



### **Science Arts & Métiers (SAM)**

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>  
Handle ID: <http://hdl.handle.net/10985/23747>

#### **To cite this version :**

Soufiane CHERROUD, Xavier MERLE, Paola CINNELLA, Xavier GLOERFELT - Sparse Bayesian Learning of Explicit Algebraic Reynolds-Stress models for turbulent separated flows - International Journal of Heat and Fluid Flow - Vol. 98, p.19 - 2022

Any correspondence concerning this service should be sent to the repository

Administrator : [scienceouverte@ensam.eu](mailto:scienceouverte@ensam.eu)



# Sparse Bayesian Learning of Explicit Algebraic Reynolds-Stress models for turbulent separated flows

Soufiane CHERROUD<sup>a</sup>, Xavier MERLE<sup>a</sup>, Paola CINNELLA<sup>a,b</sup>, Xavier GLOERFELT<sup>a</sup>

<sup>a</sup>*DynFluid Laboratory, Arts et Metiers ParisTech, 151 Bd de l'Hopital, Paris, 75013, France*

<sup>b</sup>*Institut Jean Le Rond d'Alembert, Sorbonne Universite, 4 place Jussieu, Paris, 75252, France*

---

## Abstract

A novel Sparse Bayesian Learning (SBL) framework is introduced for generating parsimonious stochastic algebraic stress closures for the Reynolds-Averaged Navier–Stokes (RANS) equations from high-fidelity data. The models are formulated as physically-interpretable frame-invariant tensor polynomials and built from a library of candidate functions. By their stochastic formulation, the learned model coefficients are described by probability distributions and are therefore equipped with an intrinsic measure of uncertainty. The SBL framework is used to derive customized stochastic closure models for three separated flow configurations, characterized by different geometries but similar Reynolds number. The resulting SBL models are then propagated through a CFD solver for all three configurations. The results show significantly improved predictions of velocity profiles and friction coefficient in the separation / reattachment region in comparison with a baseline LEVM (namely, the  $k - \omega$  SST model), for training as well as for test cases. In all cases, the computed uncertainty intervals encompass reasonably well the reference data. Furthermore, the stochastic outputs enable a global sensitivity analysis with respect to the model terms selected by the algorithm, thus providing insights in view of further improvements of EARSM-type corrections.

**Keywords:** Turbulence modelling, Machine Learning, Sparse Bayesian Learning, Explicit Algebraic Reynolds-stress models, Separated flows, Sensitivity analysis

**PACS:** 0000, 1111

**2000 MSC:** 0000, 1111

---

## 1. Introduction

Despite the growth in computing power, high-fidelity turbulent flow simulations such as Direct Numerical Simulation (DNS) or Large Eddy Simulation (LES) remain too expensive for daily-use in industrial applications, and engineering design and optimization mostly rely on Reynolds Averaged Navier-Stokes (RANS) equations supplemented by suitable closure models. The most widely used turbulence models, often referred-to as LEVM (Linear Eddy-Viscosity Models), are based on the so-called Boussinesq assumption of alignment between the Reynolds stress and the mean strain rate tensors. Such condition is however not verified even for relatively simple flows (see [1, 2]). Besides, the eddy viscosity coefficient is computed via auxiliary relations (often transport equations for turbulent properties) introducing a number of supplementary modeling hypotheses and closure coefficients. Finally, more advanced models than LEVM exist, but they also suffer from modeling approximations and uncertain closure parameters (see [3] for a recent review). 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; iii) the calibration of the model closure parameters, known as parametric uncertainty. The first point is most often evaluated from expert judgment, based on a trade-off between targeted accuracy and computational resources. In the following, we focus on points ii) and iii).

The ever increasing amount of high-fidelity simulations and experimental data available for diverse flow configurations has recently fostered the use of data-assimilation (DA) and machine learning (ML) techniques for informing turbulent closures (see review papers [3, 4, 5]). Early studies mostly deal with the quantification of uncertainties associated with turbulence models by using interval analysis or statistical inference tools. The analysis is conducted either by perturbing directly the Reynolds stress anisotropy tensor computed with a baseline LEVM [6, 7, 8] or by associating probability distributions to the model closure coefficients [9, 10, 11, 12]. While the first approach accounts for structural uncertainties, the second does not. In turn, the first approach is intrusive in the sense that its implementation involves modifications of the RANS solver, while in the second one model parameters can be fed as inputs to the CFD solver.

As an attempt to quantify model-form uncertainty in a probabilistic frame-

work, early work was conducted by [13] by using Demster-Shafer evidence theory, which uses multiple models for predicting a given flow configuration. More recently, [14] explored a Bayesian framework christened 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 [15] and compressor cascades [16, 17]. A shortcoming of multi-model methods is that prediction accuracy cannot be better than the best model in the considered set, even if it outperforms the worst one.

In the attempt of overcoming structural deficiencies intrinsic to turbulence modeling assumptions, data-driven methods for turbulence modeling based on supervised machine learning have been introduced in recent years. Some of the earliest examples can be found in [18] where the uncertainty quantification framework of [6] is used to parametrize stochastic Reynolds stress discrepancies as a function of local mean-flow variables, with extended Kernel regression ML being leveraged to train the function. In [19] the authors extended the early work of [18], and used a field inversion technique relying on adjoints to learn a corrective multiplicative term for the turbulent transport equations, as well as its variance. ML was then used to express the correction as a black-box function of selected flow features and to extrapolate it to a class of flows similar to the training one. The approach allows extrapolation of the learned model correction beyond the training configuration. However, the corrected model remains a LEVM. Using a similar framework, Xiao et al. [20, 21] (see also [22]) 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. [23] 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.

In the quest for reducing turbulence modeling errors, more recent research work has mostly leveraged ML techniques for learning improved turbulence mod-

els from data, while the quantification of uncertainties associated with the predictions has been overshadowed. Seminal work of Ling et al. [24] introduced a novel neural network architecture (tensor basis neural network, TBNN) that allows for embedding frame invariance constraints within the learned 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 Pope [25], leading to a form of generalized Explicit Algebraic Reynolds Stress model, whose function coefficients are regressed from high-fidelity data using ML. Both Deep Neural Networks (DNN) and Random Forests (RF) can then be used for this task [24, 26, 27]. In the same line, a general framework for producing data-augmented turbulence models is proposed by Wu et al. [28], who build on the tensor basis decomposition of Ling et al. by introducing a larger set of frame-invariant features. In this class of methods, uncertainty intervals on the predictions are generally excluded from considerations. An exception is represented by Ref. [29], who used Mondrian Forests to infer stochastic Reynolds-stress corrections, but the applicability of the approach to unobserved flows remains to be investigated.

The preceding approaches make use of so-called black-box ML, such as DNN or RF. They enable flexible approximation of complex functional relations but do not provide any explicit, physically interpretable, mathematical expression for the learned correction. Furthermore, their implementation within a CFD solver to make predictions of new flows may be delicate. An interesting alternative is represented by so-called open-box ML approaches, which select explicit mathematical expressions and/or operators from a pre-defined library to build a suitable regressor for the data. Examples include Genetic Programming (GEP) [30] and symbolic identification [31, 32]. Once again, the Reynolds-stress anisotropy is projected onto a minimal integrity basis and ML is used to regress the function coefficients of the decomposition. Corrective terms for the turbulent transport equations are also developed using a similar tensor decomposition. Sparsity-promoting formulations of the cost function are used in [31, 32] to minimize the number of active terms and limit the occurrence of overfitting. The resulting models correspond now 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 [33, 34, 35, 36, 37, 38].

Despite the progress made in developing data-driven turbulence model corrections, 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. The identification of flow regions of greater sensitivity to turbulence modeling errors also represents valuable information for designers. For this reason, in this work we explore a novel efficient methodology for developing data-augmented RANS models with quantified uncertainty intervals.

Specifically, we propose a novel algorithm for generating stochastic parsimonious algebraic stress models for the closure of RANS equations from high-fidelity CFD data. For this purpose we resort to Bayesian inference, owing to its natural ability to quantify uncertainty in parameter estimates. Furthermore, Bayesian estimation methods are less prone to overfitting, a common problem when learning from scarce data. Specifically, the proposed method in this paper makes use of the Sparse Bayesian Learning (SBL) approach initially proposed by [39] and recently applied to the discovery of hidden differential equations for dynamical systems [40, 41]. A nonlinear extension of SBL in the context of aeroelasticity problems has been proposed in [42]. The principle behind the SBL is similar to that of regularized regression used in symbolic identification methods. The algorithm yields sparse solutions as it relies on prior coefficient distributions that are sharply peaked around zero. Contrary to the deterministic sparse identification approach, however, SBL delivers probability distributions of the parameters.

The starting point of the present approach is similar to the SpaRTA (Sparse Regression of Turbulent Stress Anisotropy) algorithm of [31]. A general EARSM model involving a very large number of features is first constructed, based on Pope’s generalized eddy viscosity model. In the present stochastic formulation, however, parameters are treated as random variables with associated prior probability distributions. Differently from SpaRTA, –which uses a fully deterministic regularized regression algorithm and requires separate steps for model identification (selection of terms) and parameter estimation– here we leverage the SBL algorithm [39, 43], supplemented by demi-Bayesian LASSO hierarchical priors [43], to simultaneously select a small number of relevant terms and estimate posterior probability distributions for their associated parameters. The latter can be propagated back through the model, providing confidence intervals on predictions. The algorithm finally delivers stochastic EARSM models, which can be seen as the stochastic counterpart of deterministic EARSM developed in [31]. The discovered models are propagated through a RANS solver by using a suitable uncertainty quantification method. This results in estimates of the statistical moments of the solution, but also of global sensitivity indices of the computed flow to the selected

coefficients. The proposed approach is thoroughly assessed for a set of separated flow configurations, previously investigated in [31], which enables comparisons.

The rest of the paper is organized as follows. In Section 2, we introduce the turbulence model learning problem. In Section 3, we recall the general formulation of the Sparse Bayesian Learning algorithm and we propose an adaptation to the present learning task. In Section 4 we present the flow configurations used in the numerical experiments and we discuss the results. Concluding remarks are reported in Section 5.

## 2. Learning RANS closures

### 2.1. Problem formulation

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

$$\frac{\partial U_i}{\partial x_i} = 0 \quad (1)$$

$$U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \left( \nu \frac{\partial U_j}{\partial x_i} - \tau_{ij} \right) \quad (2)$$

with  $U_i$  the  $i$ -th mean velocity component,  $P$  the mean pressure,  $\rho$  the fluid density and  $\nu$  the kinematic viscosity. Equations (1) are supplemented with a LEVM model for the kinematic Reynolds stress tensor  $\tau_{ij} = \langle u'_i u'_j \rangle$ , with  $u'_i$  the  $i$ -th fluctuating velocity component.

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

$$\tau_{ij} = \underbrace{\frac{2k}{3} \delta_{ij}}_{\text{isotropic}} + \underbrace{a_{ij}}_{\text{anisotropic}} \quad (3)$$

(where  $k = \text{tr}(\tau_{ij})/2$  is the turbulent kinetic energy and  $\delta_{ij}$  is the Kronecker symbol), a LEVM is obtained by assuming that the Reynolds stress anisotropy  $a_{ij}$  is a linear function of the mean velocity gradient (Boussinesq hypothesis):

$$a_{ij} = -2\nu_t S_{ij} = 2kb_{ij}^0 \quad (4)$$

where  $b_{ij}^0$  is the normalized anisotropy tensor of the Boussinesq model. Most often, the eddy viscosity coefficient  $\nu_t$  is computed by solving auxiliary transport equations for well-chosen turbulent properties, such as in the  $k - \omega$  SST model

[44] used in the following of this study.

Following [31], we seek to correct the Boussinesq constitutive model for  $b_{ij}$  by introducing a second-order symmetric and traceless tensor  $\mathbf{b}^\Delta = \{b_{ij}^\Delta\}$ , referred-to as the extra-anisotropy, such that:

$$\tau_{ij} = 2k \left( \frac{1}{3} \delta_{ij} + b_{ij}^0 + b_{ij}^\Delta \right) \quad (5)$$

Based on [45, 31], 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 expressed in a minimal integrity basis of ten tensors and five invariants for the general case of 3D flow [25]:

$$b_{ij}^\Delta = \sum_{n=1}^{10} T_{ij}^{(n)} \alpha_n^\Delta(I_1, \dots, I_5) \quad (6)$$

For the 2D flows considered in this work, higher-order terms are identically zero and the basis reduces to a set of three tensors and two invariants only:

$$b_{ij}^\Delta = \sum_{n=1}^3 T_{ij}^{(n)} \alpha_n^\Delta(I_1, I_2) \quad (7)$$

where :

$$\begin{cases} T_{ij}^{(1)} = S_{ij}^* \\ T_{ij}^{(2)} = S_{ik}^* \Omega_{kj}^* - \Omega_{ik}^* S_{kj}^* \\ T_{ij}^{(3)} = S_{ik}^* S_{kj}^* - \frac{1}{3} \delta_{ij} S_{mn}^* S_{mn}^* \\ I_1 = S_{mn}^* S_{mn}^* \\ I_2 = \Omega_{mn}^* \Omega_{mn}^* \end{cases} \quad \text{and} \quad \begin{cases} S_{ij}^* = \frac{1}{\omega} \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \\ \Omega_{ij}^* = \frac{1}{\omega} \left( \frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i} \right) \end{cases} \quad (8)$$

$\mathbf{S}^* = \{S_{ij}^*\}$  is the non-dimensional strain rate tensor and  $\mathbf{\Omega}^* = \{\Omega_{ij}^*\}$  the non-dimensional rotation rate tensor.

In the following, the functions  $\alpha_n^\Delta$  are sought by a supervised Machine Learning procedure whose first step is to extract the LEVM model error with respect to the high-fidelity data, *i.e.* :

$$b_{ij}^{\Delta, hf} = b_{ij}^{hf} - b_{ij}^0 \quad (9)$$



Furthermore, the model-form errors in the auxiliary turbulent transport equations must also be taken into account. For that purpose we use the  $k$ -corrective-frozen methodology of [31]. The latter consists in solving the turbulent transport equations with frozen high-fidelity values for all quantities except  $\omega$ :

$$\begin{cases} \frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = P_k + R^{hf} - \beta^* k \omega + \frac{\partial}{\partial x_j} \left( (\nu + \sigma_k \nu_t) \frac{\partial k}{\partial x_j} \right) \\ \frac{\partial \omega}{\partial t} + U_j \frac{\partial \omega}{\partial x_j} = \frac{\gamma}{\nu_t} (P_k + R^{hf}) - \beta^* \omega^2 + \frac{\partial}{\partial x_j} \left( (\nu + \sigma_\omega \nu_t) \frac{\partial \omega}{\partial x_j} \right) \end{cases} \quad (10)$$

In the preceding equations,  $k$ ,  $U$ , etc. 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:

$$P_k = \min \left( 2k \left( -\frac{\nu_t}{k} S_{ij} + b_{ij}^{\Delta, hf} \right) \frac{\partial U_i}{\partial x_j}, 10\beta^* \omega k \right) \quad (11)$$

and an additional corrective term  $R$  has been introduced in the equations for  $k$  and  $\omega$ . Finally,  $\beta^*$ ,  $\sigma_k$  and  $\sigma_\omega$  are the  $k - \omega$  SST constants and can be found in [44].

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

$$R^{hf} \approx R = 2kb_{ij}^R \frac{\partial U_i}{\partial x_j} \quad (12)$$

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 integrity basis previously used for  $\mathbf{b}^\Delta$ :

$$b_{ij}^R = \sum_{n=1}^3 T_{ij}^{(n)} \alpha_n^R(I_1, I_2) \quad (13)$$

thus introducing a new set of unknown functions  $\alpha_n^R$  that are sought by the same machine learning procedure as the  $\alpha_n^\Delta$ , using now  $R^{hf}$  as the learning target.

## 2.2. Function dictionaries

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

$$\mathcal{B} = \{I_1^l, I_2^m, I_1^p I_2^q | 0 \leq l, m \leq 9, 2 \leq p + q \leq 4\} \quad (14)$$

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

Of note, hereafter we prefer to learn  $a_{ij}^\Delta = \tau_{ij} - (\frac{2}{3}k - \nu_t S_{ij}) = 2kb_{ij}^\Delta$  rather than  $b_{ij}^\Delta$  as in [46] because the value of  $b_{ij}^\Delta$  at the wall is mathematically undetermined:

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

making use of data and physical interpretation of the results difficult close to the wall. By multiplying  $b_{ij}^\Delta$  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:

$$\begin{cases} \mathbf{a}^\Delta = \mathbf{C}_{b^\Delta} \mathbf{w}_{b^\Delta} \\ \mathbf{R} = \mathbf{C}_R \mathbf{w}_R \end{cases} \quad (15)$$

where:

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

### 3. Sparse Bayesian Learning algorithm

This section provides details the SBL algorithm for regression, introduced by [39] and applied to the specific case of SVM (Support Vector Machines), and adapt it to our specific learning problem.

### 3.1. Inference

First, let us consider a data set of input-output pairs  $\{x_n, t_n\}_{n=1, \dots, N}$ . We follow the standard probabilistic formulation considering that the targets  $\mathbf{t} = (t_1, \dots, t_N)^T$  are sampled from a linear model  $\{\mathbf{C}; \mathbf{w}\}$  with additive noise  $\boldsymbol{\epsilon}$ :

$$\mathbf{t}(\mathbf{x}; \mathbf{w}) = \mathbf{C}(\mathbf{x})\mathbf{w} + \boldsymbol{\epsilon} = \sum_{i=1}^M \mathbf{C}_i(\mathbf{x})w_i + \boldsymbol{\epsilon} \quad (16)$$

where  $\mathbf{C}(\mathbf{x}) = (\mathbf{C}_1(\mathbf{x}), \dots, \mathbf{C}_M(\mathbf{x}))$  is the design matrix,  $\mathbf{w} = (w_1, \dots, w_M)^T$  is the vector of the model parameters, and  $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_N)^T$  is a vector of independent noise processes assumed to be Gaussian of zero-mean and variance  $\sigma^2$ . Following these assumptions, one can derive the likelihood of observing the data given the model parameters  $\mathbf{w}$  and  $\sigma^2$ :

$$p(\mathbf{t}|\mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{t} - \mathbf{C}\mathbf{w}\|^2\right) \quad (17)$$

The parameters are traditionally determined by maximizing the logarithmic likelihood of observing the data knowing the model parameters, *i.e.*

$$\mathcal{L}_{\mathcal{I}} = \log p(\mathbf{t}|\mathbf{w}, \sigma^2) = -\frac{1}{2} \left( N \log(2\pi\sigma^2) + \frac{1}{\sigma^2} \|\mathbf{t} - \mathbf{C}\mathbf{w}\|^2 \right) \quad (18)$$

Nevertheless, with as many parameters in the model as training examples, the maximum-likelihood is expected to lead to severe overfitting, *i.e.* that the model is catching the data's noise rather than its proper dynamics.

The idea of [39], motivated by the work of [47] is to constrain the model parameters by defining an explicit prior distribution over them. The principle is similar to the Automatic Relevance Determination (ARD) initially introduced by [48, 49]. We follow the choice made in [39] of zero-mean Gaussian prior distribution over  $\mathbf{w}$ :

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=1}^M \mathcal{N}(w_i|0, \alpha_i^{-1}) \quad (19)$$

with  $\boldsymbol{\alpha}$  a vector of  $M$  unknown hyper-parameters, controlling the width of the marginal prior for the parameters  $w_i$ , *i.e.* the relevance of such parameters. When a hyper-parameter  $\alpha_i$  is high, the prior distribution of  $w_i$  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

$\sigma^2$ . For that purpose, we use a hierarchical prior approach, where we assign  $\alpha$  a Laplace prior probability distribution:

$$p(\alpha) = \prod_{i=1}^M \frac{\lambda_i}{2} \exp\left(-\frac{\lambda_i}{2\alpha_i}\right) \quad (20)$$

where  $\lambda = (\lambda_1, \dots, \lambda_M)^T$  are additional hyper-parameters that must be specified by the modeler. The choice of such priors, referred to as *demi-Bayesian LASSO* and introduced by [43], is motivated by the improved sparsity conferred to the algorithm with respect to the original formulation [39] where Gamma prior distributions were adopted. By increasing  $\lambda_i$ , we impose sharper prior distributions for the  $\frac{1}{\alpha_i}$ , *i.e.* higher values of the hyper-parameter  $\alpha_i$  become more likely, and consequently the corresponding model coefficient probability distribution  $p(w_i)$  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 is removed. For convenience, we fix  $\lambda_1 = \dots = \lambda_M = \lambda$ . 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  $\mathbf{w}$ ,  $\alpha$  and  $\sigma^2$ :

$$p(\mathbf{w}, \alpha, \sigma^2 | \mathbf{t}) = \frac{p(\mathbf{t} | \mathbf{w}, \alpha, \sigma^2) p(\mathbf{w}, \alpha, \sigma^2)}{p(\mathbf{t})} \quad (21)$$

where  $p(\mathbf{t} | \mathbf{w}, \alpha, \sigma^2)$  is the model likelihood (17) where we made explicit the dependency on hyperparameter vector  $\alpha$ ,  $p(\mathbf{w}, \alpha, \sigma^2)$  is the joint prior probability and  $p(\mathbf{t})$  is the model evidence. Unfortunately,  $p(\mathbf{t})$  is a multi-dimensional integral

$$p(\mathbf{t}) = \int p(\mathbf{t} | \mathbf{w}, \alpha, \sigma^2) p(\mathbf{w}, \alpha, \sigma^2) d\mathbf{w} d\alpha d\sigma^2 \quad (22)$$

and it 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:

$$p(\mathbf{w}, \alpha, \sigma^2) = p(\mathbf{w} | \mathbf{t}, \alpha, \sigma^2) p(\alpha, \sigma^2 | \mathbf{t}) \quad (23)$$

In the preceding equation,  $p(\mathbf{w} | \mathbf{t}, \alpha, \sigma^2) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  as a result of (17), (19) and

(20), where the posterior covariance and mean are respectively:

$$\begin{cases} \mathbf{\Sigma} = \left( \mathbf{A} + \frac{1}{\sigma^2} \mathbf{C}^T \mathbf{C} \right)^{-1} \\ \boldsymbol{\mu} = \frac{1}{\sigma^2} \mathbf{\Sigma} \mathbf{C}^T \mathbf{t} \end{cases}, \quad \mathbf{A} = \text{diag}(\alpha_1, \dots, \alpha_M). \quad (24)$$

On the other hand, regarding  $p(\boldsymbol{\alpha}, \sigma^2 | \mathbf{t})$ , we follow [39] and adopt a point estimate method by considering that the hyper-parameter posterior  $p(\boldsymbol{\alpha}, \sigma^2 | \mathbf{t})$  can be represented as a delta Dirac function at its most probable values  $\boldsymbol{\alpha}_{MP}$  and  $\sigma_{MP}^2$ . The SBL algorithm becomes the search for these most probable values, *i.e.* the maximization of  $p(\boldsymbol{\alpha}, \sigma^2 | \mathbf{t})$  with respect to  $\boldsymbol{\alpha}$  and  $\frac{1}{\sigma^2}$ .

### 3.2. Optimization of hyperparameters

Using Bayes' rule, the hyper-parameter posterior is of the form:

$$p(\boldsymbol{\alpha}, \sigma^2 | \mathbf{t}) \propto p(\mathbf{t} | \boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{\alpha}) p(\sigma^2) \quad (25)$$

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  $\sigma^2$ , the loss function to maximize becomes:

$$\begin{aligned} \mathcal{L}_{II} &= \log(p(\mathbf{t} | \boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{\alpha})) \\ &= \log \left( \int p(\mathbf{t} | \mathbf{w}, \sigma^2) p(\mathbf{w} | \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) d\mathbf{w} \right) \\ &= -\frac{1}{2} \left[ \log |\sigma^2 \mathbf{I} + \mathbf{C} \mathbf{A}^{-1} \mathbf{C}^T| + \mathbf{t}^T (\sigma^2 \mathbf{I} + \mathbf{C} \mathbf{A}^{-1} \mathbf{C}^T) \right] - \lambda \sum_{i=0}^M \frac{1}{\alpha_i} \end{aligned} \quad (26)$$

$\mathcal{L}_{II}$  is referred to as *the marginal likelihood* or *the evidence of hyper-parameters*, and its maximisation as the *type-II maximum likelihood method* [50] or *the evidence procedure* [47].

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

$$\begin{cases} \alpha_i^{new} = \frac{1 + \sqrt{1 + 8\lambda(\mu_i^2 + \Sigma_{ii})}}{2(\mu_i^2 + \Sigma_{ii})} \\ (\sigma^2)^{new} = \frac{\|\mathbf{t} - \mathbf{C}\boldsymbol{\mu}\|^2}{N - \sum_i \gamma_i} \end{cases} \quad (27)$$

For more details see [39, 43]. Finally, equation (27) still depends on the sparsity-promoting hyperparameter  $\lambda$ . The latter is selected using cross-validation, as discussed in the next section.

## 4. Results

### 4.1. Test cases and computational setup

Three incompressible flow cases representing typical separated flows over curved surfaces at similar Reynolds numbers are investigated. These flow cases are particularly challenging for LEVM. Specifically, the present baseline model is known to over-predict the size of the separation region. The calculations are performed using a modified version of the open-source finite-volume solver OpenFoam [51]. The governing equations are solved using the well-known SIMPLE algorithm. The convective terms are discretized using linear upwinding and viscous terms with 2nd order central difference scheme. The solution is advanced to the steady state using a Gauss-Seidel smoother. For each case, the grids are fine enough to ensure that the discretization error is much smaller compared to the error due to turbulence modelling.

The test cases at stake are described below.

- **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 [52] for  $Re = 10595$  (PH10595), where  $Re$  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  $120 \times 130$  cells. 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.
- **Converging-diverging channel (CD):** This configuration corresponds to a 2D channel of half-height  $H$  with an asymmetric bump of height  $h \simeq \frac{2}{3}H$  located on the bottom wall. The Reynolds number (based on the channel half-height and inlet conditions) is  $Re = 12600$ . A small separation occurs on the lee-side of the bump. For this test case, high-fidelity DNS data from [53] are available. The RANS simulations are based on a mesh of 140

x 100 cells. A velocity profile obtained from a companion channel-flow simulation is imposed at the inlet of the computational domain.

- **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, based on the inlet velocity and step height, is 13700. High-fidelity LES data from [54] are used for training. For the RANS simulations, the mesh consists of 140 x 150 cells. 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.

In the following, we consider three learning configurations or scenarios, summarized in Table 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  $\lambda$ . All models obtained from the same scenario  $s$  are noted as  $M^{(s)}$ .

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

Table 1: Learning scenarios and nomenclature for the resulting models.

## 4.2. Results

### 4.2.1. Model learning and cross-Validation

Data-driven SBL models are trained using data available for each scenario in Table 1 using different values of the regularization  $\lambda$  in  $[10^2, 10^3, 10^4, 10^5, 2.10^5]$ . The resulting models  $M = (\mathbf{M}_{b^\Delta}, \mathbf{M}_{b^R})$  (given in Appendix A) take the form:

$$\begin{cases} \mathbf{M}_{b^\Delta} = \sum_n \left( \sum_{l,m} (\mu_{(l,m)}^{\Delta(n)} \pm \sigma_{l,m}^{\Delta(n)}) I_1^l I_2^m \right) \mathbf{T}^{(n)} \pm \mathbf{1}\epsilon^\Delta \\ \mathbf{M}_{b^R} = \sum_n \left( \sum_{l,m} (\mu_{(l,m)}^{R(n)} \pm \sigma_{l,m}^{R(n)}) I_1^l I_2^m \right) \mathbf{T}^{(n)} \pm \mathbf{1}\epsilon^R \end{cases} \quad (28)$$

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

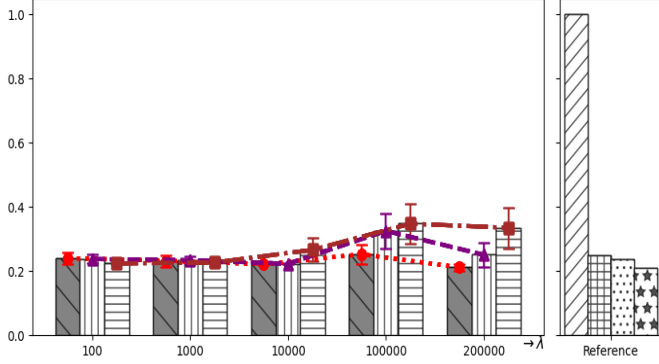
to  $I_1^l I_2^m$  in  $\mathbf{T}^{(n)}$  expansion in  $\mathbf{M}_{\mathbf{b}^\Delta}$  (resp.  $\mathbf{M}_{\mathbf{b}^R}$ ),  $\mathbf{1}$  the second order tensor with all the elements equal to one and  $\epsilon^\Delta$  (resp.  $\epsilon^R$ ) the standard deviation of the noise associated to the model  $\mathbf{M}_{\mathbf{b}^\Delta}$  (resp.  $\mathbf{M}_{\mathbf{b}^R}$ ).

In Appendix A we report  $M^{(1)}$ ,  $M^{(2)}$  and  $M^{(3)}$  derived for each value of  $\lambda$ . One can see that when  $\lambda$  is increased, models  $\mathbf{M}_{\mathbf{b}^\Delta}$  become sparser. On the contrary,  $\mathbf{M}_{\mathbf{b}^R}$  contains only one term regardless the value of  $\lambda$ , the regularization affecting only the magnitude of the corrective terms, which decreases as  $\lambda$  increases. Values of  $\lambda$  bigger than  $2 \cdot 10^5$  lead to cancel both  $\mathbf{M}_{\mathbf{b}^\Delta}$  and  $\mathbf{M}_{\mathbf{b}^R}$  thus leaving the  $k - \omega$  SST model unchanged.

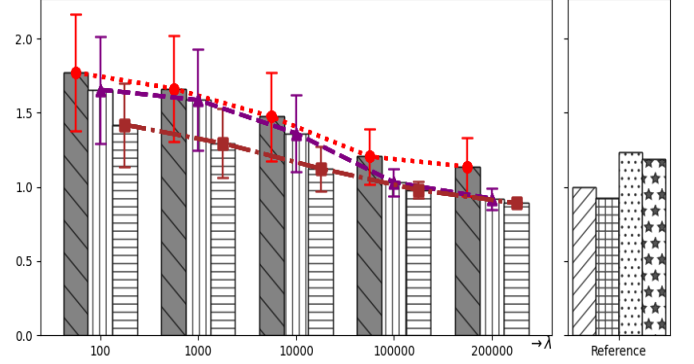
The models are cross-validated by propagating the MAP values of the coefficients through the CFD solvers, and computing MSE errors with respect to high-fidelity values available for various quantities. The errors obtained for various quantities of interest are reported in Figures 1, 2, 3 and 4 under the form of histograms, where column patterns refer to data-set used to train the model, while the abscissas indicate the value of the regularization parameter  $\lambda$ . In the figures, grey-shaded bars are used to indicate the extrapolation scenario of each model, i.e. where a model is used to make prediction on a case not included in its training set. White bars are used for 'post-dictions', i.e. for models evaluated on scenario they have been trained on. The bars of the derived corrections for each value of  $\lambda$  are equipped with an uncertainty interval, obtained by propagating the models' probability distributions through the CFD code with the same methodology explained previously and evaluating the maximum and minimum value of the MSE of every QoI output of the code when being fed by the drawn samples from these distributions. The results are systematically compared to those obtained using the baseline model and the three models of [46], obtained by deterministic SpaRTA algorithm. The latter are noted  $M_{Spa}^{(k)}$ , 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 1 shows the MSE for the streamwise velocity. It appears that the sparsity (and therefore the form of the model) affects the accuracy of horizontal velocity predictions:  $M^{(1)}$  at  $\lambda = 2 \cdot 10^5$  gives the most accurate results for both CD and PH cases, whereas  $M^{(3)}$  at  $\lambda = 10^4$  gives the most accurate prediction of horizontal velocity for CBFS case. These models exhibit a better or comparable accuracy than SpaRTA models, the confidence intervals of the former encompassing the predictions of the latter.

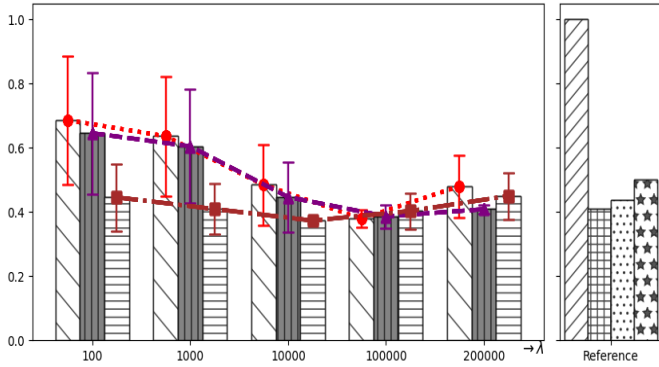




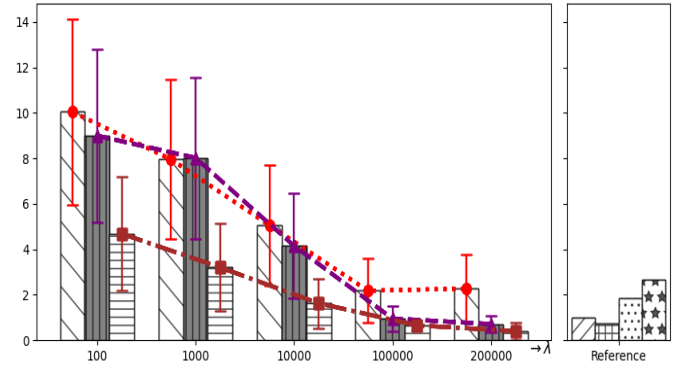
(a) Converging-diverging channel.



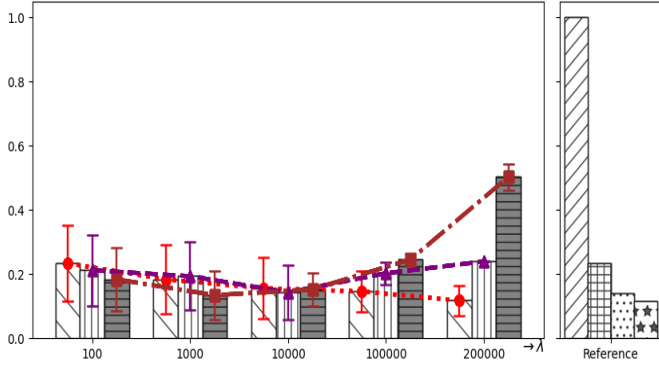
(a) Converging-diverging channel.



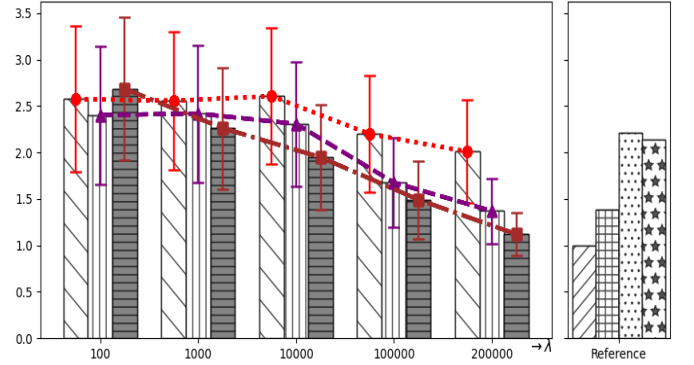
(b) Curved Backward-Facing Step.



(b) Curved Backward-Facing Step.



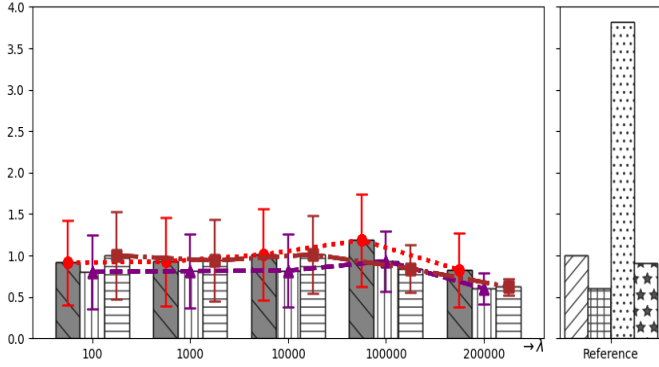
(c) Periodic Hills.



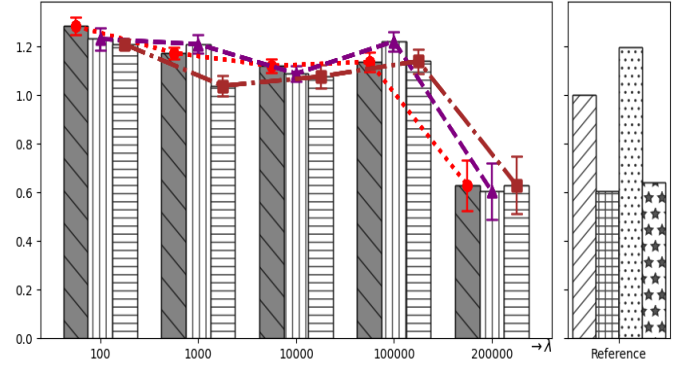
(c) Periodic Hills.

Figure 1: MSE of horizontal velocity normalized by the MSE of the baseline  $k - \omega$  SST prediction:  $\text{diagonal lines}$  for  $M^{(1)}$ ,  $\text{vertical lines}$  for  $M^{(2)}$ ,  $\text{horizontal lines}$  for  $M^{(3)}$ , compared with  $\text{diagonal lines}$  for the baseline  $k - \omega$  SST,  $\text{grid}$  for  $M_{Spa}^{(1)}$ ,  $\text{dots}$  for  $M_{Spa}^{(2)}$  and  $\text{stars}$  for  $M_{Spa}^{(3)}$ .

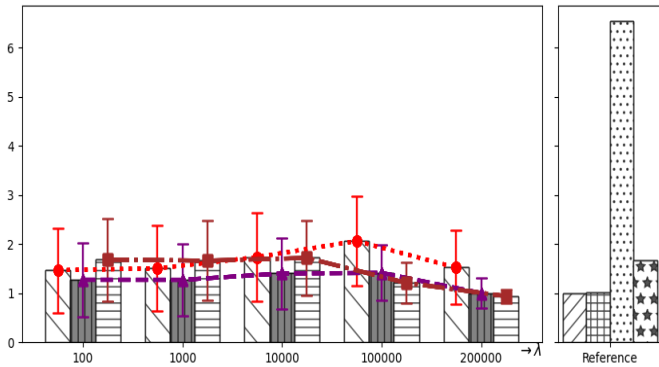
Figure 2: MSE of friction coefficient normalized by the MSE of the baseline  $k - \omega$  SST prediction:  $\text{diagonal lines}$  for  $M^{(1)}$ ,  $\text{vertical lines}$  for  $M^{(2)}$ ,  $\text{horizontal lines}$  for  $M^{(3)}$ , compared with  $\text{diagonal lines}$  for the baseline  $k - \omega$  SST,  $\text{grid}$  for  $M_{Spa}^{(1)}$ ,  $\text{dots}$  for  $M_{Spa}^{(2)}$  and  $\text{stars}$  for  $M_{Spa}^{(3)}$ .



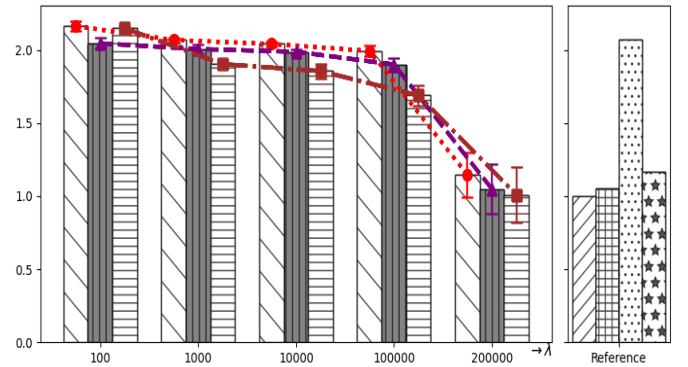
(a) Converging-diverging channel.



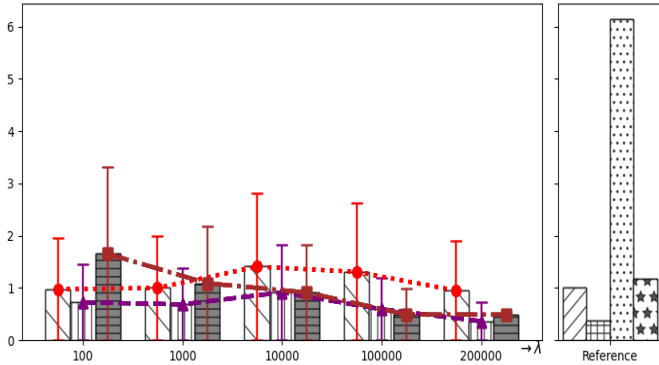
(a) Converging-diverging channel.



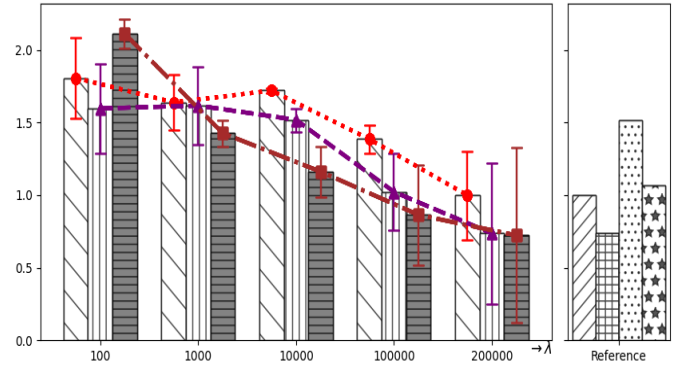
(b) Curved Backward-Facing Step.



(b) Curved Backward-Facing Step.



(c) Periodic Hills.



(c) Periodic Hills.

Figure 3: MSE of turbulent kinetic energy normalized by the MSE of the baseline  $k - \omega$  SST prediction:  $\text{diagonal lines}$  for  $M^{(1)}$ ,  $\text{horizontal lines}$  for  $M^{(2)}$ ,  $\text{vertical lines}$  for  $M^{(3)}$ , compared with  $\text{diagonal lines}$  for the baseline  $k - \omega$  SST,  $\text{horizontal lines}$  for  $M_{Spa}^{(1)}$ ,  $\text{vertical lines}$  for  $M_{Spa}^{(2)}$  and  $\text{stars}$  for  $M_{Spa}^{(3)}$ .

Figure 4: MSE of Reynolds shear stress normalized by the MSE of the baseline  $k - \omega$  SST prediction:  $\text{diagonal lines}$  for  $M^{(1)}$ ,  $\text{horizontal lines}$  for  $M^{(2)}$ ,  $\text{vertical lines}$  for  $M^{(3)}$ , compared with  $\text{diagonal lines}$  for the baseline  $k - \omega$  SST,  $\text{horizontal lines}$  for  $M_{Spa}^{(1)}$ ,  $\text{vertical lines}$  for  $M_{Spa}^{(2)}$  and  $\text{stars}$  for  $M_{Spa}^{(3)}$ .

Figures 2, 3 and 4 display MSE for the friction coefficient, the turbulent kinetic energy and the Reynolds shear stress, respectively. The figures show that models characterized by a null  $\mathbf{b}^A$  term, *i.e.* those obtained for  $\lambda = 2.10^5$ , are the most accurate ones. For these models only the  $\mathbf{b}^R$  correction of the production term is applied, in a sort corresponding 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.

From the previous analysis, it appears that the models resulting from cross-validation exhibit a variable accuracy depending on the sparsity and the observed QoI. One can see also that an optimal accuracy over a QoI may deteriorate others. Nevertheless, one should keep in mind that the proposed correction of anisotropy should not, in any case, give overall predictions that are worse than the baseline model. To evaluate the overall improvement conferred by each of the proposed corrections, we propose the following metric:

$$m(\lambda) = \sum_i \left( 1 - \frac{MSE_{BL}(QoI_i, \lambda)}{MSE_{BL}(QoI_i)} \right) \gamma_i = \sum m_i(\lambda) \gamma_i \quad (29)$$

where  $m_i(\lambda)$  is the contribution of a model in terms of accuracy with respect to the baseline model,  $\gamma_i$  a weighting coefficient of  $QoI_i$  and  $m(\lambda)$  the general contribution of a model in terms of accuracy that accounts for the decision variables ( $QoI_i$ ) with their respective weights. If  $m_i(\lambda)$  is positive (resp. negative), the model improves (deteriorates)  $QoI_i$  wrt the baseline model. The choice of the weighting coefficients determines the variables to be used in cross-validation as well as their weights. For example, if we want to cross-validate on a single variable  $QoI_i$ , we choose  $\gamma_{j \neq i} = 0$ ; if we want to cross-validate equally on two variables  $QoI_i$  and  $QoI_j$ , we choose  $\gamma_{k \neq i, j} = 0$  and  $\gamma_i = \gamma_j$ . It is noteworthy that the general improvement of a model should take into account whether the improvement metric has been evaluated on post-diction or extrapolation cases. To do so, we define the general improvement metric as follows:

$$m^G = \frac{m^{PD_1} + m^{PD_2} + 2m^{EP}}{4}, \quad (30)$$

where  $PD_1$  and  $PD_2$  refer to the post-diction cases and  $EP$  to the extrapolation case. The weight of the latter for each of the scenarios in Table 1 is doubled, thus

enabling the test and training cases to have an equal vote in the cross-validation decision. Finally, the best model  $\mathbf{M}^*$  is the one that maximizes  $m^G(\lambda)$ :

$$\mathbf{M}^* = \mathbf{M}(\arg_{\lambda} \max(m^G(\lambda))) = \mathbf{M}(\lambda^*) \quad (31)$$

In Table 2, we display the results of cross-validation over different choices of cross-validation (CV) variables for  $\mathbf{M}^{(1)}$ ,  $\mathbf{M}^{(2)}$  and  $\mathbf{M}^{(3)}$ . The most accurate and better generalizing models are the sparsest ones, i.e. the one characterized by a null  $\mathbf{b}^\Delta$ .  $\mathbf{M}^{(2)}$  and  $\mathbf{M}^{(3)}$  exhibit better accuracy over all the QoI at the same time ( $\simeq 30\%$ ) in comparison with  $\mathbf{M}^{(1)}$  (6%). The latter fails also at giving better results than the baseline model when cross-validated over  $k$ ,  $C_f$ ,  $(k, C_f)$ ,  $(\tau_{xy}, C_f)$  and  $(k, \tau_{xy}, C_f)$  (marked with ‘-’ in Table 2), which suggests that  $\mathbf{M}^{(1)}$  needs probably to be further regularized. On the contrary,  $\mathbf{M}^{(2)}$  and  $\mathbf{M}^{(3)}$  for  $\lambda^* = 2.10^5$  combine both a good accuracy and generalizability over all the QoI for all the cross-validation combinations, except when cross-validated only on the stream-wise velocity, where models with a non null  $\mathbf{b}^\Delta$  ( $\lambda^{*(2)} = 10^4$  and  $\lambda^{*(4)} = 10^3$ ) outperform the sparsest ones.

CV variables	$\lambda^{*(1)}$	$m^G(\lambda^{*(1)})$	$\lambda^{*(2)}$	$m^G(\lambda^{*(2)})$	$\lambda^{*(3)}$	$m^G(\lambda^{*(3)})$
$U, k, \tau_{xy}, C_f$	$2.10^5$	6.0 %	$2.10^5$	28.0 %	$2.10^5$	31.0 %
$U$	$2.10^5$	74.0 %	$10^4$	68.0 %	$10^4$	77.0 %
$k$	–	– %	$2.10^5$	26.0 %	$2.10^5$	36.0 %
$\tau_{xy}$	$2.10^5$	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 %
$U, \tau_{xy}$	$2.10^5$	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 %
$k, \tau_{xy}$	$2.10^5$	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 %
$\tau_{xy}, C_f$	–	– %	$2.10^5$	10.0 %	$2.10^5$	17.0 %
$U, k, \tau_{xy}$	$2.10^5$	29.0 %	$2.10^5$	36.0 %	$2.10^5$	38.0 %
$U, \tau_{xy}, C_f$	$2.10^5$	8.0 %	$2.10^5$	29.0 %	$2.10^5$	30.0 %
$U, k, C_f$	$2.10^5$	2.0 %	$2.10^5$	33.0 %	$2.10^5$	34.0 %
$k, \tau_{xy}, C_f$	–	– %	$2.10^5$	16.0 %	$2.10^5$	24.0 %

Table 2: Cross-validation statistics: best models and general improvement metrics

As a result of the preceding cross-comparison process, we select models representing the best compromise in terms of predicting the four precedent QoI, for

each of the three training scenarios. More precisely, models  $M^{(1)}(\lambda^{*(1)} = 2.10^5)$ ,  $M^{(2)}(\lambda^{*(2)} = 2.10^5)$  and  $M^{(3)}(\lambda^{*(3)} = 2.10^5)$  are retained (see Appendix A for their mathematical expressions). In the following, we drop the dependency on  $\lambda$  to simplify model notations.

For the selected models, the posterior probability distributions of the stochastic parameters are propagated through the CFD solver by means of a non-intrusive polynomial chaos method. More precisely, the stochastic collocation Method of [55] is adopted (see also [56, 57]). 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 contain only one stochastic coefficient. With this settings, three CFD simulations for each model are required to compute the statistical moments of the stochastic CFD predictions with satisfactory accuracy.

In Figures 5, 6, 7 and 8 respectively, we display selected profiles of the stream-wise velocity, turbulent kinetic energy, Reynolds shear stress, as well as the friction coefficient distribution along the bottom wall for the three flow configurations at stake. The expected solutions of  $\mathbf{M}^{(1)}$ ,  $\mathbf{M}^{(2)}$  and  $\mathbf{M}^{(3)}$  are reported along with three-standard deviation confidence intervals. For the sake of clarity, the latter are only reported for model  $\mathbf{M}^{(3)}$  that represents the best model in term of accuracy and generalizability (see Table 2), the other models providing similar results. The baseline  $k - \omega$  SST predictions and high-fidelity data are also reported for reference.

The SBL predictions for the velocity profiles (Figure 5) provide very similar solutions and overperform the baseline model in matching the high-fidelity data. The improvement is more evident in the recirculation region, where the present models accurately predict the back flow.

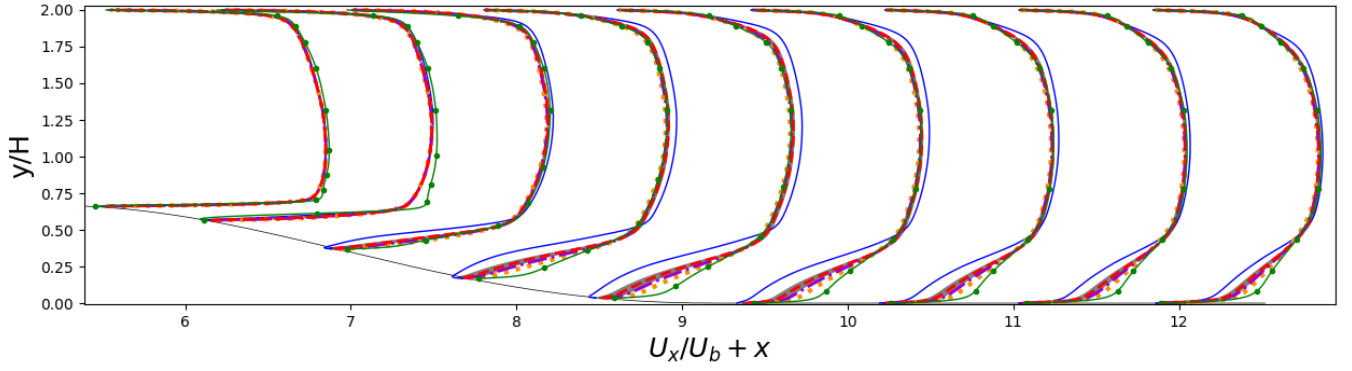
The three-standard deviation uncertainty intervals for this QoI predicted by  $\mathbf{M}^{(3)}$  are too narrow to capture everywhere the LES data, which is rather encompassed by those of  $\mathbf{M}^{(1)}$  (not shown for figure readability) whose MAP predictions are already closer to the LES data, as expected by Table 2.

The turbulent kinetic energy, presented in Figure 6, 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.

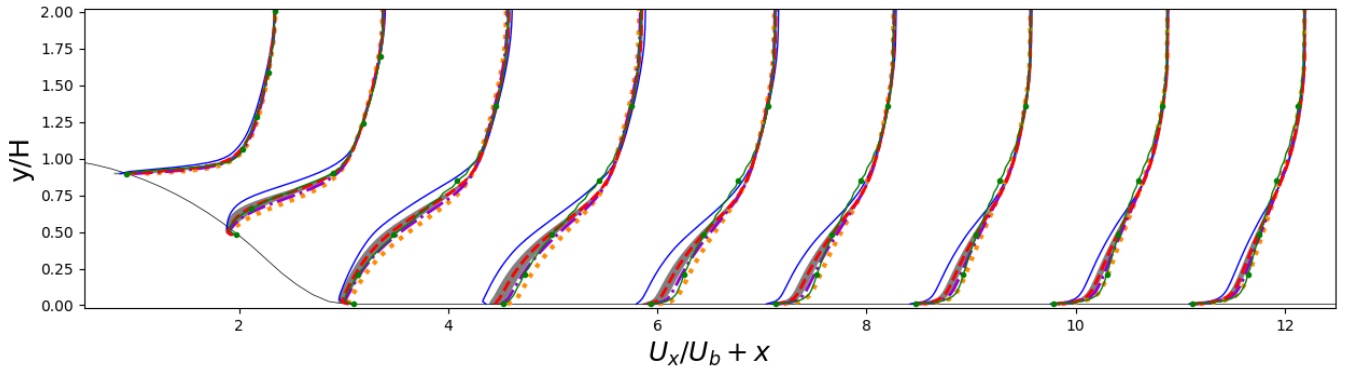
In Figure 7, the corrected Reynolds shear stresses succeed at reducing the observed discrepancy between the baseline prediction and high-fidelity data. They are however not able to encompass high-fidelity data everywhere in the flow by

their uncertainty intervals.

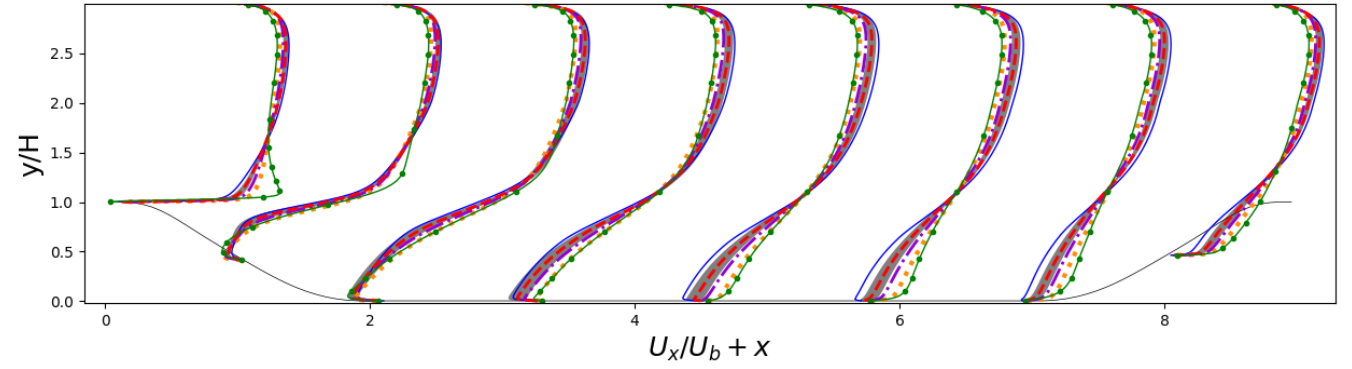
The predicted friction coefficient, reported in Figure 8, is in nice agreement with high-fidelity data, providing much more accurate estimates of the separation and reattachment points than the baseline. The confidence intervals encompass the separation and reattachment locations of the LES. For the CD case, none of the derived models is able to predict the exact location of the small separation bubble observed in LES results; nevertheless, the bubble position is shifted to the right and its size is significantly reduced. The predicted friction coefficient in the CBFS case is however slightly less accurate than the baseline in attached flow regions especially for  $\mathbf{M}^{(1)}$ , who is particularly having the highest mean value of the  $\mathbf{b}^R$  correction among the selected models. This is 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. Although the high-fidelity values are still contained in the confidence intervals of our stochastic models, further work is needed for generating more localized corrections. Recent attempts in this sense have been reported in [36].



(a) Converging-diverging channel.

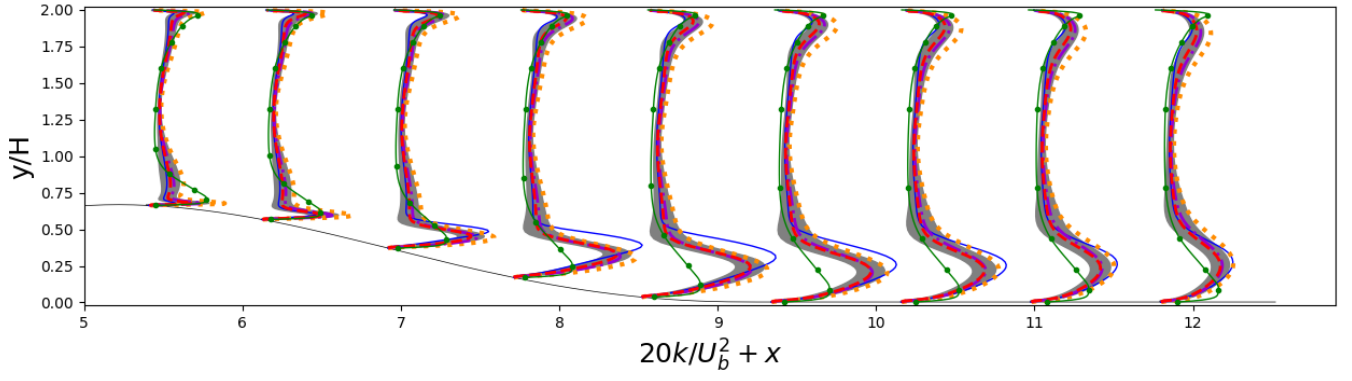


(b) Curved Backward-Facing Step.

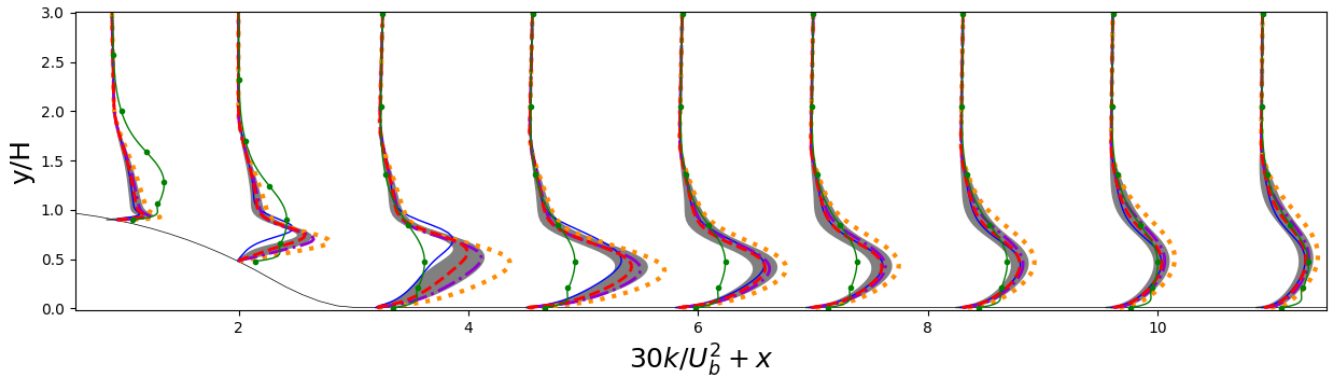


(c) Periodic Hills.

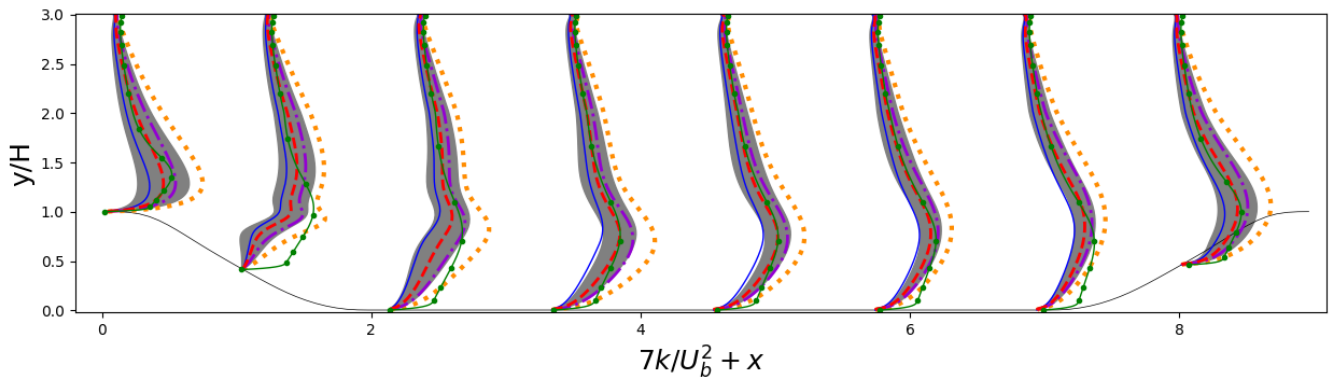
Figure 5: Streamwise velocity predictions: baseline  $k - \omega$  SST (—),  $M^{(1)}$  (⋯),  $M^{(2)}$  (⋯),  $M^{(3)}$  (---) with the uncertainty intervals (□), and LES (●).



(a) Converging-diverging channel.



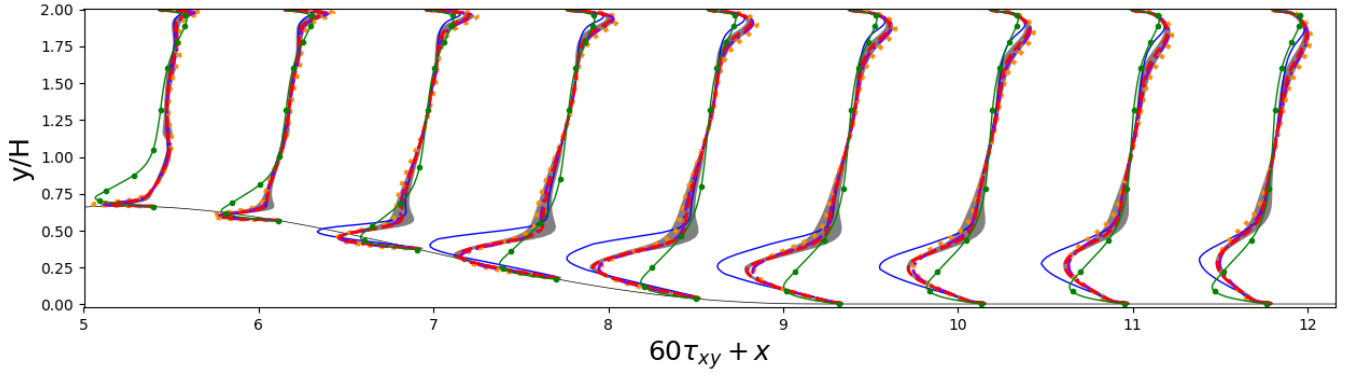
(b) Curved Backward-Facing Step.



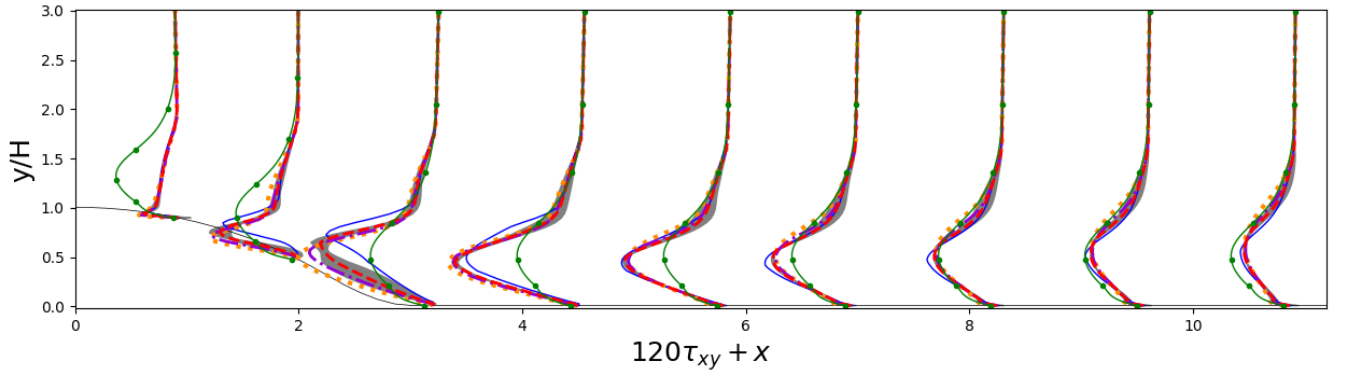
(c) Periodic Hills.

Figure 6: Turbulent kinetic energy predictions: baseline  $k-\omega$  SST (—),  $M^{(1)}$  (.....),  $M^{(2)}$  (-.-.-),  $M^{(3)}$  (- - -) with the uncertainty intervals ( ), and LES (—•—)

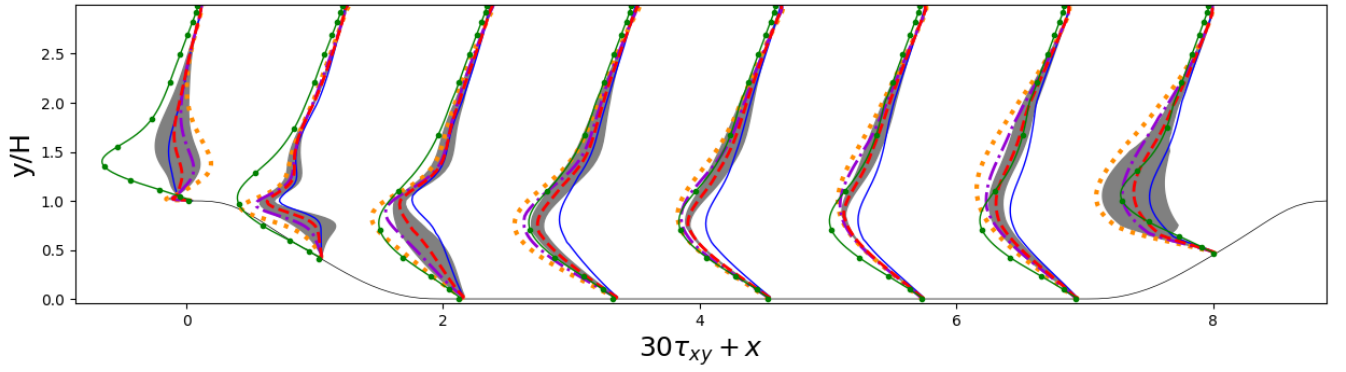




(a) Converging-diverging channel.

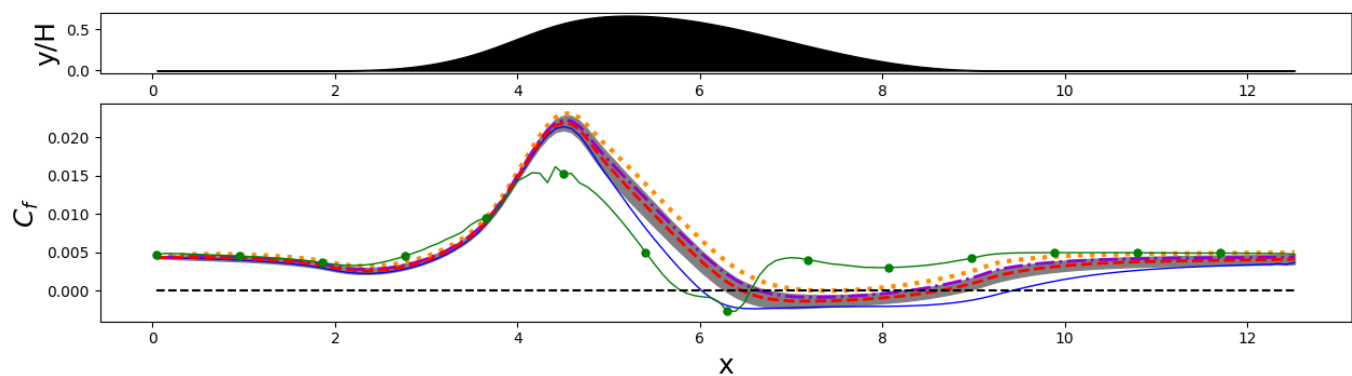


(b) Curved Backward-Facing Step.

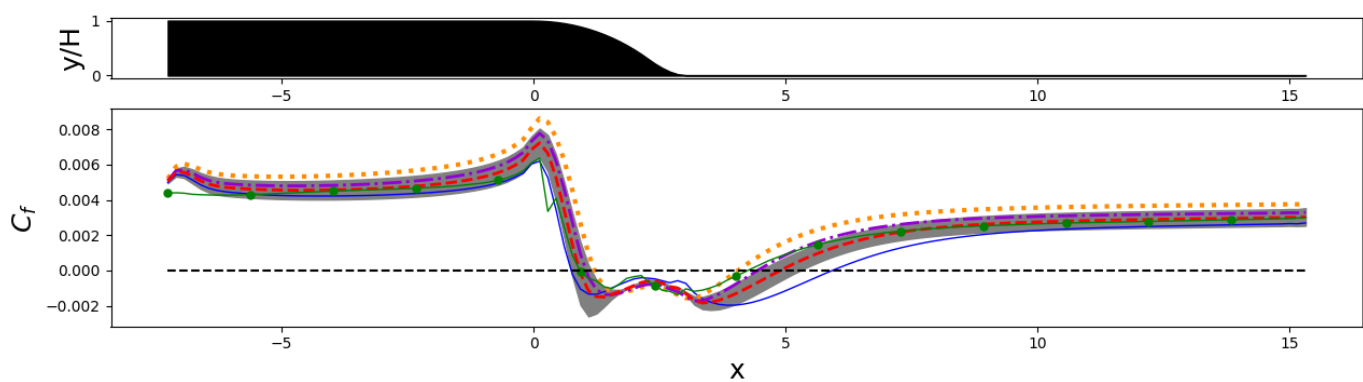


(c) Periodic Hills.

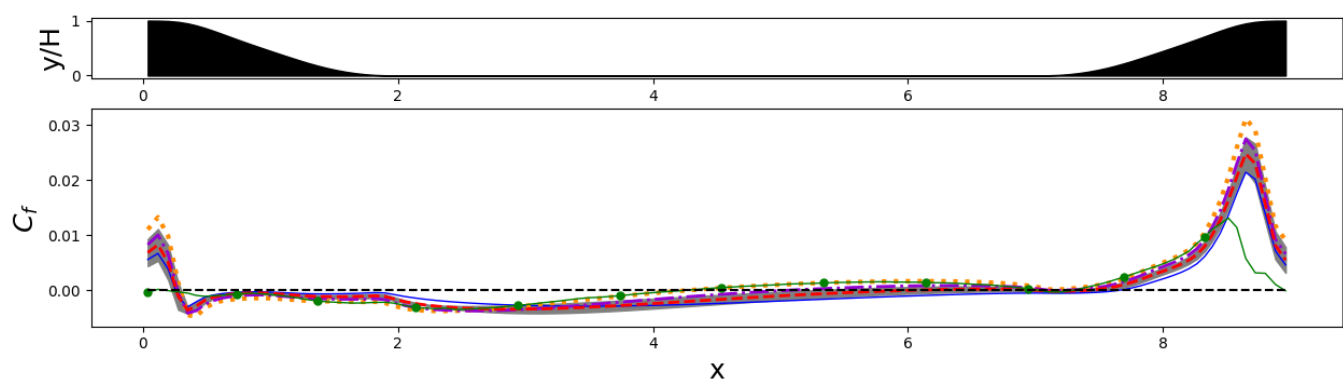
Figure 7: Reynolds shear stress predictions: baseline  $k - \omega$  SST (—),  $M^{(1)}$  (.....),  $M^{(2)}$  (---),  $M^{(3)}$  (- - -) with the uncertainty intervals (■), and LES (—●—).



(a) Converging-diverging channel.



(b) Curved Backward-Facing Step.



(c) Periodic Hills.

Figure 8: Friction coefficient predictions: baseline  $k - \omega$  SST (—),  $M^{(1)}$  (.....),  $M^{(2)}$  (-.-.-),  $M^{(3)}$  (- - -) with the uncertainty intervals (■), and LES (—●—).

#### 4.2.2. Sensitivity analysis

In the preceding section, an UQ method has been used for propagating the stochastic turbulence models through the CFD solver and the mean predictions and uncertainty intervals were obtained. Using the same UQ procedure, a Sobol sensitivity analysis of the models to the stochastic parameters can be conducted by following, e.g. the procedure described in [58].

In this section, results are reported for a more complex model, in particular the one that exhibits the highest accuracy (77%) when cross-validated on the streamwise velocity, i.e.  $\mathbf{M}^{(3)}(\lambda^{*(3)})$  for  $U$  (see Table 2). Our goal is to understand the local sensitivity of different QoI with respect to each of the correction terms in this case, namely that enables this model to outperform the sparsest ones in terms of predicting the streamwise velocity.

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 9a we display as an exemple 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  $a_{12}^\Delta$ . 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 being however the linear 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 9b, 9c and 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  $\mathbf{M}^{(3)}$  in predicting streamwise velocity with respect to the sparsest models is probably due to its ability to capture the anisotropy effects on velocity in the highly distorted regions.

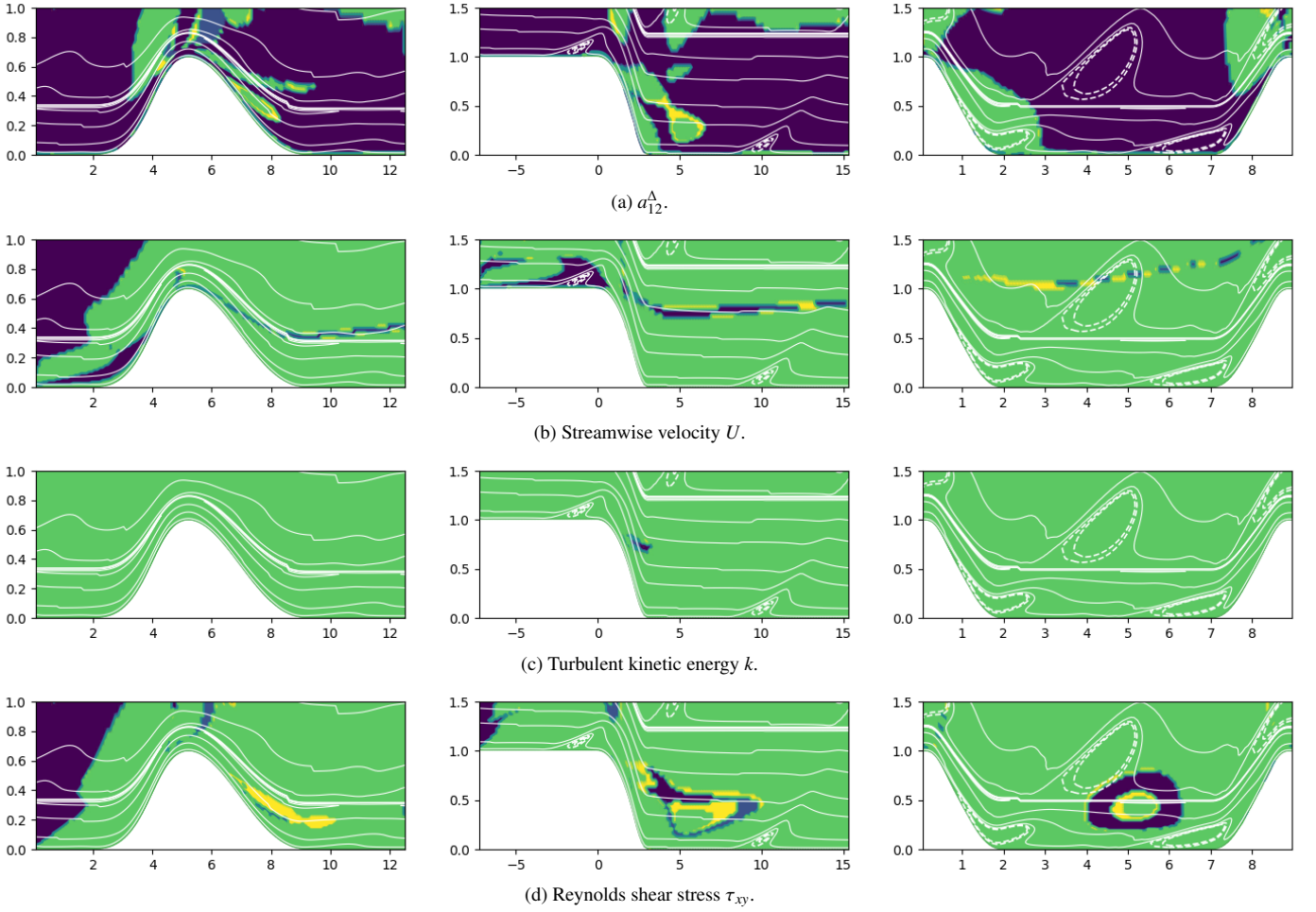


Figure 9: Maps of Sobol sensitivity indices dominance for various QoI: CD (left), CBFS (center) and PH (right) cases;  $s_1$  ■,  $s_2$  ■,  $s_3$  ■,  $s_4$  ■,  $\sum_{i,j} s_{ij}$  ■; streamlines —.

## 5. Conclusion

A novel Sparse Bayesian Learning (SBL) framework was introduced for generating stochastic EARSIM closures for turbulence models. The learned models are highly sparse, and generalize well to flows similar to those included in the training set. Thanks to their stochastic formulation, the model can be propagated through a CFD solver by means of an uncertainty quantification method, thus producing estimates of solution variability under turbulence modeling uncertainties. The estimated confidence intervals encompass in most of the cases the reference data. 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 likely not true for more complex, 3D flows, for which non linear terms are necessary to capture, e.g. secondary flows. An extension of the proposed method to 3D cases is planned in the near future. Finally, the present data-driven model correction procedure generates augmented models that are applied throughout the domain. However, this may lead to less accurate solutions in attached, equilibrium flow regions where the baseline model performs well. The development of localized corrections will make the object of future research work.

## References

- [1] D. C. Wilcox. *Turbulence modeling for CFD*. DCW Industries, 3 edition, 2006.
- [2] F. G. Schmitt. About Boussinesq’s turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity. *Comptes Rendus Mécanique*, 335:617–627, 2007.
- [3] Heng Xiao and Paola Cinnella. Quantification of model uncertainty in RANS simulations: A review. *Progress in Aerospace Sciences*, 108:1–31, 2019.
- [4] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019.

- [5] Karthik Duraisamy. Perspectives on machine learning-augmented reynolds-averaged and large eddy simulation models of turbulence. *Physical Review Fluids*, 6:050504, 2021.
- [6] Michael Emory, Johan Larsson, and Gianluca Iaccarino. Modeling of structural uncertainties in reynolds-averaged navier-stokes closures. *Physics of Fluids*, 25(11):110822, 2013.
- [7] 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*, 25(5):055105, 2013.
- [8] Roney L Thompson, Aashwin Ananda Mishra, Gianluca Iaccarino, Wouter Edeling, and Luiz Sampaio. Eigenvector perturbation methodology for uncertainty quantification of turbulence models. *Physical Review Fluids*, 4(4):044603, 2019.
- [9] P. D. A. Platteeuw, G. J. A. Loeven, and H Bijl. Uncertainty quantification applied to the  $k-\epsilon$  model of turbulence using the probabilistic collocation method. In *10th AIAA Non-Deterministic Approaches Conference*, 2008. Paper no.: 2008-2150.
- [10] 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*, 96(9):1137–1149, 2011.
- [11] W. N. Edeling, P. Cinnella, R. P. Dwight, and H. Bijl. Bayesian estimates of parameter variability in the  $k-\epsilon$  turbulence model. *Journal of Computational Physics*, 258:73–94, 2014.
- [12] L Margheri, M Meldi, MV Salvetti, and P Sagaut. Epistemic uncertainties in RANS model free coefficients. *Computers & Fluids*, 102:315–335, 2014.
- [13] Svetlana V Poroseva, M Yousuff Hussaini, and Stephen L Woodruff. Improving the predictive capability of turbulence models using evidence theory. *AIAA Journal*, 44(6):1220–1228, 2006.
- [14] WN Edeling, Paola Cinnella, and Richard P Dwight. Predictive rans simulations via bayesian model-scenario averaging. *Journal of Computational Physics*, 275:65–91, 2014.

- [15] Wouter N Edeling, Martin Schmelzer, Richard P Dwight, and Paola Cinnella. Bayesian predictions of reynolds-averaged navier–stokes uncertainties using maximum a posteriori estimates. *AIAA Journal*, 56(5):2018–2029, 2018.
- [16] 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*, 201:104473, 2020.
- [17] 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*, 2020.
- [18] B. Tracey, K. Duraisamy, and J.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.
- [19] Eric J Parish and Karthik Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774, 2016.
- [20] Heng Xiao, J-L Wu, J-X Wang, Rui Sun, and CJ Roy. Quantifying and reducing model-form uncertainties in reynolds-averaged navier–stokes simulations: A data-driven, physics-informed bayesian approach. *Journal of Computational Physics*, 324:115–136, 2016.
- [21] Jin-Long Wu, Jian-Xun Wang, and Heng Xiao. A bayesian calibration–prediction method for reducing model-form uncertainties with application in rans simulations. *Flow, Turbulence and Combustion*, 97(3):761–786, 2016.
- [22] Jin-Long Wu, Carlos Michelén-Ströfer, and Heng Xiao. Physics-informed covariance kernel for model-form uncertainty quantification with application to turbulent flows. *Computers & Fluids*, 193:104292, 2019.
- [23] Wouter Nico Edeling, Gianluca Iaccarino, and Paola Cinnella. Data-free and data-driven rans predictions with quantified uncertainty. *Flow, Turbulence and Combustion*, 100(3):593–616, 2018.

- [24] Julia Ling and Jeremy Templeton. Evaluation of machine learning algorithms for prediction of regions of high reynolds averaged navier stokes uncertainty. *Physics of Fluids*, 27(8):085103, 2015.
- [25] SBj Pope. A more general effective-viscosity hypothesis. *Journal of Fluid Mechanics*, 72(2):331–340, 1975.
- [26] Julia Ling, Reese Jones, and Jeremy Templeton. Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318:22–35, 2016.
- [27] Mikael L.A. Kaandorp and Richard P. Dwight. Data-driven modelling of the reynolds stress tensor using random forests with invariance. *Computers Fluids*, 202:104497, 2020.
- [28] Jin-Long Wu, Heng Xiao, and Eric Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids*, 3(7):074602, 2018.
- [29] Ashley Scillitoe, Pranay Seshadri, and Mark Girolami. Uncertainty quantification for data-driven turbulence modelling with mondrian forests. *Journal of Computational Physics*, 430:110116, 2021.
- [30] Jack Weatheritt and Richard Sandberg. A novel evolutionary algorithm applied to algebraic modifications of the rans stress–strain relationship. *Journal of Computational Physics*, 325:22–37, 2016.
- [31] Martin Schmelzer, Richard P Dwight, and Paola Cinnella. Discovery of algebraic reynolds-stress models using sparse symbolic regression. *Flow, Turbulence and Combustion*, 104(2):579–603, 2020.
- [32] S Beetham and J Capecelatro. Formulating turbulence closures using sparse regression with embedded form invariance. *Physical Review Fluids*, 5(8):084611, 2020.
- [33] Jasper P. Huijing, Richard P. Dwight, and Martin Schmelzer. Data-driven RANS closures for three-dimensional flows around bluff bodies. *Computers and Fluids*, 225:104997, 2021.
- [34] S. Beetham, R.O. Fox, and J. Capecelatro. Sparse identification of multi-phase turbulence closures for coupled fluid–particle flows. *Journal of Fluid Mechanics*, 914:A11, 2021.



- [35] Haochen Li, Yaomin Zhao, Jianchun Wang, and Richard D. Sandberg. Data-driven model development for large-eddy simulation of turbulence using gene-expression programming. *Physics of Fluids*, 33(12):125127, 2021.
- [36] Julia Steiner, Richard P. Dwight, and Axelle Viré. Data-driven RANS closures for wind turbine wakes under neutral conditions. *Computers and Fluids*, 233:105213, 2022.
- [37] Mingchuan Wang, Cai Chen, and Weijie 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*, 159:104742, 2022.
- [38] Xiaowei Xu, Fabian Waschowski, Andrew S.H. Ooi, and Richard 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*, 187:122557, 2022.
- [39] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.
- [40] R. Fuentes, R. Nayek, P. Gardner, N. Dervilis, T. Rogers, K. Worden, and E.J. Cross. Equation discovery for nonlinear dynamical systems: A bayesian viewpoint. *Mechanical Systems and Signal Processing*, 154:107528, 2021.
- [41] Yi-Chen Zhu, Paul Gardner, David J. Wagg, Robert J. Barthorpe, Elizabeth J. Cross, and Ramon Fuentes. Robust equation discovery considering model discrepancy: A sparse bayesian and gaussian process approach. *Mechanical Systems and Signal Processing*, 168:108717, 2022.
- [42] Rimple Sandhu, Mohammad Khalil, Chris Pettit, Dominique Poiré, and Abhijit Sarkar. Nonlinear sparse bayesian learning for physics-based models. *Journal of Computational Physics*, 426:109728, 2021.
- [43] Suhrid Balakrishnan and David 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*, pages 346–359, 2010.
- [44] Florian R Menter. Improved two-equation k-omega turbulence models for aerodynamic flows. *Nasa Sti/recon Technical Report N*, 93:22809, 1992.

- [45] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.
- [46] Martin Schmelzer, Richard P Dwight, and Paola 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*, pages 1789–1795. International Centre for Numerical Methods in Engineering, CIMNE, 2020.
- [47] David JC MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.
- [48] David JC MacKay. Bayesian methods for backpropagation networks. In *Models of neural networks III*, pages 211–254. Springer, 1996.
- [49] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.
- [50] James O Berger. Bayesian analysis. In *Statistical Decision Theory and Bayesian Analysis*, pages 118–307. Springer, 1985.
- [51] Henry G Weller, Gavin Tabor, Hrvoje Jasak, and Christer Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics*, 12(6):620–631, 1998.
- [52] Michael Breuer, Nikolaus Peller, Ch Rapp, and Michael Manhart. Flow over periodic hills—numerical and experimental study in a wide range of reynolds numbers. *Computers & Fluids*, 38(2):433–457, 2009.
- [53] Jean-Philippe Laval and Matthieu Marquillie. Direct numerical simulations of converging–diverging channel flow. In *Progress in wall turbulence: understanding and modeling*, pages 203–209. Springer, 2011.
- [54] Yacine Bentaleb, Sylvain Lardeau, and Michael A Leschziner. Large-eddy simulation of turbulent boundary layer separation from a rounded step. *Journal of Turbulence*, (13):N4, 2012.

- [55] GJA Loeven, JAS Witteveen, and Hester Bijl. Probabilistic collocation: an efficient non-intrusive approach for arbitrarily distributed parametric uncertainties. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, page 317, 2007.
- [56] Paola Cinnella and SJ Hercus. Robust optimization of dense gas flows under uncertain operating conditions. *Computers & Fluids*, 39(10):1893–1908, 2010.
- [57] Pietro Marco Congedo, Paola Cinnella, Samuel Hercus, Christophe Eric Corre, et al. Efficient robust optimization techniques for uncertain dense gas flows. In *CFD & Optimization-ECCOMAS Thematic Conference*, 2011.
- [58] Bruno Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety*, 93(7):964–979, 2008.

## Appendix A. SBL models obtained for various $\lambda$

- $\lambda = 10^2$ :

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

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

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

- $\lambda = 10^3$ :

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

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

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

- $\lambda = 10^4$ :

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

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

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

- $\lambda = 10^5$ :

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

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

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

- $\lambda = 2.10^5$ :

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

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

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

## Appendix B. Deterministic symbolic models obtained by SpaRTA in [31]

$$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(1)} = 0 \\ \mathbf{M}_{\mathbf{b}^R}^{(1)} = 0.39\mathbf{T}^{(1)} \end{cases} \quad (\text{B.1})$$

$$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(2)} = 0.1\mathbf{T}^{(1)} + 4.09\mathbf{T}^{(2)} \\ \mathbf{M}_{\mathbf{b}^R}^{(2)} = 1.39\mathbf{T}^{(1)} \end{cases} \quad (\text{B.2})$$

$$\begin{cases} \mathbf{M}_{\mathbf{b}^\Delta}^{(3)} = 0 \\ \mathbf{M}_{\mathbf{b}^R}^{(3)} = 0.93\mathbf{T}^{(1)} \end{cases} \quad (\text{B.3})$$