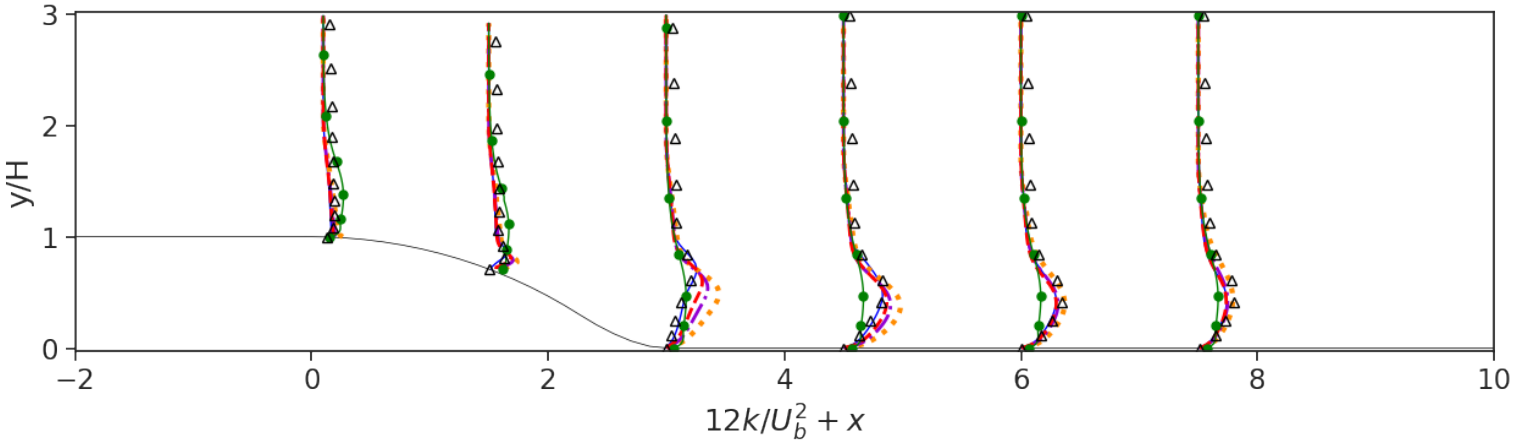


(c) Periodic Hills.

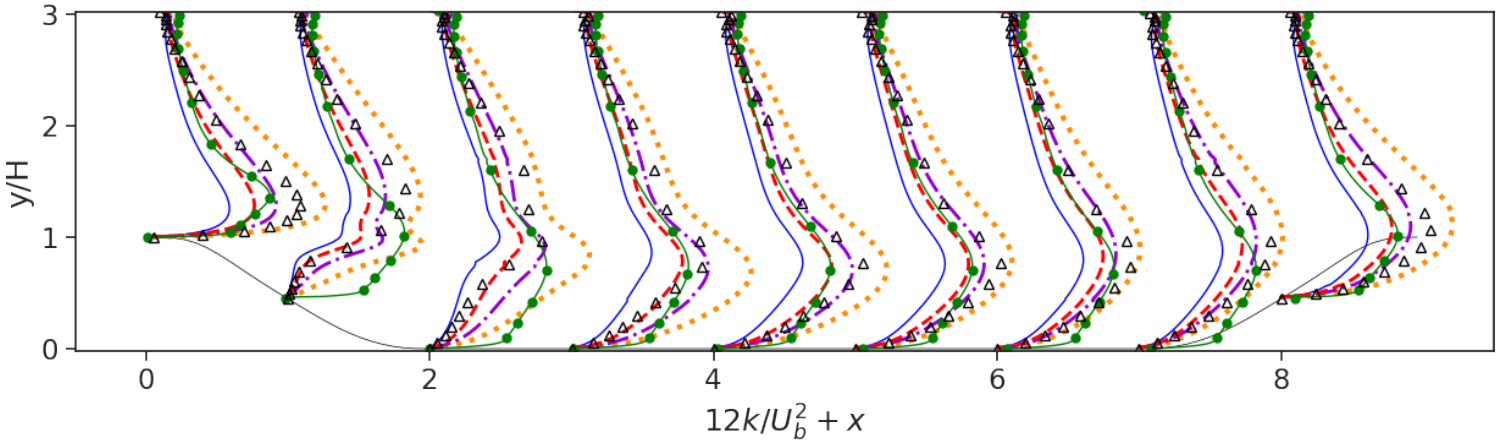
Figure B.2: Friction coefficient predictions. Baseline $k-\omega$ SST (—), LES (—●—), $\mathbf{M}^{(1)}$ (·····), $\mathbf{M}^{(2)}$ (-·-·-) and $\mathbf{M}^{(3)}$ (- - -) compared to BSL-EARSM (Δ).



(a) Converging-diverging channel.



(b) Curved Backward-Facing Step.



(c) Periodic Hills.

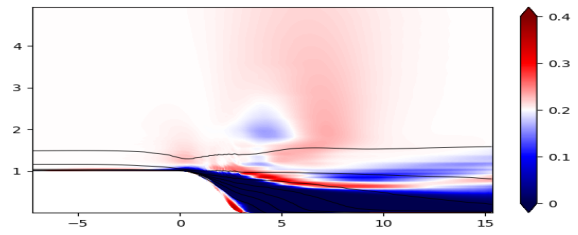
Figure B.3: Turbulent kinetic energy profiles. Baseline $k-\omega$ SST (—), LES (—●—), $\mathbf{M}^{(1)}$ (·····) , $\mathbf{M}^{(2)}$ (-·-·-) and $\mathbf{M}^{(3)}$ (- - -) compared to BSL-EARSM (△).

Appendix C

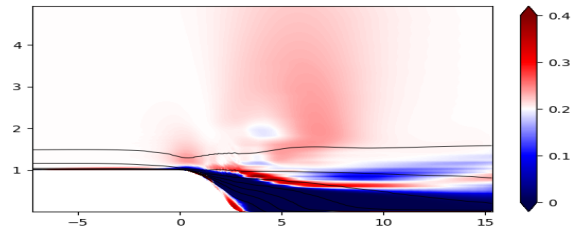
Model weights for non-intrusive X-MA (Chapter 5)

In this Appendix, we present additional data to support the findings in Chapter 5. First, we show in Section C.1 the optimal model weights for a set of training flow cases, including separated flow cases and turbulent boundary layers under various adverse pressure gradients (Figures C.1, C.2, C.3, C.4, C.5 and C.6). Then, in Section C.2, we present the optimal model weights for two test flow cases (Figures C.7 and C.8).

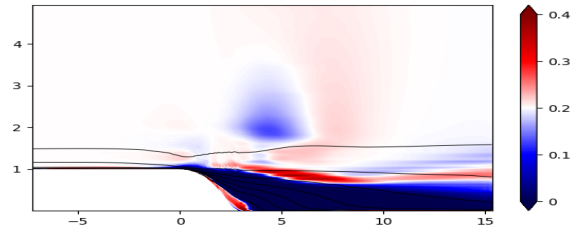
C.1 Model weights used for training



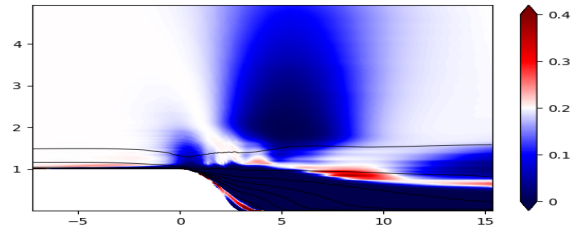
(a) w_{ZPG}



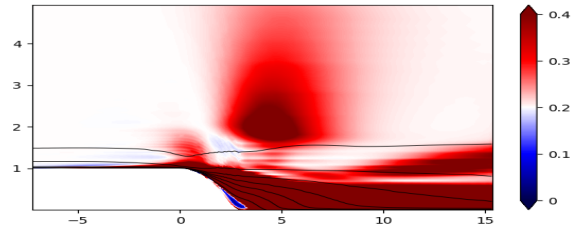
(b) w_{CHAN}



(c) w_{APG}

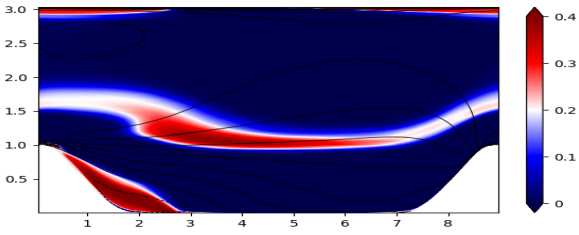


(d) w_{ANSJ}

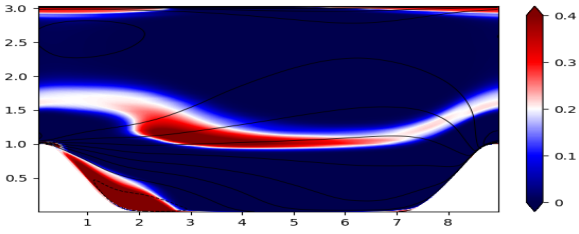


(e) w_{SEP}

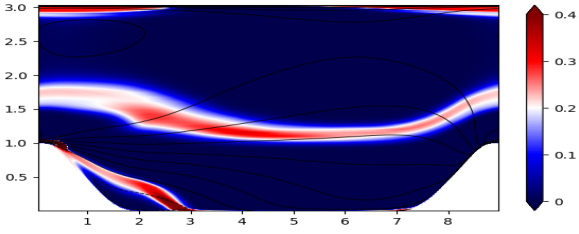
Figure C.1: Colormaps of exact optimal model weights for the CBFS flow (various SBL-SpaRTA) and iso-contours of the longitudinal velocity (baseline model).



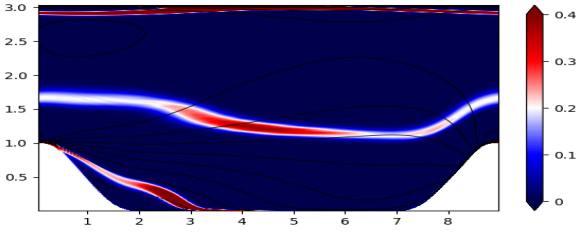
(a) w_{ZPG}



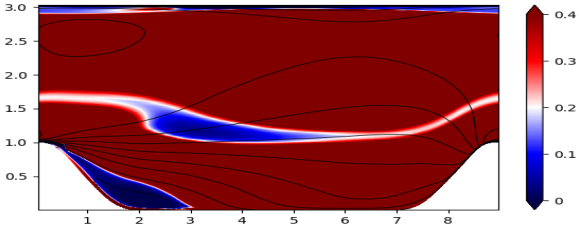
(b) w_{CHAN}



(c) w_{APG}

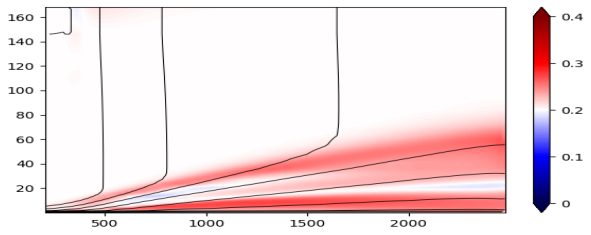


(d) w_{ANSJ}

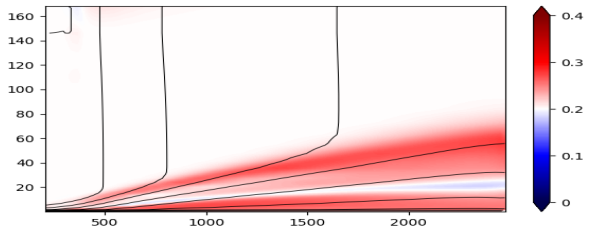


(e) w_{SEP}

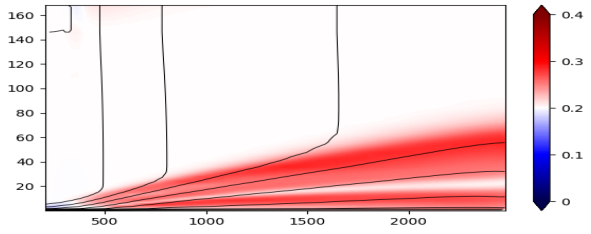
Figure C.2: Colormaps of exact optimal model weights for the PH flow (various SBL-SpaRTA) and iso-contours of the longitudinal velocity (baseline model).



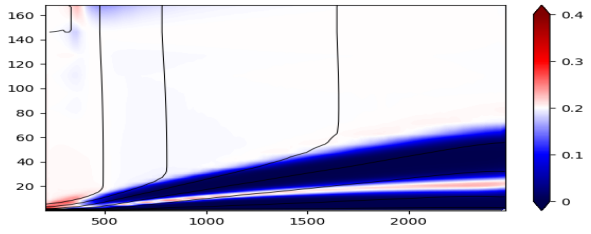
(a) w_{ZPG}



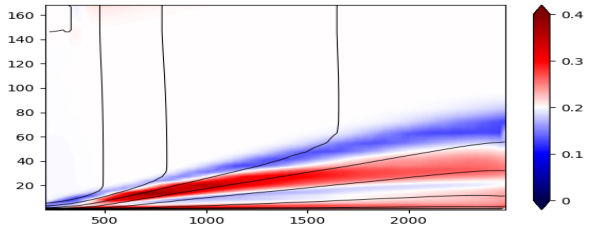
(b) w_{CHAN}



(c) w_{APG}

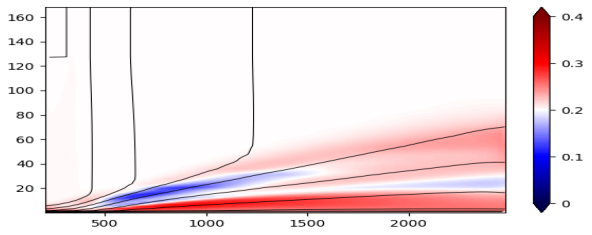


(d) w_{ANSJ}

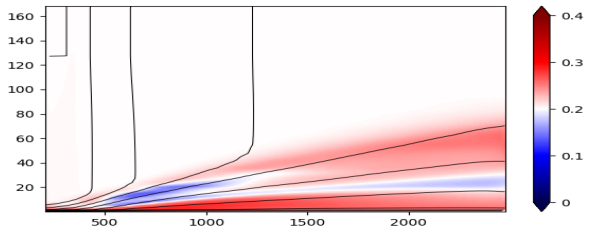


(e) w_{SEP}

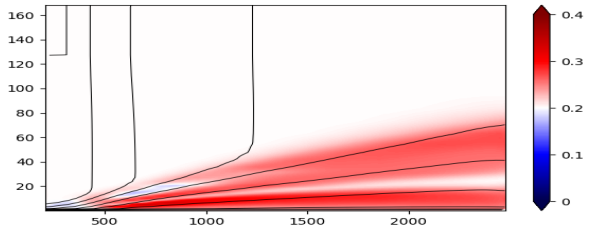
Figure C.3: Colormaps of exact optimal model weights for the APG-TBL-b1n flow (various SBL-SpaRTA) and iso-contours of the longitudinal velocity (baseline model).



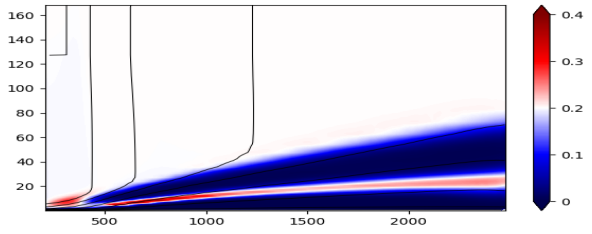
(a) w_{ZPG}



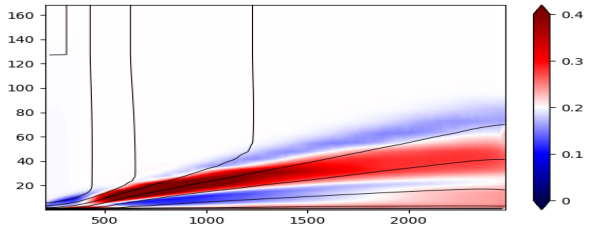
(b) w_{CHAN}



(c) w_{APG}

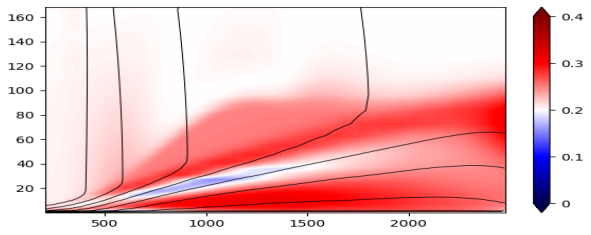


(d) w_{ANSJ}

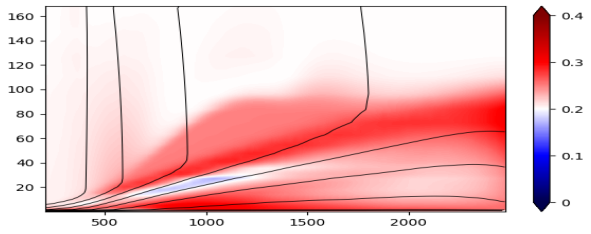


(e) w_{SEP}

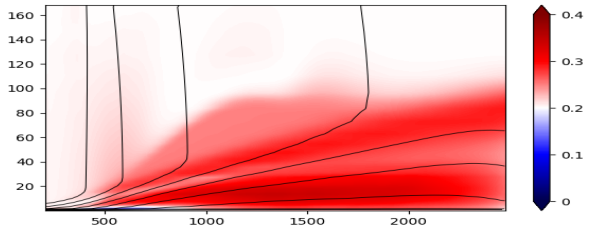
Figure C.4: Colormaps of exact optimal model weights for the APG-TBL-b2n flow (various SBL-SpaRTA) and iso-contours of the longitudinal velocity (baseline model).



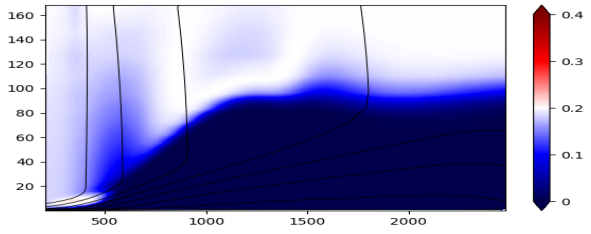
(a) w_{ZPG}



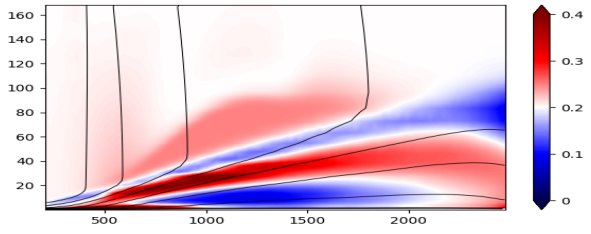
(b) w_{CHAN}



(c) w_{APG}

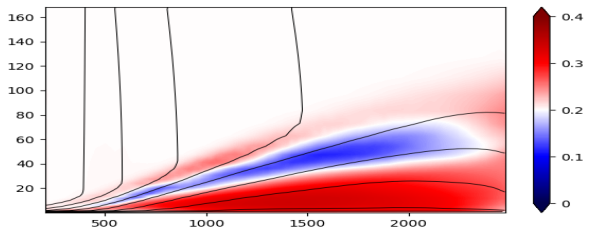


(d) w_{ANSJ}

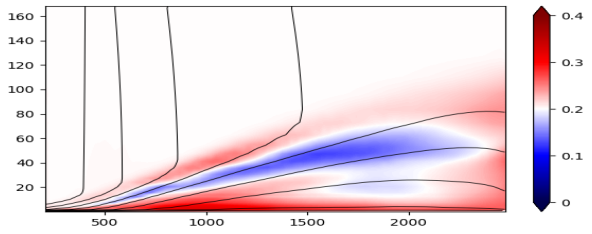


(e) w_{SEP}

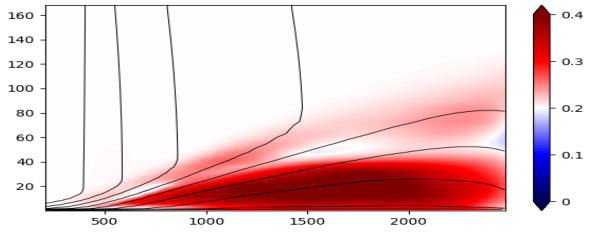
Figure C.5: Colormaps of exact optimal model weights for the APG-TBL-m18n flow (various SBL-SpaRTA) and iso-contours of the longitudinal velocity (baseline model).



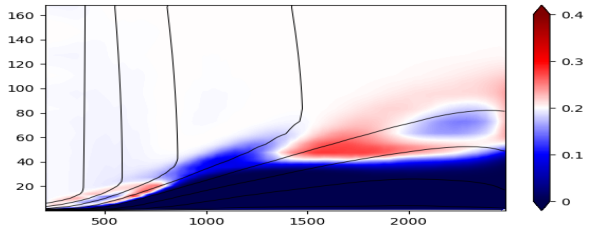
(a) w_{ZPG}



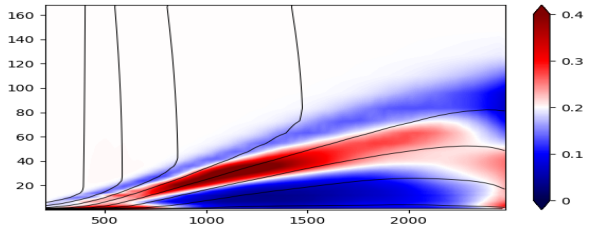
(b) w_{CHAN}



(c) w_{APG}



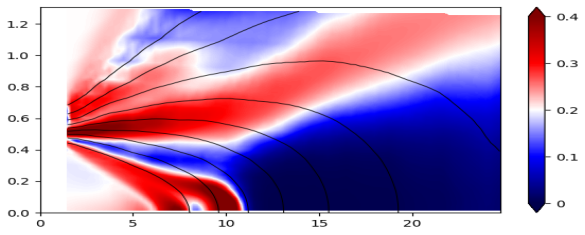
(d) w_{ANSJ}



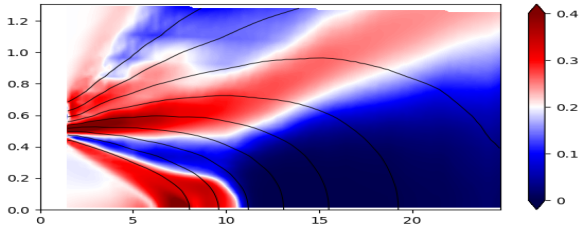
(e) w_{SEP}

Figure C.6: Colormaps of exact optimal model weights for the APG-TBL-m13n flow (various SBL-SpaRTA) and iso-contours of the longitudinal velocity (baseline model).

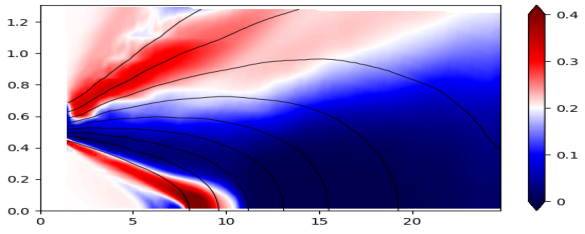
C.2 Model weights used for testing



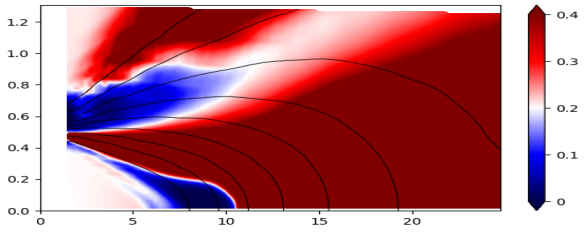
(a) w_{ZPG}



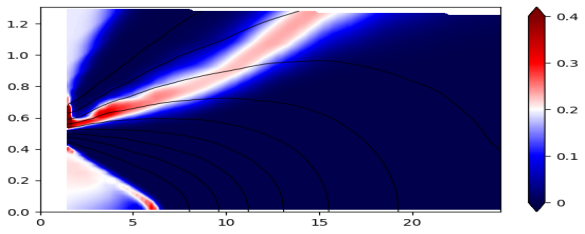
(b) w_{CHAN}



(c) w_{APG}

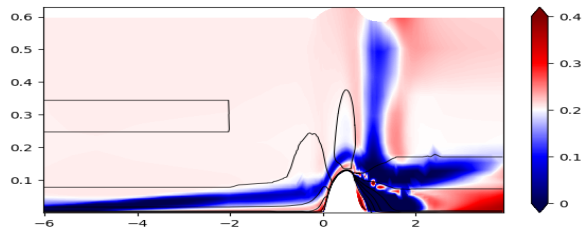


(d) w_{ANSJ}

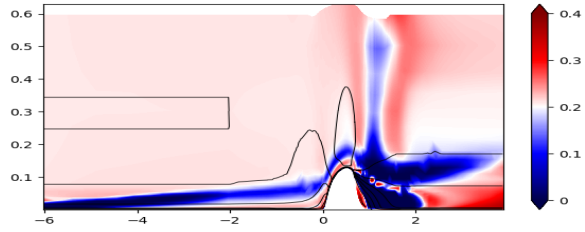


(e) w_{SEP}

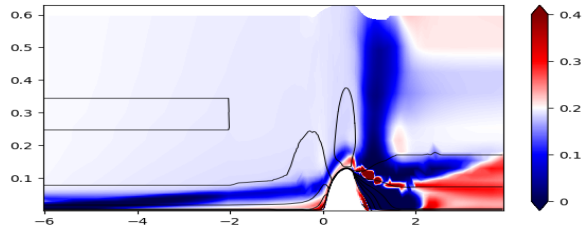
Figure C.7: Colormaps of exact optimal model weights for the ASJ flow (various SBL-SpaRTA) and iso-contours of the longitudinal velocity (baseline model).



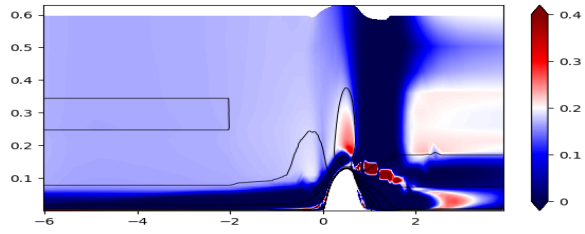
(a) w_{ZPG}



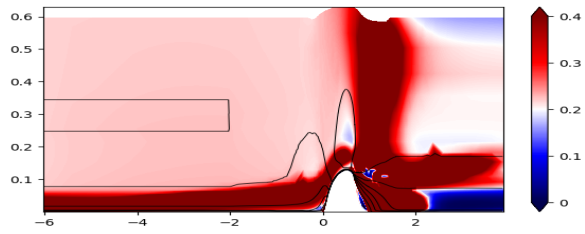
(b) w_{CHAN}



(c) w_{APG}



(d) w_{ANSJ}



(e) w_{SEP}

Figure C.8: Colormaps of exact optimal model weights for the 2DWMH flow (various SBL-SpaRTA) and iso-contours of the longitudinal velocity (baseline model).

Appendix D

Comparison of non-intrusive and intrusive X-MA (Chapter 6)

D.1 Complementary training results for the turbulent separated flows

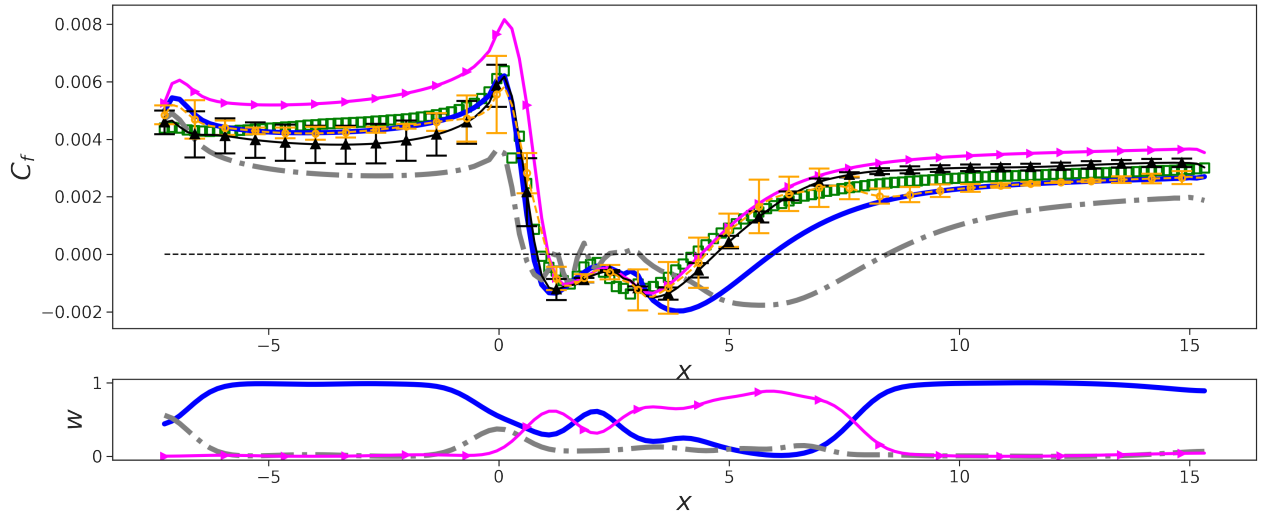
In this section of the Appendix, we display and comment on plots showing various QoI for two turbulent separated flows used in X-MA training, but not covered in Chapter 6. The analysis provides further insight into the behavior of the models under study.

Let's start with the Curved Backward-Facing Step (CBFS). Figure D.1 shows plots of the friction coefficient C_f, the horizontal velocity U, and the Reynolds shear stress τ_{xy} along the $x-$-axis. Regarding C_f, in Figure D.1a, the baseline $k-\omega$ SST model tends to overestimate the size of the recirculation bubble. On the other hand, the $\mathbf{M}^{(SEP)}$ model designed for separated flows accurately predicts the size and location of the separation bubble, but tends to overestimate the skin friction values in the flat plate segments before and after reattachment. In contrast, the $\mathbf{M}^{(ANSJ)}$ model underestimates the skin friction values and significantly overpredicts the size of the separation bubble. For this range of behavior, the non-intrusive prediction closely matches the high-fidelity data along the entire wall. Upstream, the baseline model receives the highest weight, which is consistent because this region features a flat plate with minimal pressure gradient, and the baseline model is known to perform well under such conditions. As the separation point approaches, the $\mathbf{M}^{(SEP)}$ model is activated and persists until reattachment

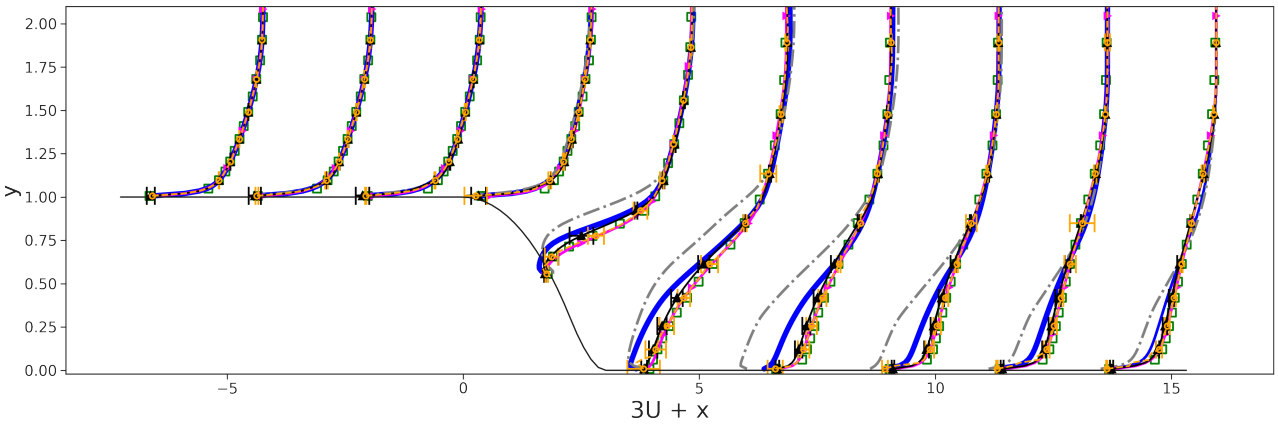
and beyond. Finally, after reattachment, the baseline model regains dominance in the flat plate segment. The $\mathbf{M}^{(ANSJ)}$ model shows a significant decrease along the wall, except near the domain inlet. This may be related to the onset of the boundary layer at the domain inlet. The interpretation of the model weights is less straightforward here and a feature mismatch is suspected.

The intrusive X-MA prediction follows a similar pattern, capturing the position and location of the recirculation bubble reasonably well. However, well upstream of the separation point, the skin friction values are slightly lower than the baseline model and the non-intrusive X-MA prediction. This observation is reminiscent of the behavior of the BSL-EARSM model in this region, as shown in Figure B.2b. We attribute this slight decrease in our case to the initial triggering of $\mathbf{M}^{(ANSJ)}$ near the entry boundary condition. While this is primarily a boundary effect, this small contribution of $\mathbf{M}^{(ANSJ)}$, albeit on a very short portion, is advected downstream and slightly contributes to a slight decrease in the skin friction values. However, this happens without significantly affecting the ability of the intrusive X-MA to capture the recirculation bubble. Another notable effect is observed after reattachment, where the friction coefficient prediction lags behind the high-fidelity data levels, in contrast to the relatively fast response of the non-intrusive X-MA prediction. Again, we attribute this latency to the downstream transport of the $\mathbf{M}^{(SEP)}$ correction as it adjusts the flow behavior after reattachment. In Figures D.1b and D.1c, we note a very good agreement of both intrusive and non-intrusive X-MA predictions of horizontal velocity and Reynolds shear stress with high-fidelity data over the entire physical domain.

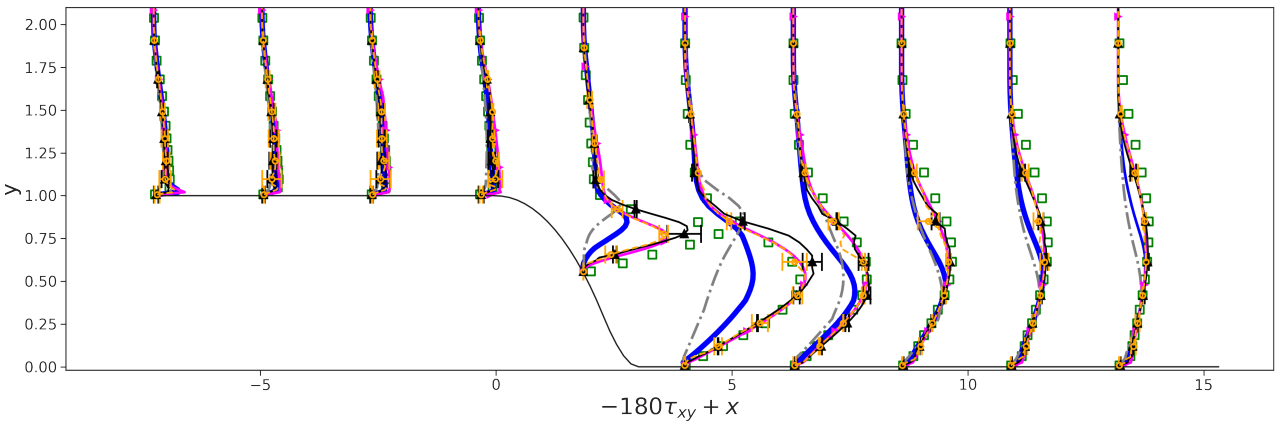
Next, in Figure D.2, we examine the Periodic Hills (PH). Using the same analytical approach, we again examine the friction coefficient, horizontal velocity, and Reynolds shear stress profiles. With respect to skin friction, both the intrusive and non-intrusive X-MA predictions agree very well with the trends shown by $\mathbf{M}^{(SEP)}$. This alignment is largely dictated by the values of the model weights near the wall, which are dominated by M_{SEP}, leading to accurate predictions of separation and reattachment. A small observation concerns a region near $x = 6$ and $x = 8.2$, where the intrusive X-MA prediction appears to be more resilient to the abrupt change in



(a) C_f.



(b) $3U + x$.

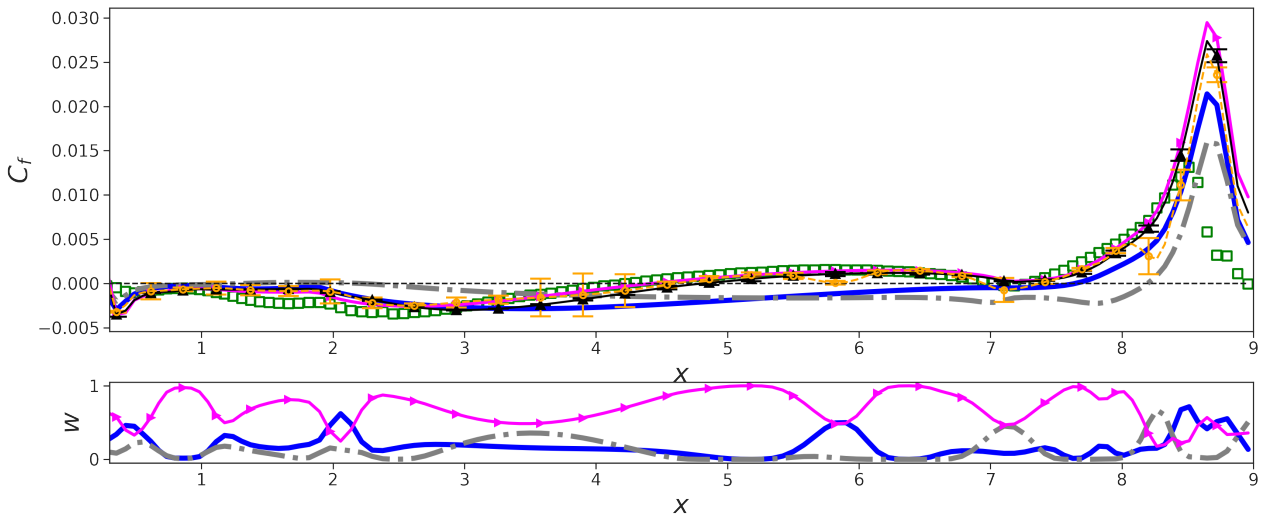


(c) $-180\tau_{xy} + x$.

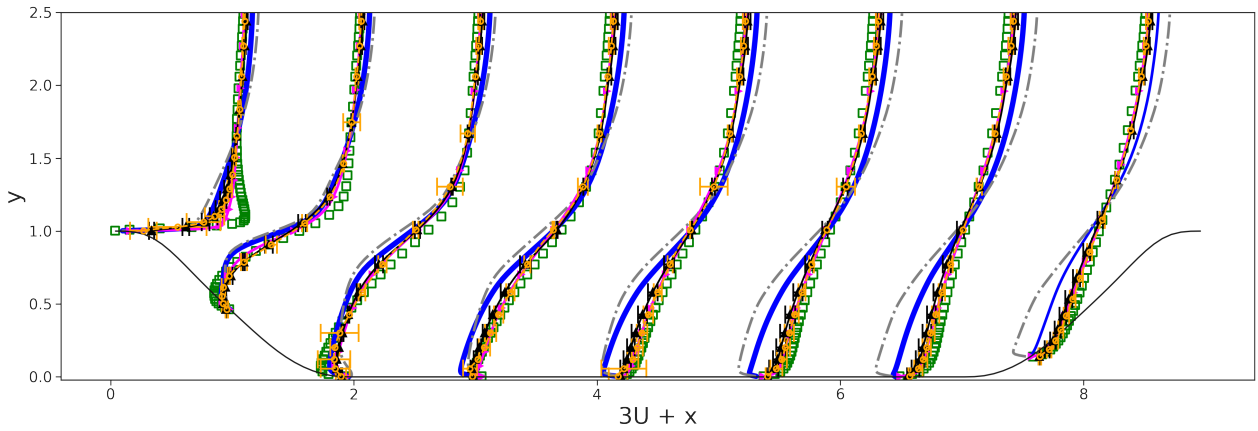
Figure D.1: Horizontal velocity U and Reynolds shear stresses τ_{xy} at various x positions for the CBFS flow case. Baseline $k-\omega$ SST (—); $\mathbf{M}^{(ANSJ)}$ (-·-); $\mathbf{M}^{(SEP)}$ (-▸-); High-fidelity data (□); Non-intrusive X-MA (-⧳-); Intrusive X-MA (-⧳-).

model weights, showing a smoother response, while the non-intrusive X-MA prediction shows a slight "wiggling" perturbation. We believe that this difference is due to the intrusive X-MA's inherent advantage in handling abrupt changes in model weights, as the resulting SBL-SpaRTA composite model is fed into the CFD solver, which is able to mitigate and smooth these perturbations, resulting in a smoother response in the intrusive X-MA case. In terms of horizontal velocity and Reynolds shear stress, both the non-intrusive and intrusive X-MA predictions follow the trends observed in the high-fidelity data and $\mathbf{M}^{(SEP)}$. In the outer region of the recirculation bubble, the predictions of both paradigms show a slight decrease in performance compared to the $\mathbf{M}^{(SEP)}$ model. However, they remain significantly superior to the baseline $k\omega$ SST model for both QoIs.

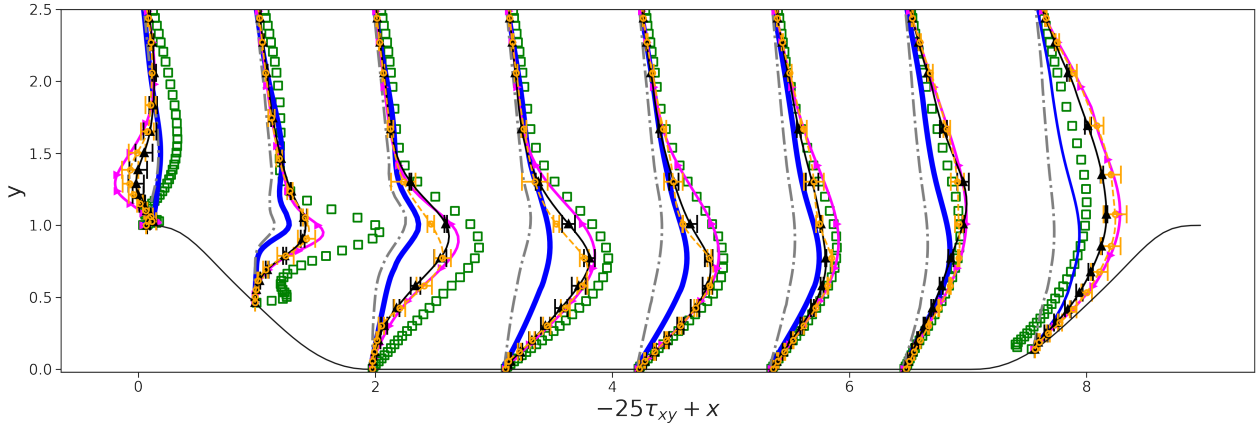
Overall, these results on these training cases highlight the potential of the intrusive X-MA approach to deliver improved results, despite the observed variation in performance across different regions of the flow.



(a) C_f.



(b) $3U + x$.



(c) $-25\tau_{xy} + x$.

Figure D.2: Friction coefficient C_f, horizontal velocity U and Reynolds shear stresses τ_{xy} at various x positions for the PH flow case. $\mathbf{M}^{(CHAN)}$ (—); $\mathbf{M}^{(ANSJ)}$ (-·-·-); $\mathbf{M}^{(SEP)}$ (—►—); High-fidelity data (□); Non-intrusive X-MA (-↕-); Intrusive X-MA (—↕—).

D.2 Training results using optimal $w_{\tau_{xy}}$

In this section of the Appendix, we provide additional details about the procedure used to train both intrusive and non-intrusive X-MA. Specifically, in Table D.1 we present the $Imp_{QoI}(\%)$ when using τ_{xy} as the QoI for constructing the model weights used for aggregations. As noted in the 6 chapter, the improvements observed with this choice are not as significant as those achieved with U (w_U). In particular, optimizing $w_{\tau_{xy}}$ to approximate high-fidelity values of Reynolds shear stresses does not necessarily yield a comparable level of improvement in approximating the high-fidelity values of U (U^{HF}) using $w_{\tau_{xy}}$ for aggregation, as opposed to using w_U. In many cases, there is only a small improvement or even a deterioration in the aggregated prediction of U. This again raises the question of conditioning the RANS equations.

case	$2\sigma_w^2(U)^*$	QoI	Non-intrusive X-MA	Intrusive X-MA	$\mathbf{M}^{(CHAN)}$	$\mathbf{M}^{(ANSJ)}$	$\mathbf{M}^{(SEP)}$
CHAN	1	U	-1335.6	-2.8	0	-15654	-319.3
		τ_{xy}	79.3	1.3	0	-90.6	-247
ANSJ	10^{-1}	U	12.0	61.3	0	78.7	-260.7
		τ_{xy}	83.3	47.7	0	78.7	-334.3
CD	10^{-1}	U	-3.5	66.6	0	-642.3	31.3
		τ_{xy}	39.6	46.7	0	-198.3	31.1
CBFS	10^{-2}	U	41.5	72.8	0	-393.2	93.6
		τ_{xy}	82.0	79.4	0	-84.7	83.5
PH	10^{-1}	U	29.8	69.5	0	-132.8	83.3
		τ_{xy}	64.3	51.7	0	-70.4	27.6

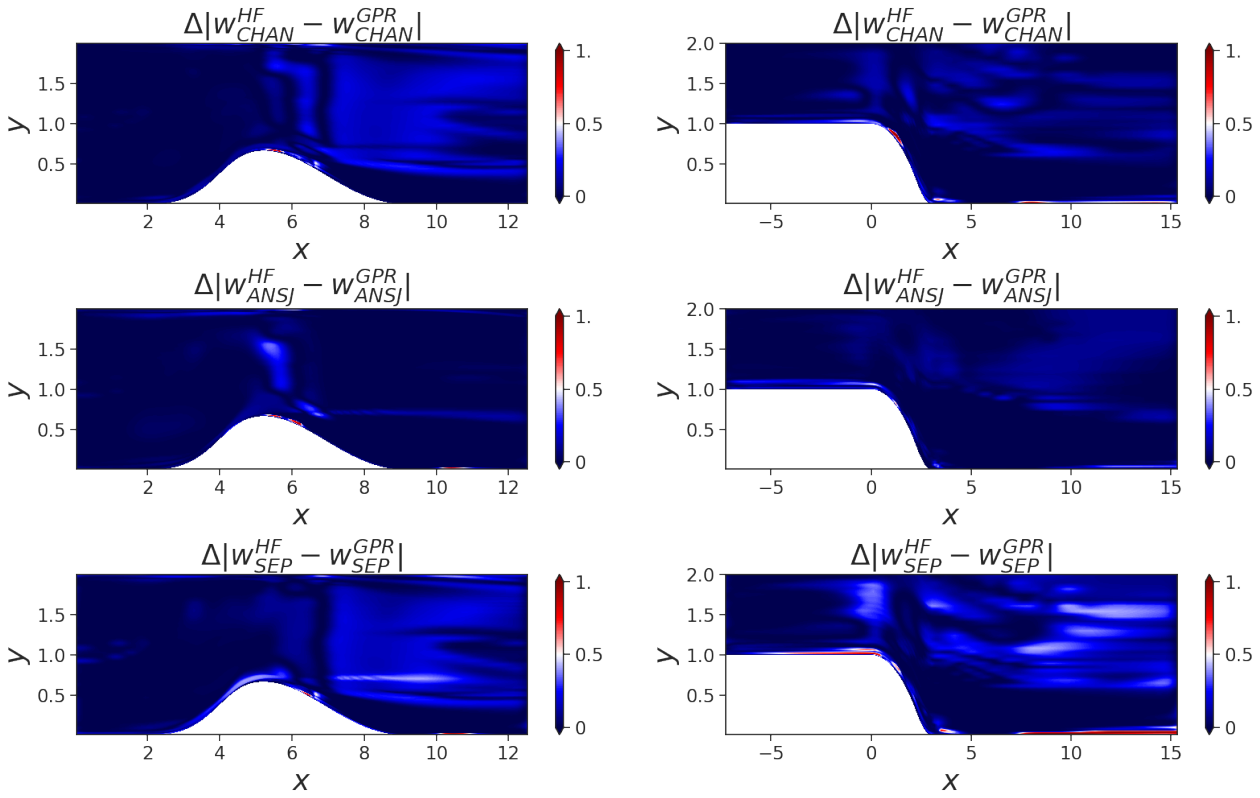
Table D.1: Improvements in (%) wrt the baseline k-ω SST on training cases using the optimal $w_{\tau_{xy}}$

D.3 Errors in the regression of model weights

For validation purposes, a key emphasis is placed on the regressor's capacity to effectively predict the model's weights in both training and unseen test cases. In the following, we recall that for both training and test, the model weights are obtained by providing the GPR with local flow features calculated using the baseline $k-\omega$ SST model.

D.3.1 Training errors

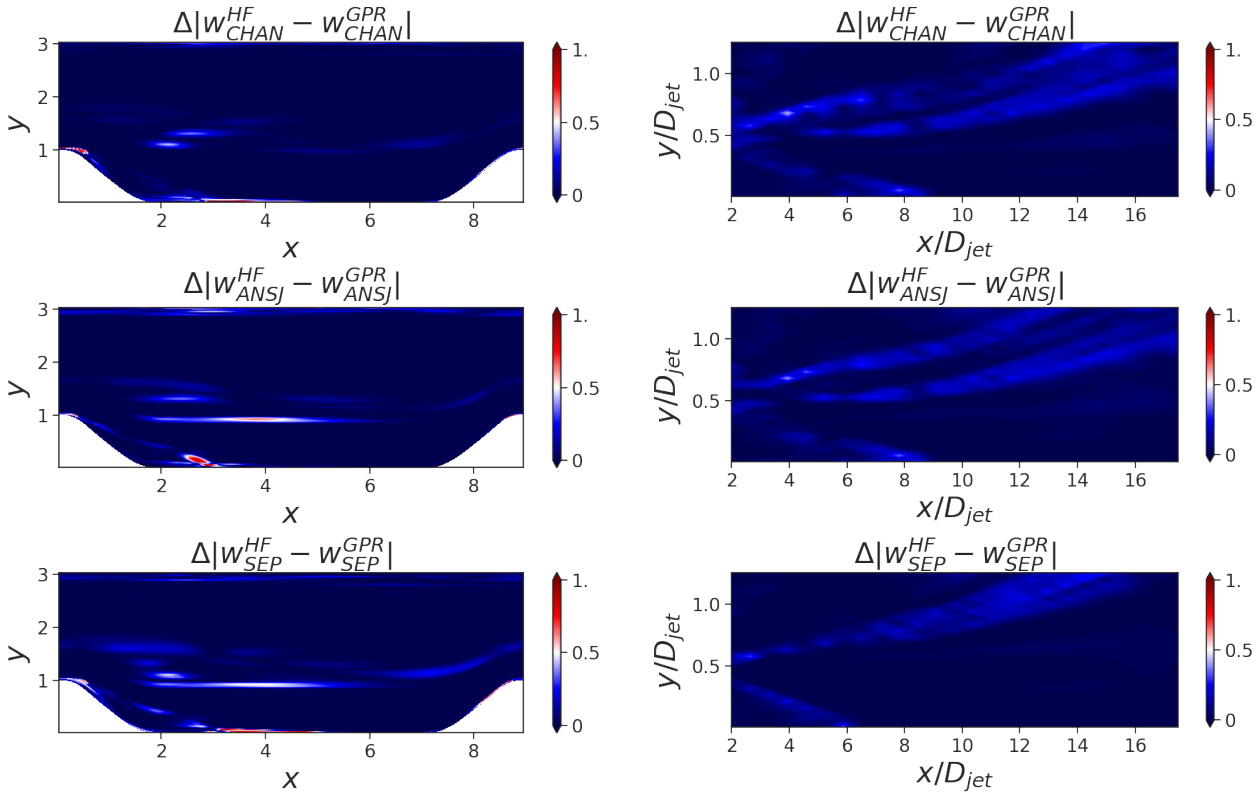
In Figures D.3 and D.4, the difference between the optimal model's weights (w^{HF}), calculated using high-fidelity values of horizontal velocity and the offline converged solution of the 3 considered SBL-SpaRTA, and the model weights predicted by the GPR (w^{GPR}), is calculated across the computational domain of 4 training cases. The GPR-predicted weights fields closely match their training values, with some distortions stemming from the regressor's architecture.



(a) $\Delta|w^{GPR} - w^{HF}|$ for CD

(b) $\Delta|w^{GPR} - w^{HF}|$ for CBFS

Figure D.3: Regression errors of model weights using GPR on the CD and CBFS training cases.



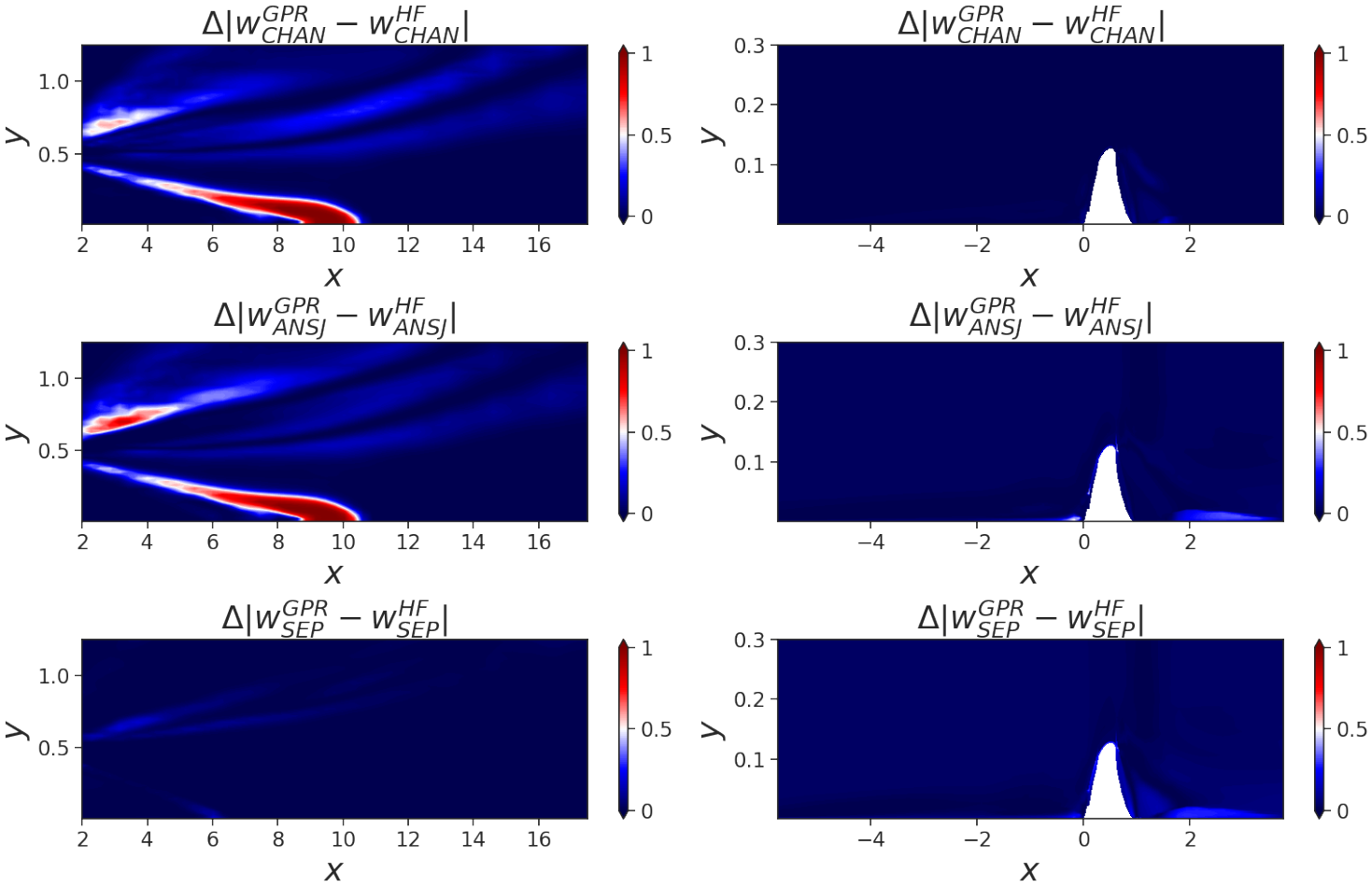
(a) $\Delta|w^{GPR} - w^{HF}|$ for PH

(b) $\Delta|w^{GPR} - w^{HF}|$ for ANSJ

Figure D.4: Regression errors of model weights using the GPR regressor on PH and ANSJ training cases.

D.3.2 Test errors

We present the model weights predictions for two unseen flow cases used for testing: ASJ and 2DWMH. Figure D.5 reveals overall minimal discrepancies across the flow field for both cases. However, for the ASJ case there is a noticeable difference in a section of the free shear layer for both $w_{(ANSJ)}$ and $w_{(CHAN)}$. The observed difference might have its reason in a potential mismatch in features within this specific area. This mismatch can be influenced by variations in operating conditions between the test and training cases, even though they have a similar geometry. Similarly, in the 2DWMH case, a noteworthy discrepancy is observed primarily in a section of the boundary layer after reattachment for $w_{(ANSJ)}$ and $w_{(SEP)}$. Let's not forget that the training dataset includes separated flow cases under moderate Reynolds numbers of approximately $Re \simeq 10^4$. In contrast, the test case stands out with a notably higher Reynolds number of $Re = 80 \times 10^6$. This indicates the necessity for future research to delve more deeply into the derivation of features as well as including a wider range of operating conditions during the training.



(a) $\Delta|w^{GPR} - w^{HF}|$ for ASJ

(b) $\Delta|w^{GPR} - w^{HF}|$ for 2DWMH

Figure D.5: Regression errors of model weights using GPR on the ASJ and 2DWMH test cases.

Appendix E

Uncertainty Quantification (UQ) budget

In this Appendix, we present the budget required for uncertainty quantification in all the conducted studies, mainly comparing the Point Collocation Method and the sparse Polynomial Chaos Expansion (PCE) in this analysis. By "budget", we refer to the number of RANS simulations necessary to obtain the desired results, which in this case are either the uncertainty bars or the Sobol indices. Across all the tables, the superiority of the sparse PCE over the Point Collocation Method is evident.

Point Collocation method	Sparse PCE
$3^4 = 81$	15

Table E.1: Sobol indices calculation budget per flow case.(Chapter 4)

	Point Collocation method	Sparse PCE
ZPG	$3^2 = 9$	4
CHAN	$3^0 = 1$	1
APG	$3^1 = 3$	3
ANSJ	$3^1 = 3$	3
SEP	$3^2 = 9$	4
Total samples per case	25	15

Table E.2: Non-intrusive X-MA UQ budget per flow case in the case of aggregating 5 models' (ZPG, CHAN, APG, SEP and ANSJ) predictions.(Chapter 5)

	Point Collocation method	Sparse PCE
Non-intrusive X-MA	$3^0 + 3^2 + 3^1 = 13$	$1 + 4 + 3 = 8$
Intrusive X-MA	$3^3 = 27$	10

Table E.3: Non-intrusive and intrusive X-MA UQ budget (number of calculations needed) per flow case in the case of aggregating 3 models: CHAN, SEP and ANSJ. (Chapter 6)

	Point Collocation method	Sparse PCE
Non-intrusive X-MA	$3^2 + 3^0 + 3^1 + 3^2 + 3^1 = 25$	$4 + 1 + 3 + 4 + 3 = 15$
Intrusive X-MA	$3^6 = 729$	28

Table E.4: Non-intrusive and intrusive X-MA UQ budget (number of calculations needed) per flow case in the case of aggregating 5 models: ZPG, CHAN, APG, SEP and ANSJ.

Soufiane CHERROUD

Sparse Bayesian Learning and Adaptative Aggregation of Turbulence Models

L'objectif de cette thèse est l'amélioration des modèles de turbulence RANS existants par le biais de l'apprentissage automatique (ML). Elle se structure en trois volets principaux. Tout d'abord, l'algorithme de l'apprentissage bayésien parcimonieux (SBL) est utilisé pour identifier des fermetures parcimonieuses et stochastiques de type EARSM pour le modèle $k-\omega$ SST. Cela vise à traiter les lacunes de ce modèle dans la prédiction des écoulements turbulents séparés. Les modèles ainsi obtenus, appelés SBL-SpaRTA, se caractérisent par leur interprétabilité, leur invariance galiléenne et leur capacité à améliorer les prédictions par rapport au modèle de base, tout en fournissant des intervalles de confiance autour des prédictions. Cependant, leur généralisation à d'autres types d'écoulements est limitée. Cette limitation constitue la motivation de la deuxième partie de la thèse, où le cadre précédemment développé est utilisé pour créer des modèles SBL-SpaRTA personnalisés pour divers cas d'écoulement types, notamment des plaques planes, des écoulements séparés et des jets. Ensuite, un régresseur ML est entraîné pour attribuer automatiquement des poids locaux aux prédictions de chaque modèle, en fonction de leur vraisemblance et de la physique locale de l'écoulement. Bien que cette approche "non intrusive" se distingue par sa capacité de généralisation et son amélioration significative par rapport au modèle de base, ses prédictions n'adhèrent pas nécessairement aux équations de conservation. Enfin, dans la troisième partie, une alternative est proposée en appliquant une méthodologie intrusive pour le mélange des modèles. Les modèles SBL-SpaRTA personnalisés sont ainsi intégrés et combinés automatiquement dans le code CFD. Les résultats sont comparés à l'approche non intrusive, permettant ainsi d'évaluer les avantages et les limites de chaque méthode.
Mots clés : Modélisation de turbulence, Apprentissage automatique, Apprentissage bayésien parcimonieux, Modèles EARSM, Écoulements séparés, Écoulements de jet, Couches limites, Analyse de sensibilité, Mélange de modèles d'experts, Mélange de modèles.

This PhD thesis aims to enhance the current RANS turbulence models using Machine Learning (ML), and is organized in three main parts. First, the Sparse Bayesian Learning (SBL) algorithm is used to derive sparse and stochastic EARSM-type closures for the baseline $k-\omega$ SST model, to address turbulent separated flows. The resulting models, denoted SBL-SpaRTA models, are interpretable, Galilean frame-invariant, and enable improved velocity and friction coefficient predictions compared to the baseline, while providing confidence intervals around the predictions. While effective on their training flow category, these models show weak generalizability. This motivates the second part of the thesis where the precedent framework is used to derive customized SBL-SpaRTA for a set of typical flow cases comprising flat plates, separated flows and jets. Then, a ML regressor is trained to automatically attribute local weights to the predictions of every model, reflecting its likelihood and knowing the local underlying physics. While this "non-intrusive" approach exhibits good generalizability and substantial enhancements over the baseline model for both training and unseen test cases, its final prediction does not necessarily adhere to the conservation equations. Finally, in the third part, this issue is addressed by applying an intrusive methodology for model aggregating, where the customized SBL-SpaRTA are automatically blended in the CFD code using ML. This framework is compared to the non-intrusive paradigm using a systematic methodology, thus enabling to evaluate their merits and drawbacks.
Keywords : Turbulence modeling, Machine Learning, Sparse Bayesian Learning, Explicit Algebraic Reynolds Stress models, Separated flows, jet flows, boundary layers, Sensitivity analysis, Mixture-of-Experts, Model Aggregation.