Aggregation of turbulence models for turbomachinery flows using Bayesian calibration and machine learning

Cécile Roques

SORBONNE UNIVERSITÉ

École doctorale Sciences Mécaniques, Acoustique, Électronique et Robotique

THÈSE DE DOCTORAT

SPÉCIALITÉ MÉCANIQUE

AGGREGATION OF TURBULENCE MODELS FOR TURBOMACHINERY FLOWS USING BAYESIAN CALIBRATION AND MACHINE LEARNING

Agrégation de modèles de turbulence pour les écoulements dans les turbomachines par calibration Bayésienne et apprentissage machine

Presentée par **Cécile ROQUES**

pour obtenir le grade de Docteur de Sorbonne Université

Sous la direction de Paola CINNELLA, Professeure, Sorbonne Université

Co-encadrée par Grégory DERGHAM, Ingénieur de recherche, Safran Tech

Xavier MERLE, Maître de Conférences, ENSAM

à l'Institut Jean Le Rond ∂'Alembert, Sorbonne Université, CNRS, UMR 7190

Soutenue le 23/10/2024 devant le jury composé de :

M. Jean-Camille CHASSAING	Professeur, Sorbonne Université	Président du jury
M. Francesco MONTOMOLI	Professeur, Imperial College	Rapporteur
M. Gael POETTE	CEA, Bordeaux-INP, IMB	Rapporteur
M. Didier LUCOR	Directeur de recherche CNRS	Examinateur
Mme Paola CINNELLA	Professeure, Sorbonne Université	Directrice de thèse
M. Grégory DERGHAM	Docteur Ingénieur, Safran Tech	Co-encadrant
M. Xavier MERLE	Maître de Conférences, ENSAM	Co-encadrant

Résumé

Malgré la croissance exponentielle de la puissance de calcul, les simulations haute fidélité restent trop coûteuses pour les configurations industrielles complexes, telles que les turbomachines. Par conséquent, l'utilisation de modèles moyennés (RANS) demeure la méthode prédominante pour la conception et l'optimisation industrielles. Cependant, en l'absence de consensus sur quel modèle RANS est le plus performant, chacun étant seulement fiable dans les configurations pour lesquelles il a été calibré, le jugement d'experts joue un rôle crucial dans la sélection des modèles, soulignant l'importance de quantifier et de minimiser les incertitudes associées à la modélisation RANS. Pour réduire ces incertitudes, les études précédentes se sont concentrées sur la calibration et/ou la combinaison d'un ensemble de modèles RANS. Bien que ces approches soient prometteuses, elles supposent généralement des pondérations constantes, et ne procèdent souvent pas à des mises à jour des modèles pour les améliorer. Cette thèse explore trois approches basées sur les données pour calibrer et agréger des prédictions de différents modèles. La première méthode consiste à agréger les prédictions des modèles calibrés dans un cadre bayésien, avec des pondérations dépendantes de l'espace exploitant les perfomances locales des modèles. Pour améliorer la généralisabilité de cette approche, la deuxième méthode repose sur la calibration et l'agrégation des modèles au sein de dynamiques physiques cohérentes, identifiées à l'aide d'un algorithme de clustering. Cette méthode démontre une meilleure généralisabilité et une mise en œuvre plus simple que la première. Les deux méthodes sont développées et testées sur une cascade de compresseurs linéaire 2D. S'appuyant sur la calibration au sein d'une dynamique physique, la troisième approche utilise des données expérimentales de quatre configurations simples, représentantes d'une dynamique physique, pour calibrer les modèles. Les prédictions sont agrégées en utilisant des pondérations dépendantes de l'espace, optimisant ainsi l'utilisation de chaque modèle dans sa dynamique de calibration. Soulignons que cette méthode est appliquée pour la première fois à une configuration 3D de turbomachines—la cascade linéaire NACA65—et démontre des performances supérieures à celles de tous les modèles de l'ensemble initial.

Abstract

Despite the rapid growth in computational power, Direct Numerical Simulations (DNS) and Large Eddy Simulations (LES) remain prohibitively expensive for simulating complex industrial flows, such as those in turbomachinery. Consequently, Reynolds-Averaged Navier-Stokes (RANS) modelling continues to be the primary tool for industrial design and optimisation. However, selecting the appropriate RANS model poses a significant challenge due to the absence of a consensus on which model performs best, as each is only reliable within the flow configurations for which it was calibrated. As a result, expert judgement plays a crucial role in model selection, highlighting the importance of accurately quantifying and minimising the uncertainties associated with RANS modelling. To address these uncertainties, previous research has focused on updating and/or combining a predefined set of RANS models. While these approaches show promise, they typically assume constant combination weights across the covariate space, ignoring the fact that model accuracy can vary depending on the region of interest, and often do not involve updates to improve model predictions. This thesis explores three data-driven approaches for updating and combining the solutions of multiple competing turbulence models. The first method aggregates predictions from updated models using Bayesian calibration, with space-dependent weights that leverage local flow information. To enhance the generalisability of this approach, the second method updates and aggregates models within coherent physical dynamics, identified through an unsupervised clustering algorithm and novel local flow features. This method demonstrates improved generalisability and easier implementation. Building on the physical-dynamic-based calibration, the third approach employs experimental data from four simple configurations, each representing a different physical dynamic, to calibrate the models. The predictions from these updated models are then aggregated using space-dependent weights, optimising the use of each model within its calibrated physical dynamic. Notably, this method is applied for the first time to a 3D turbomachinery configuration—the NACA65 linear compressor cascade—demonstrating superior predictive performance than any of the baseline models.

Remerciements

Bien que le doctorat soit souvent perçu comme un voyage solitaire, il est en réalité tissé de liens et le fruit du soutien apporté par tant de personnes précieuses. Sans elles, aucun de mes efforts n'aurait pu aboutir. À travers ces lignes, je tiens à exprimer ma plus profonde gratitude envers toutes celles et ceux qui m'ont accompagnée dans ce parcours. Je remercie tout particulièrement les membres de mon jury, les membres de mon comité de suivi et mes encadrants Grégory Dergham, Paola Cinnella et Xavier Merle, dont les conseils, l'accompagnement et la bienveillance ont été indispensables à la réalisation de ce travail. La confiance et le soutien qu'ils m'ont apportés au cours de ces trois années ont été précieux dans ma motivation, mon apprentissage et mon épanouissement dans ce projet de recherche.

Parce que ce doctorat marque la fin de mon parcours étudiant, il me tient à cœur de remercier mes enseignants, qui ont nourri ma curiosité et m'ont donné les clés pour avancer. Merci à Mme Ordines qui, au lycée de Provence, m'a fait découvrir la beauté des maths et leur capacité à révéler les lois du monde. Merci également à M. Grécias et M. Martin, dont les conseils, bien au-delà des concours et de la CPGE au lycée Thiers, m'accompagneront toute ma vie. À Arnaud Le Diffon et Laure-Lise Chapellet, à l'ENS, je leur suis infiniment reconnaissante pour m'avoir redonné goût à la physique et à la chimie en préparant l'agrégation avec eux. Un immense merci à François Farago pour m'avoir ouvert les portes du CNES et éveillé un rêve d'enfant. Enfin, ma gratitude éternelle à Yves Plancherel, qui incarne l'excellence en tant qu'encadrant : curieux, bienveillant et capable d'insuffler une collaboration joyeuse où chacun veut donner le meilleur de soi-même.

Pendant ces neuf années d'études, j'ai eu la chance de croiser des personnes exceptionnelles qui ont marqué mon quotidien. Je pense tout particulièrement à mes collègues chez Safran : Dimitrios, Mickaël, Marc, Alberto, Armando, Mohamed, Thibaut, Thomas, Yoan, et surtout Enza, qui a été un pilier et une source constante de joie. Travailler avec eux a été un bonheur, et j'espère avoir la chance de collaborer à nouveau avec des collègues aussi inspirants.

Mon expérience entre Safran et le laboratoire d'Alembert m'a également permis de rencontrer des personnes qui ont su transformer le labo en un lieu de vie et d'amitié. Merci à Yutao, Clément, Serena, Quentin, Mahshid, Marcellin, Samy, Patricia, Tomas, Ludo, Hector, Belka, Camilla, Lucas, Amine, Simona, Mina, Nicolas, Gabriel, Lila, Krishan, et à tous les autres avec qui j'ai partagé des moments inoubliables. Une mention spéciale à Ariadni, Mikail, Alejandro et Camille, mes complices, pour leur soutien indéfectible et leur amitié.

À mes amis de Cachan, merci pour ces innombrables fous rires, ces souvenirs gravés, et tous ces dancefloors enflammés. Merci Maza, Oliv, Hélo, Claire, Noémie, Francis, Keut, Pniak, Lala, Guiblanc, Jules, Louis, Lucien, Maël, Marion, Leela, JX, Marie, Elric, Alexandre et tant d'autres. Votre amitié est une source inestimable de joie, et c'est un immense privilège de vous avoir dans ma vie.

À mes amis de toujours, vous êtes un repère essentiel à chaque étape de ma vie. Je ne peux imaginer mes souvenirs sans l'un d'entre vous à mes côtés. Merci Albrand, Tess, Robi, Ndup, Piga, Hanoun, Mathilde, Margot, Redouane, Yoann, Romane, pour votre présence qui illumine ma vie depuis tant d'années.

Enfin, à mes parents, à ma soeur et mon frère et à ma grand-mère. Les mots me manquent. Vous avez toujours cru en moi, souvent bien plus que moi-même. Votre amour, votre soutien et vos encouragements sont ma plus grande source de force et de détermination. Merci pour tout. Je vous aime très fort.

Je terminerai ces lignes en mentionnant ma grand-mère, qui s'inquiète toujours de savoir si j'ai « des bons copains ». J'ai la chance de pouvoir lui répondre : « Oui, Mamie et ils sont formidables! »

Table of contents

1

Introduction

In the aeronautical sector, there is no room for compromise when it comes to flight safety. Aviation authorities across the globe impose strict regulations and conduct thorough certification processes for every aircraft component, whether physical or software-based. Beyond the critical concern of safety, the failure or under-performance of any component can have severe repercussions for an Original Equipment Manufacturer (OEM), including significant delays and financial losses amounting to millions of euros due to contract penalties.

OEMs implement sophisticated risk management strategies to ensure performance and minimise costs from the design phase through to testing. For instance, the development of aircraft engines involves iterative, time-intensive, and costly methods aimed at reducing risks during the prototyping and real-life testing phases. These stages are crucial for ensuring the reliability of all engine components. A typical aeronautical jet engine (the LEAP-1B is displayed in figure 1.1), is typically composed of a fan, that pulls fresh air into the engine, and a compressor elevates the pressure of part of the incoming air. The latter then passes through a combustion chamber, where the fluid is mixed with fuel and ignited. The high-pressure, high-temperature

gases are then expanded through a turbine, which supplies energy to drive the fan and compressor. Finally, the gases are accelerated through a propulsive nozzle, producing the thrust needed to move the aircraft as they exit at high speed.

While the design of such an engine requires extensive and interdisciplinary expertise in various

(a) Global view (b) Detailed view

Figure 1.1 – Global and detailed views of the LEAP-1B engine. (Source : Safran website)

fields of engineering, including structural design, fluid dynamics, combustion, aeroelasticity, etc., The focus of this thesis is set exclusively on fluid dynamics problems. More specifically, we target the improvement of models used to design the rotating machinery equipment, *i.e.* fans, compressors and turbines. Furthermore, given the fundamental nature of the present study, we will restrict the present applications to a simplified configuration, representative of a cascade of compressor blades.

Improved fluid-dynamic design is key to the development of the next generation, low-carbon-emission aeronautical engines, and thus fulfil the European Commission's ambitious "Green Deal", the Clean Aviation Strategic Research and Innovation Agenda (SRIA), as well as the Advisory Council for Aviation Research and Innovation (ACARE) vision report "Fly the Green Deal" [1], set the following twofold objective for the European aviation industry:

1. By 2035, realise ultra-efficient aircraft propulsion systems that enable >30% Green House gas emission reductions at the aircraft level vs. the 2020 state-of-the-art and >85% Green House gas emission reductions when combined with sustainable aviation fuels (SAF).
2. By 2050, realise climate-neutral aviation.

However, these future engine architectures and enabling technologies require opening up

the design space and pose needs and challenges that go far beyond the capabilities of today's industrial design processes [48].

Today's aerodynamic engine (and aircraft) designs are based on computational fluid dynamics (CFD) simulations that solve the 3D Reynolds-averaged Navier-Stokes (RANS) equations and exhibit significant uncertainties when a typical and well-defined design space is exceeded, e.g., in the presence of strong secondary flow effects. In addition, manifold uncertainties related to manufacturing tolerances, random variations of the operating conditions, imperfectly known fluid properties, etc. may also be present, putting the design at risk.

Following the American Institute of Aeronautics and Astronautics (AIAA) [4], **uncertainty** is a *potential* deficiency in any phase or activity of the modelling process due to a lack of knowledge. This contrasts with **errors**, defined as a *recognisable* deficiency in any phase or activity of modelling and simulation not caused by a lack of knowledge.
Zio and Apostolakis distinguish between two types of uncertainties [114]:

— **Aleatoric Uncertainty:** arises from the intrinsic randomness of a system and, as such, is inherently irreducible.

— **Epistemic Uncertainty:** results from a lack of knowledge, which can be mitigated as new knowledge is provided, e.g. by moving to better models or providing more data.

Oberkampf and Blottner [76] identified four primary sources of uncertainty and error in the numerical simulation of physical problems governed by Partial Differential Equations:

1. **Physical Modelling:** relates to assumptions about the problem, boundary conditions, and any auxiliary models, such as thermodynamic or turbulence modelling.
2. **Discretisation and Solution Errors:** this category comprises errors in the numerical representation of geometry, grid convergence issues, convergence errors arising from large matrix inversions, and truncation errors introduced by the numerical schemes used to approximate the spatial and time derivatives.
3. **Computer Round-off Errors:** These errors are due to the finite representation of numbers on a computer, and to their manipulation (arithmetic operations).
4. **Programming and User Errors:** These are generally negligible if high-precision arithmetic is used and the code and setup are meticulously prepared and debugged.

In this work, we will focus on the first category of uncertainties, *i.e.* those associated with physical modelling, deliberately excluding discretisation and solution errors from our scope. While

setting a suitable modelling level, e.g., including or not the viscous effects, is generally a deliberate choice (due to prior knowledge of the problem and the available computational resources) made by the users, some modelling choices are more controversial, and the quality of the solution can be highly problem-dependent. This is the case of models used to account for the effects of turbulence, omnipresent in flow problems of engineering interest.

The key difficulty of turbulence rests in the non-linear convective term in the Navier-Stokes equations describing fluid dynamics. Without an analytic solution to the Navier-Stokes equations, the best candidate to make a prediction is a numerical simulation of turbulence. Different strategies can be considered, from Direct Numerical Simulation (DNS), the highest fidelity strategy in which all spatial and temporal scales are resolved, to Reynolds-Averaged Navier-Stokes (RANS), the lowest fidelity strategy in which all the scales are modelled. A compromise between DNS and RANS simulations is Large Eddy Simulations (LES) in which only the larger and more energetic scales are resolved. Choosing the appropriate level of fidelity involves a compromise between computational cost and predictive accuracy.

Despite the recent advances in computational capabilities, DNS and LES remain prohibitively expensive for routine use in industry. A 3D DNS simulation requires at least $\mathbf{Re}^{9/4}$ grid cells to resolve every spatial and temporal scale. LES simulations are less expensive but remain too costly for wall-bounded industrial applications characterised by a high Reynolds number $\mathbf{Re}$ [83]. The only affordable strategy for those kinds of flows is RANS simulations where every spatial and temporal turbulent scale is modelled. Despite being notably flawed in terms of accuracy, its computational affordability makes RANS modelling the workhorse for all industrial applications of CFD, including turbomachinery design. Unfortunately, RANS models come with considerable uncertainties [110], which may hinder the design of turbomachinery components that radically differ from known designs, for which industrial know-how can be used to compensate for modelling inaccuracies. In the aim of ensuring a robust design of novel turbomachinery components, it is essential to quantify the uncertainties that originate from the use of RANS models instead of higher fidelity methods. While a wealth of uncertainty quantification methods have been developed to deal with aleatoric uncertainties, *i.e.* for propagating uncertainties in the CFD model input parameters to the output solution [75], dealing with physical modelling uncertainties is a more challenging task. A full review is out of the scope of the present thesis. In the following, we focus on attempts to quantify uncertainties associated with turbulence models, and more precisely with RANS models.

1.1 Quantification of RANS modelling uncertainties

Turbulence model uncertainties are epistemic, *i.e.* due to incomplete knowledge of the physics of turbulence for general complex flows and to computational limitations in modelling tools. Building on the frameworks established by Kennedy and O'Hagan [55], Arendt *et al.* [2], and Duraisamy *et al.* [31], the use of a RANS model can introduce four types of uncertainties:

— **Structural uncertainties** related to the validity of the averaging process itself.

— **Representation uncertainties** in representing the unclosed Reynolds stress tensor as a function of the mean field, for example, the choice of a linear eddy viscosity representation over another possible representation.

— **Model-form uncertainties** in the mathematical form of the constitutive laws used to relate the Reynolds stresses to the mean field, as well as in the auxiliary transport equations for turbulence properties (e.g., turbulent kinetic energy or dissipation).

— **Parametric uncertainties** associated with the closure parameters of a given model form, which affect the representation of the Reynolds stress tensor as a function of the mean field.

In this work, we aim to provide a Bayesian probabilistic framework for RANS-based predictions of turbulent flows that accounts for both parametric uncertainties and model-form uncertainties. Consequently, the following discussion primarily focuses on these types of uncertainties. Furthermore, we preferentially look at RANS modelling uncertainties under the prism of Bayesian statistics. Elements of Bayesian statistical inference instrumental to the present study are presented in Chapter 3. For a more comprehensive review of model uncertainties in RANS models, the reader is referred to [110].

1.1.1 Parametric Uncertainties

As previously mentioned, parametric uncertainties arise from a lack of knowledge regarding model parameters. The Bayesian framework provides a natural approach for quantifying parametric uncertainties, treating parameters not as fixed values but as random variables endowed with probability distributions. Consequently, the model output is also probabilistic, enabling the estimation of prediction confidence intervals.

Bayesian inference involves updating prior knowledge about RANS model parameters based on experimental or high-fidelity data. These inferred parameter distributions are then propagated

through the computer code to obtain output distributions. Bayesian inference is based on Bayes' theorem of inverse probability :

$$p(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)}$$

Here, the posterior distribution $p(\boldsymbol{\theta}|D)$ of the parameter is proportional to its prior distribution $p(\boldsymbol{\theta})$ and to the likelihood $p(D|\boldsymbol{\theta})$. The prior summarises existing knowledge about the parameter before data observation, while the likelihood indicates the probability of observing the data given the model parameterised by $\boldsymbol{\theta}$. The denominator $p(D)$ is called the evidence, and it denotes the total probability of observing the data. While for simple statistical models for which the likelihood and prior are analytical functions, and the posterior can often be computed analytically (see, e.g.[46]for more information), in CFD applications the likelihood depends on the solver output, *i.e.* it is not analytically tractable. As a consequence, approximate data-driven methods must be used. Two data-driven methods are prominent: Markov Chain Monte Carlo (MCMC) sampling and Maximum a Posteriori (MAP) estimation. MCMC sampling involves sequential sampling where each subsequent sample depends solely on the current one. This algorithm constructs a Markov chain whose stationary distribution matches the target distribution. Achieving convergence in MCMC demands a large number of samples, which can be impractical for computationally intensive codes, such as those used in Computational Fluid Dynamics (CFD) applications. Moreover, Bayesian methods allow for continuous updates of inferred coefficients as new calibration data becomes available [47]. The resulting posterior probability distributions can further be utilised to analyse the model's sensitivity–or lack thereof–to closure parameters. Alternatively, when only low-order moments like mean and variance of the quantity of interest are required instead of the entire distribution, methods using MAP estimation can be employed to obtain the mode of the posterior distribution. This significantly reduces computational costs [110].

Following the seminal work by Kennedy and O'Hagan [55] on computer model calibration and aided by advances in computational power, Cheung *et al.* [17] pioneered Bayesian calibration in Computational Fluid Dynamics (CFD). Their study focused on calibrating the Spalart-Allmaras RANS model using Direct Numerical Simulation (DNS) data from plane channel flows. Subsequent research by Oliver and Moser employed a similar methodology to calibrate three additional models: Baldwin-Lomax, Chien $k-\epsilon$, and v^2-f RANS models. Similarly, Edeling *et al.* [35] extended this calibration approach to five RANS models ($k-\epsilon$, $k-\omega$, Spalart-Allmaras, Baldwin-Lomax, and stress-ω models) using data from 14 flat plate flows under varying external pressure gradients.

More recently, de Zordo-Banliat *et al.* applied the Bayesian framework to calibrate the Wilcox $k-\omega$, Launder-Sharma $k-\epsilon$, and Spalart-Allmaras RANS models for flows around a NACA65 V103 cascade.

Notably, this approach is non-intrusive in the sense that its implementation does not involve modifications of the RANS solver as the stochastic parameters can be just fed as inputs to the CFD solver and propagated by using a suitable uncertainty quantification method. Non-intrusiveness may be an essential requisite for some industrial applications, e.g. those involving the use of black-box CFD solvers the sources of which are not easily accessible.

1.1.2 Model-form uncertainties

Model uncertainty arises from the mathematical form used to approximate the physical phenomenon. Whilst constructing a RANS model, one type of uncertainty originates from the choice of the form of the Reynolds tensor, choosing to use a linear eddy viscosity or an algebraic stress model for example. Once a form is chosen, a functional mathematical model needs to be formulated, leading to the second type of model-form uncertainty. Following the review of Xiao and Cinnella [110], two approaches for quantifying model-form uncertainties are possible, depending on where the uncertainties are introduced.

Non-parametric Approaches

Non-parametric approaches introduce uncertainties in the formulation of RANS models. Here again, various methods for uncertainty quantification (UQ) are obtained depending on where in the model formulation uncertainty is introduced.

Duraisamy and collaborators [91, 32, 92, 79, 93, 30] have introduced uncertainty into the transport equations of RANS models. They directly target the functional form of uncertainty by estimating discrepancies in the source terms of transport equations. For instance, in [91], they introduce a field $\beta(\mathbf{x})$ as a multiplier of the production term in the surrogate viscosity $\tilde{\nu}_t$ transport equation of the Spalart-Allmaras model:

$$\frac{D\tilde{\nu}_t}{Dt} = \beta(\mathbf{x})P\left(\tilde{\nu}_t, \mathbf{U}\right) - D\left(\tilde{\nu}_t, \mathbf{U}\right) + T\left(\tilde{\nu}_t, \mathbf{U}\right)$$

Here, $\mathbf{U}$ contains Reynolds-averaged flow variables, and $P\left(\tilde{\nu}_t, \mathbf{U}\right)$, $D\left(\tilde{\nu}_t, \mathbf{U}\right)$, and $T\left(\tilde{\nu}_t, \mathbf{U}\right)$ denote the production, destruction, and transport terms, respectively. The non-dimensional field $\beta(\mathbf{x})$ is initially assumed to be one and is estimated through inverse problems like Bayesian

inversion. A field inversion algorithm (*i.e.* the solution of an inverse problem) is performed to infer the function $\beta(\mathbf{x})$. The taller is then mapped onto a vector of flow features $\eta = \eta(\mathbf{x})$ that can be used for further prediction. The mapping is generally accomplished by using Gaussian processes or supervised learning.

The field inversion approach has been applied to turbomachinery flows by Ferrero *et al.* [41], who adapted Duraisamy *et al.*'s methodology to modify the production term in the surrogate viscosity $\tilde{\nu}_t$ transport equation of the Spalart-Allmaras model using a smooth ramp function instead of $\beta(\mathbf{x})$.

Another approach, proposed by Dow and Wang [26], focuses on uncertainties in the eddy viscosity field predicted by widely-used RANS models based on Boussinesq's approximation. Their method involves computing the optimal eddy viscosity ν_t^{opt} that minimises the discrepancy between mean velocity predictions and Direct Numerical Simulation (DNS) data:

$$\nu_t^{\text{opt}} = \underset{\nu_t}{\text{argmin}}\ J, \quad \text{where} \quad J = \left\|\mathbf{U}(\nu_t) - \mathbf{U}^{\text{DNS}}\right\|_{L^2}^2 + \gamma \left\|\nabla \nu_t\right\|_{L^2}^2$$

The term $\gamma \left\|\nabla \nu_t\right\|_{L^2}^2$ ensures problem regularisation and promotes smoothness in the eddy viscosity field. They model $\delta_{\log \nu_t} = \log(\nu_t^{opt}/\nu_t^{RANS})$ as a zero-mean Gaussian process parameterised by variance σ and length scale λ, estimated via maximum likelihood estimation. Uncertainty propagation involves sampling from this Gaussian process to generate eddy viscosity realisations, subsequently used to compute an ensemble of velocity predictions.

In all the aforementioned approaches, predictions remain bounded by the limitations of Boussinesq's approximation when RANS models are based on it. To address this, some researchers have focused on introducing uncertainty directly into the Reynolds stress tensor itself.

For instance, Tracey *et al.* [101] introduced a supervised learning technique to develop a local error model. Using high-fidelity data, they established a probabilistic mapping between input features from the RANS (low-fidelity) model and outputs from the high-fidelity model. They initially derived perturbations to the Reynolds tensor eigenvalues at each spatial point $\delta_\lambda(\mathbf{x})$, using these to map $\delta_\lambda(\mathbf{x})$ via Gaussian process regression into the chosen feature space $\eta(\mathbf{x})$. Their method demonstrated significant improvements over Menter's SST $k-\omega$ turbulence model, notably in predicting anisotropy. In a data-free approach, Emory *et al.* [37] utilises realizability constraints on the Reynolds stress tensor to create realizability maps guiding the exploration of

Reynolds stress uncertainty. They perturb the anisotropy eigenvalues in the three corners of the barycentric triangle. However, these maps do not directly constrain the magnitude of turbulent kinetic energy k and anisotropy eigenvectors. Iaccarino *et al.* [51] use maximum and minimum states of turbulent kinetic energy production, identified by specific alignments between the Reynolds stress tensor and mean velocity gradient tensor, to obtain prediction intervals through simulations corresponding to these states. The approach has been extended to a data-driven framework in [34]. Another approach employing realizability constraints is proposed by Xiao *et al.* [111], where the Reynolds stress tensor is modelled as a 3×3 random matrix following a maximum entropy distribution defined on the set of positive semi-definite matrices. Uncertainty in Reynolds stress is estimated by sampling from this distribution, ensuring realizability across all samples.

In an attempt to overcome structural deficiencies intrinsic to turbulence modelling assumptions, data-driven methods for turbulence modelling based on supervised machine learning have recently been introduced. However, the more recent research work has mostly leveraged machine learning techniques for learning improved turbulence models from data, while the quantification of uncertainties associated with the predictions has been overshadowed. For example, Ling *et al.* [64] use deep neural networks (DNNs) to learn models for the Reynolds stress anisotropy tensor from high-fidelity simulation data. They propose a special network architecture, termed tensor basis neural network embedding rotational invariance, based on Pope's [82] decomposition of the normalised Reynolds stress anisotropy tensor. Recently, Ströfer and Xiao [99] extended Ling *et al.*'s work by applying DNNs to learn eddy viscosity models from indirect observations derived from velocity or pressure fields. Their framework integrates DNNs into the turbulence model, with a RANS solver computing the objective function J iteratively. The training involves gradient descent algorithms, necessitating RANS equation solutions at each step. Additionally, Weatheritt and Sandberg [105] employed Pope's decomposition in symbolic regression and gene expression programming to enhance the baseline $k-\omega$ SST RANS model. Their method seeks optimal algebraic expressions for coefficients to create explicit models implementable in RANS solvers. Training on DNS data demonstrated effective predictive capabilities for flows similar to the training data. However, their approach yields varying mathematical forms for coefficients across runs, prompting Schmelzer *et al.* [89] to introduce a method constraining mathematical form search towards sparse algebraic models. Their approach involves building a function library, selecting models using sparse regression, inferring coefficients from high-fidelity data, and cross-validating resulting models.

While achieving significant improvements in predictive accuracy, non-parametric approaches introduce challenges due to uncertainties or model corrections quantified as spatial fields with infinite degrees of freedom, leading to potentially prohibitive computational costs. Furthermore, most of the recent machine-learning-assisted turbulence models do not provide estimates of predictive uncertainties. This seriously hinders their applicability to flows outside their training set. Finally, these methods are intrusive and require meticulous implementation in CFD solvers, which is not straightforward in most industrial solvers.

While machine-learning-assisted turbulence modelling is an extremely active field of research, in this study, we deliberately restrict ourselves to non-intrusive approaches, and more precisely to a methodology involving mostly adjustments of the model parameters, the so-called parametric approaches.

Parametric approaches

Rather than working on random discrepancy fields, parametric approaches introduce uncertainties in the closure coefficients of the chosen turbulence model used to make predictions. Parametric approaches are, therefore, by essence, constrained by the baseline formulation of the model like Boussinesq's approximation. However, they have the advantage of being non-intrusive and easy to use with any black-box RANS solver allowing the user to change the closure coefficients of the chosen model. Furthermore, considerable flexibility can be gained by aggregating various calibrated models, as discussed later in this manuscript. Note that only data-driven parametric approaches are described in this section.

By definition, parametric approaches only introduce uncertainties in the closure coefficients of the RANS models considered to make predictions. At first glance, parametric approaches would only be able to quantify parametric uncertainties, as mentioned above, but Zio and Apostolakis [114] proposed two methods to assess model-form uncertainty. One of them consists of introducing an adjusting factor on the model prediction to account for the uncertainty associated with it. The other one is a model-mixture approach whereby the prediction is constructed as the average of predictions from different models. In their seminal work, Kennedy and O'Hagan [55] follow the first approach and suggest representing the relationship between the true process ζ and the computer code output y parametrized by $\boldsymbol{\theta}$ given the observation point $\mathbf{x}$:

$$\zeta(\mathbf{x}) = y(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x})$$

where δ is a model inadequacy function, independent of the code output, which needs to be modelled. They model the prior knowledge about the inadequacy function as a Gaussian process parametrised by a set of hyper-parameters $\boldsymbol{\phi}$. The complete set of parameters in need of calibration is then constituted by the model parameters $\boldsymbol{\theta}$ and the hyper-parameters $\boldsymbol{\phi}$.

Following Kennedy and O'Hagan's method, Cheung *et al.* [17] not only calibrated the Spalart-Allmaras model coefficients but also formulated three different stochastic representations of model inadequacy. They used a multiplicative factor and modelled the observed data $D(\mathbf{x})$ as follows :

$$D(\mathbf{x}) = \zeta(\mathbf{x}) + e(\mathbf{x}) = \delta(\mathbf{x})y(\mathbf{x}, \boldsymbol{\theta}) + e(\mathbf{x})$$

where $e(\mathbf{x})$ represents the observational uncertainty (e.g. experimental noise) and follows a normal distribution $e(\mathbf{x}) \sim \mathcal{N}(0, \sigma_e)$. The stochastic models used in their study are (1) an adequate model $\delta_1(\mathbf{x}) = 1$, (2) an independent Gaussian uncertainty $\delta_2(\mathbf{x}) \sim \mathcal{N}(1, \sigma^2)$, σ being the standard deviation of the distribution and (3) a correlated Gaussian uncertainty $\boldsymbol{\delta}_3(\mathbf{x}) \sim \mathcal{N}(\mathbf{1}, K_m)$, K_m being the covariance matrix determined from a Gaussian random field model of the velocity field. The hyper-parameters involved in modelling the experimental or model-form uncertainty are calibrated against experimental data alongside the model parameters. Cheung *et al.*'s model-form uncertainty models are developed for calibrating RANS models using the mean flow velocity, which is both the observable used for calibration and the quantity of interest. However, this approach is not general enough for more complex flows. For example, since the Reynolds stress tensor cannot be computed given only the velocity field, the previous approaches fail when the Reynolds stress is used for calibration or is predicted. Oliver and Moser [77] make two significant extensions of Cheung *et al.*'s work. First, by considering three additional RANS models allowing model comparison. Second, by developing new stochastic models in terms of Reynolds stress. The sixteen models are then calibrated against DNS data of a channel flow and are compared based on two items: posterior plausibility, indicating which model is preferred according to the data and predictions of the quantity of interest. Extending their work, Edeling *et al.* [33, 35] calibrated four RANS models and their model-form and experimental uncertainties models against fourteen different experimental data sets of wall-bounded flows with favourable or adverse pressure gradients. Their results suggest that the "best" (most likely) model can change significantly from one flow case to another, meaning that there is no best universal model that will outperform the other ones for every scenario considered, despite only a class of 2D turbulent boundary layers being considered.

To better account for model-form variability, Zio and Apostalakis [114] have proposed a model-mixture approach that centres around the idea of alternative hypotheses, where a set of models, each with varying plausibility, are combined. They suggest that predictions should be derived by integrating the outputs of these different models. This approach is built on two key assumptions: the mutual exclusivity of the models and the collective exhaustiveness of the model set. As verifying the second assumption is often impractical, the authors acknowledge that predictions are necessarily conditional on the selected models used to describe the phenomenon.

In data-driven research, the Bayesian framework provides a natural foundation for combining models. Draper [27] promoted a mathematical method known as Bayesian Model Averaging (BMA), offering a systematic approach for managing and propagating model-form uncertainty. Within this Bayesian context, predictions from different models are averaged according to their posterior probabilities. Hoeting *et al.* [49] provided a comprehensive tutorial on the practical implementation of BMA, which has subsequently been employed across a range of fields. Fragoso *et al.* [42] reviewed the growing interest in BMA, observing an exponential rise in related publications, with around 120 appearing annually by 2016. This surge can be attributed to advances in computational tools, increased processing power, and the availability of MCMC algorithms, all of which have made BMA more computationally feasible. Fragoso and colleagues highlighted its application in fields ranging from economics to genetics, with its primary use in engineering found in hydrology, meteorology, and climatology, following the work of Raftery *et al.* [84]. Raftery's extension of BMA to dynamic systems, such as weather forecasting, demonstrated the method's improved predictive accuracy and sharper confidence intervals.

An extension to BMA, initially discussed by Draper [27] but formally introduced by Meyer *et al.* [74], incorporates the concept of "scenarios." In this framework, a scenario is defined as a set of alternative future conditions, encompassing the system's geometry, governing equations, and influencing factors. This extension, termed Bayesian Model and Scenario Averaging (BMSA), generates predictions as the weighted average of BMA predictions across different scenarios, using scenario probabilities as weights. Like BMA, BMSA has been applied successfully in hydrology, as evidenced by the work of Meyer *et al.* and Rojas *et al.* [86].

In the context of turbulence modelling, Edeling *et al.* [35] introduced BMSA to predict quantities of interest in turbulent boundary layer scenarios. This work was followed by applications in compressor cascades [20, 22] and 3D wing configurations [36]. Edeling calibrated four RANS models across fourteen different scenarios, comparing uniform probability distributions with a

more refined approach that favoured scenarios similar to the one being predicted. Their findings indicated that the latter method significantly reduced the variance in predictions. Building on this, de Zordo-Banliat *et al.* [22] examined three weighting strategies for BMSA, ultimately showing that more complex criteria, while computationally demanding, outperformed simpler approaches in terms of predictive accuracy.

One of the major challenges in applying BMSA to turbulent flow predictions lies in its computational cost, given the frequent need to invoke CFD solvers, particularly in turbomachinery applications. The cost arises during the calibration phase when multiple calls to the CFD solver are needed to compute the full posterior or MAP estimate of the parameters, but also in the prediction phase, due to the need to propagate the posterior parameter distributions through the solver for all models in the mixture. To alleviate the latter problem, Edeling *et al.* [36] suggested using Maximum A Posteriori (MAP) estimates to approximate the posterior distribution, thus avoiding the need for full parameter distribution propagation. On the other hand, De Zordo-Banliat *et al.* [20] employed surrogate models to reduce computational demands associated with both calibration and propagation. Such an approach yielded results with minimal loss of accuracy.

Bayesian Model (Scenario) Averaging can be seen as a stochastic variant of the Model Aggregation framework, as developed by Stoltz [98] and others. This approach, often referred to as Sequential Model Aggregation (SMA), is part of a broader family of methods known as Multiplicative Weight Updating Algorithms [13]. These methods combine multiple predictions from various models (or experts) to generate a collective, improved solution, a concept widely recognised in machine learning as the "wisdom of the crowd." In SMA, weights are updated using the Exponentially Weighted Average (EWA), a loss-based function, and may or may not involve model parameter updates during training. While BMA applies a Bayesian method to compute posterior model probabilities, it typically assigns uniform weights across the spatial domain, which may not be optimal, as model performance often varies with local flow physics. For example, Edeling *et al.* [36] demonstrated that scenario probabilities for transonic flow around a wing differed significantly at different spanwise locations, underscoring the need for region-specific model weighting.

Some ensemble methods, such as Mixture-of-local-Experts [52] and Mixture Models, offer space-varying weights, with models assigned higher weights in regions where they perform best. These methods typically rely on parametric gate functions to partition the input space and soft-max functions to determine model probabilities, with parameters trained via Expectation

Maximisation [24]. However, while effective at promoting the best models in each partition, this approach tends to overlook uncertainty in model selection.

Seeking to integrate the strengths of BMA and Mixture-of-Experts, Yu *et al.* [113] developed Clustered Bayesian Averaging (CBA), which allows for spatially varying model weights. CBA uses local Bayes factors to inform a clustering algorithm that partitions the spatial domain, with model parameters iteratively updated using a subset of the data. Though successful in fields such as hydrology and solid mechanics, CBA has yet to be extended to CFD applications. Nonetheless, its ability to adjust model weights based on local performance holds promise for improving turbulence modelling in CFD.

Inspired by both CBA and SMA, de Zordo-Banliat *et al.* [19] introduced the space-dependent Model Aggregation (XMA) algorithm, designed specifically for CFD. Unlike previous BMSA implementations [35, 36, 20, 22], XMA incorporates spatially varying weights. By simplifying the CBA algorithm and employing an EWA loss function, XMA circumvents the need for model calibration and reduces computational costs, while assigning weights based on flow features rather than physical coordinates. The XMA algorithm has been shown to outperform individual RANS models for all flow quantities of interest.

1.2 Thesis objectives and outline

This thesis aims to develop a framework for quantifying and reducing uncertainties in RANS turbulence models through non-intrusive Bayesian calibration and model aggregation methods. Building on the work of De Zordo-Banliat *et al.* [21], which introduced a surrogate-model-based Bayesian calibration framework and the XMA strategy for model aggregation, our objectives are:

1. **Enhance Calibration Efficiency**: Improve the scalability of surrogate models for calibrating large sets of space-dependent data, such as full velocity or pressure fields.
2. **Efficient Posterior Estimation**: Implement a gradient-assisted algorithm to efficiently find model parameters' posterior distribution and MAP.
3. **Integrate XMA with Bayesian Calibration**: Combine the XMA model aggregation approach with Bayesian calibration to enhance model accuracy.
4. **Optimise Data Selection and Model Aggregation**: Address the challenge of selecting suitable calibration data and optimally combining specialised models. This involves exploring:

- A clustering-based approach (inspired by CMA [113]), which partitions data, calibrates models within each cluster and aggregates them based on cluster probabilities.
- A "building-block" approach (inspired by Lozano-Duran et al. [66]), which calibrates models on specific flow phenomena and combines them using XMA.

5. **Apply to 3D Turbomachinery**: For the first time, apply model aggregation techniques to a 3D turbomachinery configuration, specifically a linear cascade of compressor blades.

The manuscript is organised as follows: Chapter 2 reviews the governing equations and turbulence models and briefly describes the numerical solver used in the simulations. Chapter 3 introduces Bayesian inference concepts relevant to RANS model analysis, discusses numerical algorithms for sampling posterior distributions, Bayesian model mixtures for model-form uncertainties, and the use of Gaussian Process surrogates for efficient calibration and inference. Chapter 4 outlines the key flow configurations analysed, including their operating conditions and data for calibration, training, and validation. Chapter 5 investigates the XMA-based model aggregation method, focusing on calibrating component models before aggregation and applying this approach to the NACA65 V103 linear cascade. Chapter 6 presents a calibration approach leveraging clustering techniques to create specialised models, using fuzzy clustering for better prediction accuracy and generalisation. Finally, Chapter 7 develops a methodology using "building-block" flows for creating specialised models, and aggregates these models using XMA to ensure robust generalisation. The conclusion summarises findings and proposes future directions for research.

2

Governing equations, turbulence models and flow solver

In this chapter, we recall the governing equations and turbulence models considered throughout the study and a short description of the numerical solver we used for all the simulations.

Chapter contents

2.1 Governing equations

2.1.1 Navier–Stokes equations

Throughout the present study, we consider single-phase compressible flows in turbomachinery configurations. The latter can be modelled using the compressible Navier-Stokes equations, written as:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \dfrac{\partial (\rho u_j)}{\partial x_j} = 0 \\ \dfrac{\partial (\rho u_i)}{\partial t} + \dfrac{\partial (\rho u_i u_j)}{\partial x_j} = -\dfrac{\partial p}{\partial x_i} + \dfrac{\partial \tau_{ij}}{\partial x_j} \\ \dfrac{\partial (\rho E)}{\partial t} + \dfrac{\partial (\rho H u_j)}{\partial x_j} = -\dfrac{\partial q_j}{\partial x_j} + \dfrac{\partial (\rho u_i \tau_{ij})}{\partial x_j} \end{cases} \tag{2.1}$$

where ρ, u_i, p, E and H are respectively the density, the velocity components along the i-th Cartesian coordinate, the pressure, the total energy, $E = e + \frac{1}{2} u_i u_i$ where e is the internal energy per mass unit and the total enthalpy $H = E + P/\rho$. Einstein's convention of repeated indices is applied throughout this chapter. Note that no volume forces are considered here. The gas is supposed to be thermally and calorically perfect, so that $p = \rho r T$ and $e = \frac{r}{\gamma - 1} T$ with the specific heat ratio $\gamma = 1.4$ and the gas constant $r = 287 J.kg^{-1}.K^{-1}$ for air.

The viscous stress tensor τ_{ij} is expressed for a Newtonian fluid as:

$$\tau_{ij} = \mu \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] + \lambda_2 \frac{\partial u_i}{\partial x_i} \tag{2.2}$$

where μ, λ_2 designate the dynamic viscosity and the second viscosity. Making use of the Stokes hypothesis, the latter is set as $\lambda_2 = -\frac{2}{3}\mu$. Additionally, $\mu = \mu(T)$ is modelled through Sutherland's law, $\mu = \frac{C_1 T^{3/2}}{T+S}$ with $S = 110.4K$ and $C_1 = 1.458 \times 10^{-6} kg.m^{-1}s^{-1}.K^{-1/2}$ for air. The heat flux components q_j are modelled through Fourier's law $q_j = -\kappa \dfrac{\partial T}{\partial x_j}$, with λ the

thermal conductivity computed by ensuring a constant Prandtl number $Pr = \frac{c_p \lambda}{\mu} = 0.7$ for air, with $c_p = \frac{\gamma r}{\gamma - 1}$.

2.1.2 Reynolds-Averaged Navier–Stokes equations

The Reynolds decomposition is used for *incompressible* flows where density is supposed to be constant. The velocity and pressure are then decomposed in a mean and a fluctuating part, where the mean is generally taken over time for statistically steady flows [108] :

$$u_i = \overline{u}_i + u'_i, \quad p = \overline{p} + p'$$

where $\overline{\bullet}$ denotes the averaging operator. After introducing this decomposition in the Navier-Stokes equations 2.1 and then averaging, one obtains the Reynolds-Averaged Navier-Stokes equations, written as:

$$\begin{cases} \dfrac{\partial \overline{u}_i}{\partial x_i} = 0 \\ \\ \dfrac{\partial \overline{u}_i}{\partial t} + \dfrac{\partial (\overline{u}_i \overline{u}_j)}{\partial x_j} = -\dfrac{1}{\rho} \nabla \overline{p} + \dfrac{1}{\rho} \dfrac{\partial}{\partial x_j} \left(\overline{\tau_{ij}} - \rho \overline{u'_i u'_j} \right). \end{cases} \tag{2.3}$$

A new term called the Reynolds tensor $\tau_{ij}^R = -\rho \overline{u'_i u'_j}$ arises from the averaging process and needs to be modelled to close the problem.

The mean kinetic energy of the fluctuating field is called the turbulent energy,

$$k_t = \frac{u'_i u'_i}{2}$$

and it is related to the trace of the Reynolds stress tensor :

$$k_t = -\frac{1}{2\rho} \tau_{ii}^R$$

2.1.3 Favre-Averaged Navier-Stokes equations

In the case of compressible flows, the variable density leads to unclosed terms appearing in the mass equation if the Reynolds decomposition is applied to all variables. For this reason, a density-weighted decomposition (initially introduced by Favre) is used. For any variable f, the

Favre decomposition reads:

$$f = \widetilde{F} + f'' \text{ with } \widetilde{F} = \frac{\overline{\rho f}}{\overline{\rho}} \tag{2.4}$$

In this decomposition, $\overline{\rho f''} = 0$ but $\overline{f''} = -\frac{\overline{\rho' f'}}{\overline{\rho}}$. The Favre decomposition is used for every variable except for density and pressure, to which the Reynolds decomposition is applied. The Favre-averaged Navier-Stokes equations are finally derived as follows :

$$\begin{cases} \dfrac{\partial \overline{\rho}}{\partial t} + \dfrac{\partial (\overline{\rho}\widetilde{u}_j)}{\partial x_j} = 0 \\ \dfrac{\partial (\overline{\rho}\widetilde{u}_i)}{\partial t} + \dfrac{\partial (\overline{\rho}\widetilde{u}_i\widetilde{u}_j)}{\partial x_j} = -\dfrac{\partial \overline{p}}{\partial x_i} + \dfrac{\partial}{\partial x_j}\left(\overline{\tau}_{ij} - \overline{\rho}\widetilde{u_i'' u_j''}\right) \\ \dfrac{\partial (\overline{\rho}\widetilde{E})}{\partial t} + \dfrac{\partial (\overline{\rho}\widetilde{H}\widetilde{u}_j)}{\partial x_j} = \dfrac{\partial}{\partial x_j}\left[-\overline{q}_j - \overline{\rho u_j'' h''} + \overline{u_i'' \tau_{ij}'} - \overline{u_j'' \frac{1}{2} u_k'' u_k''}\right] + \dfrac{\partial}{\partial x_j}\left[\widetilde{u}_i\left(\overline{\tau}_{ij} - \overline{\rho u_i'' u_j''}\right)\right] \end{cases} \tag{2.5}$$

The turbulent kinetic energy is now defined as:

$$k_t = \frac{1}{2}\widetilde{u_i'' u_i''} = \frac{1}{2}\frac{\overline{\rho u_i'' u_i''}}{\overline{\rho}} \tag{2.6}$$

The utility of using the Favre decomposition for compressible flows is noticeable as the structure of the averaged equations is preserved when using the Favre decomposition.

2.2 RANS models

The non-linearity of the Navier–Stokes equations results in the appearance of unclosed terms Reynolds or Favre-Averaged Navier-Stokes equations, the Reynolds stress tensor $\boldsymbol{\tau}^R$ first of all. To close the system of equations the Reynolds stress tensor and other unclosed terms must be expressed as a function of the mean flow variables, leading to the fundamental closure problem in RANS modelling. The closure problem remains a central challenge in computational fluid dynamics, influencing the accuracy and applicability of models across different flow regimes and geometries. Selecting an appropriate closure model involves balancing accuracy with computational feasibility, making it a critical area of ongoing research.

This section briefly reviews the models considered in the following of this study.

2.2.1 Boussinesq's approximation

By analogy with the viscous stress tensor τ, Boussinesq introduces the oldest and widely used closure for the Reynolds stress tensor $\boldsymbol{\tau}_R$ based on the concept of turbulent viscosity. It is assumed that the deviatoric part of $\boldsymbol{\tau}_R$ behaves analogously to the viscous stresses in a Newtonian fluid. In the incompressible flow case, this comes to write:

$$-\rho\overline{u_i'u_j'} = 2\mu_t\bar{S}_{ij} - \frac{2}{3}\rho k_t\delta_{ij} \tag{2.7}$$

where μ_t is called the dynamic turbulent viscosity (or eddy viscosity), and $\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial\overline{u}_i}{\partial x_j} + \frac{\partial\overline{u}_j}{\partial x_i}\right)$ is the mean strain rate tensor. The isotropic term at the end of Eq. 2.7 is introduced to satisfy the condition $-\overline{\rho u_i'u_i'} = -2\rho k_t$.

(i) In contrast to the molecular viscosity which is an intrinsic physical property of the fluid, the turbulent viscosity is a function of the flow, i.e., $\mu_t = \mu_t(\mathbf{x}, t)$. The challenge is to derive a formulation for μ_t.

(ii) Replacing the three normal stresses $-\rho\overline{u_i'^2}$ and the three shear stresses $-\rho\overline{u_i'u_j'}$ for $i \neq j$, by a unique function μ_t induces a one-way energy transfer from the mean flow field towards the fluctuating field. This can be seen via the turbulent kinetic energy production term (see [83]) if $\mu_t \geq 0$:

$$\mathcal{P} = -\rho\overline{u_i'u_j'}\frac{\partial\bar{u}_i}{\partial x_j} = \left(2\mu_t\bar{S}_{ij} - \frac{2}{3}\rho k_t\delta_{ij}\right)\frac{\partial\bar{u}_i}{\partial x_j} = 2\mu_t\bar{S}_{ij}^2 \geq 0$$

(iii) The Boussinesq hypothesis implies that the anisotropy tensor a_{ij}, i.e. the deviatoric part of τ_{ij}^R, is aligned with the mean rate-of-strain tensor, i.e.,

$$a_{ij} \equiv -\overline{\rho}\left(\overline{u_i'u_j'} - \frac{2}{3}k_t\delta_{ij}\right) = \mu_t\left(\frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i}\right) = 2\mu_t\bar{S}_{ij}$$

Both a_{ij} and the mean rate of strain have five independent components. The turbulent viscosity hypothesis suggests that these five components are related to each other through the scalar coefficient μ_t. However, even in simple shear flow, this alignment does not occur.

2.2.2 Linear eddy viscosity models

The turbulence models most widely employed in engineering applications rely on this Boussinesq approximation, which postulates a linear relationship between the Reynolds stresses and the mean velocity gradients via the eddy viscosity coefficient. For this reason, such models are called linear eddy viscosity models (LEVM). The LEVM considered in the present study are further described in the following subsections. For the sake of simplicity, in the following, we present the incompressible formulation of the models (in our numerical solver, the left-hand sides are written in conservation form), and we drop the Reynolds/Favre averaging symbols from the mean-flow quantities so that, e.g. the mean velocity is just noted u_j, the mean pressure p, etc.

Spalart-Allmaras one equation model

The Spalart-Allmaras model is written in terms of the free-flow kinematic eddy viscosity $\tilde{\nu}$. The model includes eight closure coefficients and three damping functions. Its defining equations are [97]:

$$\nu_t = \tilde{\nu} f_{v1} \tag{2.8}$$

where $\nu_t = \mu_t/\overline{\rho}$

$$\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}\tilde{S}\tilde{\nu} - c_{w1} f_w \left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma}\frac{\partial}{\partial x_k}\left[(\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_k}\right] + \frac{c_{b2}}{\sigma}\frac{\partial \tilde{\nu}}{\partial x_k}\frac{\partial \tilde{\nu}}{\partial x_k}. \tag{2.9}$$

The closure coefficient and the auxiliary relations are the following :

$$c_{b1} = 0.1355, \quad c_{b2} = 0.622, \quad c_{v1} = 7.1, \quad \sigma = 2/3 \tag{2.10}$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{(1 + c_{b2})}{\sigma}, \quad c_{w2} = 0.3, \quad c_{w3} = 2, \quad \kappa = 0.41 \tag{2.11}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}. \tag{2.12}$$

Where,

$$\chi = \frac{\tilde{\nu}}{\nu}, \quad g = r + c_{w2}\left(r^6 - r\right), \quad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2} \tag{2.13}$$

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad S = \sqrt{2\Omega_{ij}\Omega_{ij}} \tag{2.14}$$

in the above, the tensor $\Omega_{ij} = \frac{1}{2}\left(\partial u_i/\partial x_j - \partial u_j/\partial x_i\right)$ is the mean rotation tensor and d is the distance from the closest surface.

$k-\omega$ Wilcox (2006) two-equation model

The $k-\omega$ model is a two-equation model where the ω equation has evolved. This version, known as the Wilcox (2006) model, uses transport equations for k_t (often noted k in the literature) and the specific dissipation rate ω [108] :

$$\nu_t = \frac{k_t}{\tilde{\omega}}, \quad \tilde{\omega} = \max\left\{\omega, \quad C_{lim}\sqrt{\frac{2S_{ij}S_{ij}}{\beta^*}}\right\}, \quad C_{lim} = \frac{7}{8} \tag{2.15}$$

The evolution of the turbulence kinetic energy k_t is governed by the following equation:

$$\frac{\partial k_t}{\partial t} + u_j\frac{\partial k_t}{\partial x_j} = -\frac{1}{\rho}\tau_{ij}^R\frac{\partial u_i}{\partial x_j} - \beta^* k_t\omega + \frac{\partial}{\partial x_j}\left[\left(\nu + \sigma^*\frac{k_t}{\omega}\right)\frac{\partial k_t}{\partial x_j}\right]. \tag{2.16}$$

Similarly, the specific dissipation rate ω evolves according to the equation:

$$\frac{\partial \omega}{\partial t} + u_j\frac{\partial \omega}{\partial x_j} = -\frac{1}{\rho}\alpha\frac{\omega}{k}\tau_{ij}^R\frac{\partial u_i}{\partial x_j} - \beta\omega^2 + \frac{\sigma_d}{\omega}\frac{\partial k_t}{\partial x_j}\frac{\partial \omega}{\partial x_j} + \frac{\partial}{\partial x_j}\left[\left(\nu + \sigma\frac{k_t}{\omega}\right)\frac{\partial \omega}{\partial x_j}\right]. \tag{2.17}$$

To complete the model, the closure coefficients and auxiliary relations are specified as follows:

$$\alpha = \frac{13}{25}, \quad \beta = \beta_o f_\beta, \quad \beta^* = \frac{9}{100}, \quad \sigma = \frac{1}{2}, \quad \sigma^* = \frac{3}{5}, \quad \sigma_{do} = \frac{1}{8}. \tag{2.18}$$

The auxiliary function σ_d is defined piecewise:

$$\sigma_d = \begin{cases} 0, & \frac{\partial k_t}{\partial x_j}\frac{\partial \omega}{\partial x_j} \leq 0 \\ \sigma_{do}, & \frac{\partial k_t}{\partial x_j}\frac{\partial \omega}{\partial x_j} > 0. \end{cases} \tag{2.19}$$

Additional relations include:

$$\beta_o = 0.0708, \quad f_\beta = \frac{1+85\chi_\omega}{1+100\chi_\omega}, \quad \chi_\omega \equiv \left|\frac{\Omega_{ij}\Omega_{jk}S_{ki}}{(\beta^*\omega)^3}\right|, \quad \epsilon = \beta^*\omega k_t, \quad \ell = k_t^{1/2}/\omega. \tag{2.20}$$

Smith $k-\ell$ Model

The $k-\ell$ model, initially proposed by Smith [95], uses transport equations for the turbulent kinetic energy k_t and the turbulent length scale l_t. This model is a simplified version of Smith's $k-kL$ model [94], specifically formulated for the wall region. The turbulent viscosity in this model is defined as:

$$\mu_t = \mu \chi f_\mu \tag{2.21}$$

The evolution of the turbulence kinetic energy k_t is governed by the following transport equation:

$$\frac{\partial k_t}{\partial t} + u_j \frac{\partial k_t}{\partial x_j} = -\frac{1}{\rho}\tau_{ij}^R \frac{\partial u_i}{\partial x_j} - \frac{(2k_t)^{3/2}}{B_1 l} - 2\nu \left(\frac{\partial \sqrt{k_t}}{\partial x_j}\right)^2 + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k_t}{\partial x_j}\right] \tag{2.22}$$

Similarly, the turbulent length scale l_t is described by the following transport equation:

$$\begin{aligned}\frac{\partial l}{\partial t} + u_j \frac{\partial l}{\partial x_j} = (2 - E_2)\frac{\sqrt{2k_t}}{B_1}\left[1 - \left(\frac{l}{\kappa \eta}\right)^2\right] - \frac{\nu_t}{\sigma_l}\frac{1}{l}\left(\frac{\partial l}{\partial x_j}\right)^2 \left(\frac{l}{\kappa \eta}\right)^2 \\ + l\frac{\partial u_j}{\partial x_j} + 2\frac{\nu_t}{\sigma_l}\frac{1}{k_t}\left(\frac{\partial l}{\partial x_j}\frac{\partial k}{\partial x_j}\right) + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_l}\right)\frac{\partial l}{\partial x_j}\right]\end{aligned} \tag{2.23}$$

To close the model, the closure coefficients and auxiliary relations are provided as follows:

$$f_1 = \exp\left[-50\left(\frac{l}{\kappa \eta}\right)^2\right], \quad c_1 = 25.5, \quad c_2 = 2, \tag{2.24}$$

$$B_1 = 18, \quad E_2 = 1.2, \quad \kappa = 0.41, \quad \sigma_k = 1.43, \quad \sigma_l = 1.43 \tag{2.25}$$

The turbulent viscosity multiplier is computed as:

$$\chi = \frac{\rho l \sqrt{2k_t}}{\mu B_1^{1/3}}, \quad f_\mu = \left(\frac{c_1^4 f_1 + c_2^2 \chi^2 + \chi^4}{c_1^4 + c_2^2 \chi^2 + \chi^4}\right)^{1/4} \tag{2.26}$$

Launder–Sharma $k-\epsilon$ Two-Equation Model

The $k-\epsilon$ model was the most popular two-equation turbulence model until the 1990s. In 1974, Launder and Sharma recalibrated its closure coefficients and developed the Standard $k-\epsilon$ model, described as follows [58]:

The kinematic eddy viscosity is expressed as:

$$\nu_t = \frac{C_\mu f_\mu k_t^2}{\epsilon} \tag{2.27}$$

The transport equation for the turbulence kinetic energy k_t is given by:

$$\frac{\partial k_t}{\partial t} + u_j \frac{\partial k_t}{\partial x_j} = \tau_{ij}^R \frac{\partial u_i}{\partial x_j} - \epsilon + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k_t}{\partial x_j}\right] - D_k \tag{2.28}$$

The dissipation rate ϵ is governed by the following equation:

$$\frac{\partial \epsilon}{\partial t} + u_j \frac{\partial \epsilon}{\partial x_j} = C_{\epsilon 1}\frac{\epsilon}{k_t}\tau_{ij}^R \frac{\partial u_i}{\partial x_j} - f_2 C_{\epsilon 2}\frac{\epsilon^2}{k_t} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] \tag{2.29}$$

The closure coefficients and auxiliary relations are defined as:

$$C_{\epsilon 1} = 1.44, \quad C_{\epsilon 2} = 1.92, \quad C_\mu = 0.09, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.3 \tag{2.30}$$

In addition to the main equations, the model incorporates viscous damping functions for low-Reynolds-number corrections:

$$f_2 = 1 - 0.3e^{-Re_T^2}, \quad f_\mu = \exp\left(\frac{-3.4}{(1 + Re_T/50)^2}\right) \tag{2.31}$$

where the turbulence Reynolds number Re_T is defined as:

$$Re_T = \frac{\rho k_t^2}{\tilde{\varepsilon}\mu} \tag{2.32}$$

The model also includes a damping source term to account for near-wall effects:

$$D_k = 2\nu \frac{\partial \sqrt{k_t}}{\partial x_i}\frac{\partial \sqrt{k_t}}{\partial x_i} \tag{2.33}$$

2.2.3 QCR correction for Linear Eddy Viscosity models

The Quadratic Constitutive Relation (QCR) correction enhances Linear Eddy Viscosity Models by incorporating a quadratic term in the Reynolds stress tensor to better capture turbulence anisotropy. Traditional LEVMs, while computationally efficient, often fail to accurately predict flows with significant anisotropic effects, such as secondary flows and flows with strong streamline curvature. The QCR correction, proposed by Spalart [96], addresses these shortcomings by adding a term that accounts for the nonlinear relationship between the Reynolds stress and the mean strain rate.

The Reynolds stress tensor with the QCR correction is expressed as:

$$\tau_{ij}^{R} = 2\rho\nu_t S_{ij} - \frac{2}{3}\rho k_t \delta_{ij} + \tau_{ij}^{QCR} \tag{2.34}$$

where τ_{ij}^{QCR} is the quadratic correction term. The QCR term τ_{ij}^{QCR} is given by:

$$\tau_{ij}^{QCR} = -QCR\left[O_{ik}\tau_{jk}^{B} + O_{jk}\tau_{ik}\right] \tag{2.35}$$

where τ_{ij}^{B} are the turbulent stresses computed from the Boussinesq relation, and O_{ik} is an antisymmetric normalised rotation tensor, defined by:

$$O_{ik} = 2\Omega_{ik} / \sqrt{\frac{\partial u_m}{\partial x_n}\frac{\partial u_m}{\partial x_n}} \tag{2.36}$$

Here, QCR is a model constant, typically taken as 0.3.

By including the QCR correction, LEVMs can more accurately predict the behaviour of turbulent flows with anisotropic characteristics, improving their applicability and reliability in engineering analyses.

2.2.4 EARSM $k - k\ell$ model

Similarly to the QCR correction, Explicit Algebraic Reynolds Stress Models (EARSM) use a non-linear constitutive relation for the Reynolds stresses, capturing turbulence anisotropy without solving the full set of Reynolds stress transport equations as so-called differential Reynolds stress models. Many of them have been proposed in the literature. The present study considers an EARSM model based on transport equations for k_t and the quantity $k_t\ell$, ℓ being the turbulent scale.

The selected model in this context is the EARSM $k-k\ell$ model, which combines the strengths of EARSM with the two-equation turbulence models by using the turbulent kinetic energy (in our notations k_t) and turbulent length scale ℓ to determine a characteristic time for turbulence.

EARSM models are obtained by assuming the local equilibrium of turbulence. Under this assumption, it is possible to neglect advection and diffusion terms in the Reynolds stress transport equations and, after some algebraic manipulations, explicit algebraic expressions for the Reynolds stresses are obtained, i.e., a nonlinear constitutive relation. Wallin & Johansson [102] proposed the following constitutive relation:

$$\tau_{ij}^R = -\rho k_t \left(\frac{2}{3}\delta_{ij} - 2C_\mu^{\text{eff}} \frac{k_t}{\varepsilon} S_{ij} + a_{ij}^{(\text{cx})} \right) \tag{2.37}$$

The preceding expression differs from the standard linear eddy viscosity models for two reasons: first, the C_μ^{eff} coefficient is not constant but depends on the velocity gradients and turbulent quantities, the last term $a_{ij}^{(\text{cx})}$ is non-linear and represents the extra anisotropy of the Reynolds stress, which is a complex function of the velocity gradients and turbulent quantities. The full expressions of C_μ^{eff} and $a_{ij}^{(\text{cx})}$ proposed by Wallin & Johansson can be found in Appendix A of their work. Their model for the anisotropy tensor has the following closure coefficients:

$$C_1 = 1.80, \quad C_1' = 0.0, \quad C_2 = 0.80, \quad C_3 = 2.0, \quad C_4 = \frac{10}{9} \tag{2.38}$$

The preceding constitutive relation is supplemented with transport equations for the turbulent kinetic energy k_t and the product of the turbulent kinetic energy with the turbulent length scale $k_t\ell = \frac{k_t^{5/2}}{\varepsilon}$ proposed by Smith (1978). Specifically, we used the equations given in (1979),

corresponding to the "Low Reynolds" version of the EARSM $k-k\ell$ model. The equations write:

$$\begin{cases} \nu_t = C_\mu f_\mu \dfrac{\Phi}{\sqrt{k}} \\ \dfrac{\partial k_t}{\partial t} + u_j \dfrac{\partial k_t}{\partial x_j} = \tau_{ij}^R \dfrac{\partial u_i}{\partial x_j} - \dfrac{k_t^{5/2}}{\Phi} - \dfrac{2\nu k_t}{\eta^2} + \dfrac{\partial}{\partial x_j}\left[\left(\nu + \dfrac{\nu_t}{\sigma_k}\right)\dfrac{\partial k_t}{\partial x_j}\right] \\ \dfrac{\partial(\rho\Phi)}{\partial t} + \dfrac{\partial\left(\rho u_j \Phi\right)}{\partial x_j} = C_{\Phi_1}\dfrac{\Phi}{k}\tau_{ij}^R\dfrac{\partial u_i}{\partial x_j} - C_{\Phi_2}k_t^{3/2} - C_{\Phi\omega}f_\omega\dfrac{\Phi^{5/2}}{\nu^2\sqrt{d}} \\ +2C_{\Phi k}\dfrac{\nu_t}{\sqrt{k_t}}\dfrac{\partial\sqrt{k_t}}{\partial x_j}\dfrac{\partial\Phi}{\partial x_j} + 4C_{\Phi\Phi}\nu_t\left(\dfrac{\partial\sqrt{\Phi}}{\partial x_j}\right)^2 \end{cases} \tag{2.39}$$

where $\Phi = k_t\ell$ and d is the wall distance.

The auxiliary functions necessary for the damping in the near-wall regions are defined as:

$$f_\omega = \exp\left(-C_{\omega 1}R_d\right) \quad , \quad R_d = \frac{d\sqrt{k_t}}{\nu} \quad , \quad f_\mu = 1 - \exp\left(-C_{\omega 2}R_d^{3/2}\right) \tag{2.40}$$

The $k-k\ell$ model has 11 closure coefficients with the following standard values :

$$\begin{gathered} C_{\Phi_1} = 1, C_{\Phi_2} = 0.58 \quad , \quad C_\mu = 0.09 \quad , \quad C_{\Phi k} = 1.53 \quad , \quad C_{\Phi\Phi} = -1.38, \quad \sigma_k = 0.9 \\ \sigma_\Phi = 1.6 \quad , \quad C_{\Phi\omega} = 0.00077 \quad , \quad C_{\omega_1} = 0.08 \quad , \quad C_{\omega 2} = 0.00325 \quad , \quad \kappa = 0.41 \end{gathered} \tag{2.41}$$

2.3 Simulations using the elsA CFD solver

The simulations presented in this study are conducted using the CFD solver elsA version v5.1.02, developed by ONERA [59]. elsA solves the compressible Euler, Navier-Stokes, or RANS equations on both structured and unstructured multi-block grids, from the low subsonic to the high supersonic flow regime. Concerning the RANS equations, many turbulence models, from eddy viscosity to full Reynolds stress models, are implemented in the solver [102]. The system of equations is solved using a cell-centred finite volume discretisation. Space discretisation schemes include classical second-order centred or upwind schemes and higher-order schemes [59]. The Cassiopee module [8] developed at ONERA was used in the pre-processing and post-processing steps. Finally, the calibration of the RANS models also required access to each model parameter.

A path to set the values of the coefficients directly from the elsA user interface has been developed in previous work by Maximilien de Zordo-Banliat. This development has been integrated and is available for every elsA user using version v4.0.2 or higher.

2.4 Chapter take-away points

In this chapter, we presented the governing equations at stake in the rest of this study, as well as the turbulence models considered in the following analyses.

The take-away points are:

1. Throughout this work, we look for steady solutions to the steady RANS equations, supplemented with turbulence models.
2. A plethora of turbulence models are available in the literature. Here, we selected a set of competing turbulence models among those that are most used in the literature or well-established in the in-house practices of aeronautical engine manufacturers.
3. Only linear eddy viscosity or quadratic eddy viscosity models are considered here, more complex models such as Reynolds stress models being little used in the industrial practice.
4. In practice, the choice of the "best" model is a matter of discussion and is mostly case-dependent. Similarly, the model closure coefficient values are determined for simple flows and are uncertain.

In the following chapter, we will present the statistical tools used for analysing turbulence model uncertainties associated with both the closure parameters and model structure.

3

Mathematical background

This chapter contains some elements of Bayesian inference, which constitute the theoretical background of the RANS model analyses presented in later chapters. After introducing the notion of Bayesian model calibration, we review the numerical algorithms used to sample from the posterior distributions for problems that lack an analytical solution, such as those considered in this study. We then introduce the notion of Bayesian model mixtures, used to account for uncertainties in the model form. A more comprehensive introduction to Bayesian statistics can be found in [46]. Finally, we present a surrogate modelling approach, based on Gaussian Process Regressors used to replace the costly CFD solver during model calibration and inference.

Chapter contents

3.1 Bayesian Calibration

Bayesian model calibration updates prior knowledge about model parameters using experimental or high-fidelity data. The goal of this process is to provide a predictive probability density function (pdf) of a quantity of interest (QoI). One of the strongest arguments for using the Bayesian approach lies in its common-sense interpretability, as well as its flexibility in application to complex problems. In the context of Bayesian probabilities, the confidence interval for an unknown QoI is immediately understandable, as it can be interpreted as the interval with a high probability of containing the QoI. The methodology presented here is based on the work of Kennedy and O'Hagan [55] and Cheung *et al.* [17]. Predicting a QoI involves three phases: stochastic modelling, Bayesian inference, and prediction.

3.1.1 Stochastic Modelling

Let us note Δ_{obs} a vector of flow quantities (e.g. the measured velocity, pressure, etc.) observed at some observation points, gathered in the vector $\mathbf{x}$. The observed quantities differ from the (unobservable) true quantity Δ_{true} by some observation noise ε, i.e.:

$$\Delta_{obs}(\mathbf{x}) = \Delta_{true}(\mathbf{x}) + \varepsilon(\mathbf{x}) \tag{3.1}$$

We also consider a model, able to predict the same quantity under a suitable choice of parameters $\boldsymbol{\theta}$ that must be inferred from the observed data. The model output will differ from the observed data due to experimental and model-form uncertainties. In [17], such discrepancies are described using a stochastic model of the form:

$$\Delta_{obs}(\mathbf{x}) = \Delta_{true}(\mathbf{x}) + \boldsymbol{\varepsilon}(\mathbf{x}) = \boldsymbol{\eta} \odot \Delta(\mathbf{x}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}(\mathbf{x}) \tag{3.2}$$

where $\Delta(\mathbf{x}, \boldsymbol{\theta})$ is the model output at $\mathbf{x}$, given the parameter vector $\boldsymbol{\theta}$, $\boldsymbol{\eta}$ represents model-form uncertainty, and $\odot$ denotes the Hadamard (element-wise) product. Both $\boldsymbol{\varepsilon}$ and $\boldsymbol{\eta}$ are random vectors that need to be modelled and several models can be imagined to account for such uncertainties. In this study, we use a model similar to the uncorrelated model of [17], i.e. we consider an independent zero-mean Gaussian noise with a standard deviation σ_ε to account for experimental uncertainty, i.e., $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2) \quad \forall \varepsilon_i \in \boldsymbol{\varepsilon}$, and an independent Gaussian model-form uncertainty $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{I}, \sigma_\eta^2 \boldsymbol{I})$ for model-form uncertainty, with $\boldsymbol{I}$ the identity matrix. Combining these two models, Δ_{obs} follows a distribution $\mathcal{N}(\Delta, \sigma_\epsilon^2 \boldsymbol{I} + \sigma_\eta^2 \Delta)^2$. The parameters introduced σ_η and σ_ε can be elicited by expert judgement or it can be inferred from the data alongside the physical model parameters $\boldsymbol{\theta}$. In this study, the model-form uncertainty will be inferred σ_η whereas the experimental hyper-parameter σ_ε will be pre-specified. The stochastic model parameters are then referred to as hyper-parameters [17] and concatenated to the initial parameter vector $\boldsymbol{\theta}$.

3.1.2 Bayesian Inference

Bayesian inference is a powerful statistical approach that updates the probability of a hypothesis based on prior knowledge and new evidence. This method relies on Bayes' theorem, which is expressed as:

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)} \tag{3.3}$$

This theorem describes how the prior probability of event A is updated to the posterior probability given the occurrence of event B.

In the context of model calibration, the objective is to infer the posterior probability density function of the model parameters and hyper-parameters $(\boldsymbol{\theta}, \sigma_\eta)$ using observed data Δ_{obs}. This is expressed by the following equation:

$$p(\boldsymbol{\theta}, \sigma_\eta|\Delta_{obs}) = \frac{p(\Delta_{obs}|\boldsymbol{\theta}, \sigma_\eta)p(\boldsymbol{\theta}, \sigma_\eta)}{p(\Delta_{obs})} \tag{3.4}$$

Here, $p(\boldsymbol{\theta}, \sigma_\eta)$ denotes the prior pdf of the parameters, encapsulating prior knowledge or assumptions about the parameters before considering the observed data; $p(\Delta_{obs}|\boldsymbol{\theta}, \sigma_\eta)$ is the likelihood function, representing the probability of the observed data given the parameters. The term $p(\Delta_{obs})$, known as the evidence or marginal likelihood, can be computed using the law of total probability:

$$p(\Delta_{obs}) = \int p(\Delta_{obs}|\boldsymbol{\theta}, \sigma_\eta) p(\boldsymbol{\theta}, \sigma_\eta) d\boldsymbol{\theta} d\sigma_\eta \tag{3.5}$$

The likelihood function $p(\Delta_{obs}|\boldsymbol{\theta}, \sigma_\eta)$ evaluates how well the model with parameters $(\boldsymbol{\theta}, \sigma)$ predicts the observed data. It depends on the model output $\Delta(\mathbf{x}, \boldsymbol{\theta})$ and, under the statistical model in equation 3.2, takes form:

$$p(\Delta_{obs}|\boldsymbol{\theta}, \sigma_\eta) = \frac{1}{\sqrt{2\pi\sigma_\eta^2}^N} \exp\left(-\frac{|\Delta(\boldsymbol{\theta}) - \Delta_{obs}|^2}{2\sigma_\eta^2}\right) \tag{3.6}$$

The joint prior distribution of model parameters $p(\boldsymbol{\theta}, \sigma_\eta)$ can be informative, established by expert judgement, or uninformative, such as a large uniform distribution if no prior knowledge is available. The evidence term, independent of the parameter vector, acts as a normalisation constant and is not required for the calibration step.

The stochastic model parameters are then referred to as hyper-parameters [17] and concatenated to the initial parameter vector. We then want to infer $(\boldsymbol{\theta}, \sigma_\eta)$ assuming independence between parameters and hyper-parameters, the joint posterior distribution of the augmented parameter vector is:

$$p(\boldsymbol{\theta}, \sigma_\eta|\Delta_{obs}) \propto p(\Delta_{obs}|\boldsymbol{\theta}) p(\sigma_\eta) \tag{3.7}$$

In all the applications considered in this thesis, evaluations of the likelihood for some sampled values of the model parameters imply running the CFD solver. As a consequence, the posterior cannot be computed analytically, and a numerical method is used to draw samples that are used to approximate the posterior statistics. The samplers used for the present calibrations are presented in section 3.2.

3.1.3 Prediction of the Quantity of Interest

After calibration, the next step is to use the posterior distributions of the parameters to predict the *posterior predictive distribution* of the quantity of interest, Δ^*. This involves computing the

following multidimensional integral, as described in [21]:

$$p(\Delta^*|\Delta_{obs}) = \int p(\Delta^*|\boldsymbol{\theta}, \Delta_{obs}) p(\boldsymbol{\theta}|\Delta_{obs})\, d\boldsymbol{\theta} = \int p(\Delta^*|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\Delta_{obs})\, d\boldsymbol{\theta}. \tag{3.8}$$

This equation assumes that the observed data Δ_{obs} and the quantity of interest Δ^* are conditionally independent when given the parameter vector $\boldsymbol{\theta}$.

A key part of this approach is determining the conditional distribution $p(\Delta^*|\boldsymbol{\theta})$, which requires careful consideration. If the predicted quantity Δ^* is of the same type as the observed data Δ_{obs} (e.g., both are pressures), it is natural to use the likelihood $p(\Delta_{obs}|\boldsymbol{\theta})$ for predictions, as it inherently includes observation errors that affect the predictions.

However, when Δ^* and Δ_{obs} are of different types (e.g., predicting velocity magnitude based on observed pressure), using the likelihood directly is not straightforward. For instance, observation noise in pressure cannot be used in the same way to predict velocity magnitude. In such cases, a new representation of $p(\Delta^*|\boldsymbol{\theta})$ needs to be derived.

Due to limited information, a common simplification is to assume:

$$p(\Delta^* \mid \boldsymbol{\theta}) = \delta(\Delta^* - \Delta(\mathbf{x}, \boldsymbol{\theta})), \tag{3.9}$$

where δ is the Dirac delta function. This approach implies that Δ^* is exactly determined by the model $\Delta(\mathbf{x}, \boldsymbol{\theta})$, without accounting for additional uncertainties beyond the parameter uncertainty captured in $p(\boldsymbol{\theta}|\Delta_{obs})$. While this simplifies the computation of (3.8), it neglects uncertainties arising from factors such as observation noise or model errors.

In this work, similar simplifications are occasionally used when computational difficulties arise.

3.2 Sampling from the Posterior Distribution

This section is based on the book *Bayesian Data Analysis* [46] and Hoffman and Gelman's work [50].

3.2.1 Markov Chain Monte Carlo sampling

Markov Chain Monte Carlo (MCMC) is a method that approximates the posterior distribution by successively drawing samples of $\boldsymbol{\theta}$ from approximate distributions.

The successive draws constitute a Markov chain whose stationary distribution is the posterior

distribution that needs to be sampled. The process must be repeated for long enough to reach the stationary distribution. It is then necessary to introduce stopping criteria that account for the convergence of the simulation.

Different implementations of MCMC sampling exist in the literature and some are presented in the following paragraphs. This study will use the No-U-Turn (NUTS) sampler implemented in the open-source library PyMC4 [50], which was used for all of the present calibrations.

3.2.2 The Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is an adaptation of a random walk. An acceptance/rejection rule is implemented to converge to the specified target distribution. The algorithm proceeds as follows:

1. Draw the starting point $\boldsymbol{\theta}^0$, for which $p\left(\boldsymbol{\theta}^0 \mid \Delta_{obs}\right) > 0$, from a starting distribution $p_0(\boldsymbol{\theta})$.
2. For $t = 1, 2, \ldots$:
   a) Sample a proposal $\boldsymbol{\theta}^*$ from a jumping distribution at iteration t, $J_t\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{t-1}\right)$.
   b) Calculate the ratio,
   $$r = \frac{p\left(\boldsymbol{\theta}^* \mid \Delta_{obs}\right) J_t\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{t-1}\right)}{p\left(\boldsymbol{\theta}^{t-1} \mid \Delta_{obs}\right) J_t\left(\boldsymbol{\theta}^{t-1} \mid \boldsymbol{\theta}^*\right)}$$
   c) Set
   $$\boldsymbol{\theta}^t = \begin{cases} \boldsymbol{\theta}^* & \text{with probability, } \min(r, 1) \\ \boldsymbol{\theta}^{t-1} & \text{otherwise.} \end{cases}$$

The ideal Metropolis-Hastings jumping rule is to sample θ^* from the target distribution, then the ratio r is always exactly 1, and the iterates $\boldsymbol{\theta}_t$ are a sequence of independent draws from the target distribution. As this is not generally possible, a good jumping distribution should have the following properties:

— it is easy to sample

— it is easy to compute the ratio r

— the jumps are not rejected too frequently

— each jump goes a reasonable distance in the parameter space

3.2.3 Hamiltonian Monte Carlo

The random walk behaviour inherent to the Metropolis-Hastings algorithm can be responsible for inefficient convergence of the posterior distribution. The sampling can take a long time zigzagging in the parameter space while moving towards the target distribution. More efficient jumping rules can improve the situation, but for complicated models, this local random walk behaviour remains, especially for high-dimensional distributions.

Hamiltonian Monte Carlo (HMC) suppresses the local random walk behaviour in the Metropolis algorithm and uses information from the gradient of the logarithmic posterior density, allowing it to move much more rapidly to the target distribution. In practice, the gradient must be computed analytically to be computationally effective.

For each component $\boldsymbol{\theta}_j$ in the target space, HMC adds a 'momentum' variable ϕ_j. Both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are updated together in a new Metropolis algorithm, in which the jumping distribution for $\boldsymbol{\theta}$ is determined largely by $\boldsymbol{\phi}$. As mentioned earlier, each iteration of HMC contains three main steps:

1. Update $\boldsymbol{\phi}$ with a draw from its posterior distribution, which is initially the same as its prior distribution.
2. The main step of the HMC iteration is the simultaneous update of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, conducted via a discrete mimicking of physical dynamics. This update involves L 'leapfrog steps', each scaled by a factor ϵ. The leapfrog step is defined as follows:
   a) Use the gradient of the log-posterior density of $\boldsymbol{\theta}$ to make a half-step of $\boldsymbol{\phi}$:

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \frac{1}{2}\epsilon \frac{\partial \left(\log \left(p(\boldsymbol{\theta} \mid \Delta_{obs}\right)\right)}{\partial \boldsymbol{\theta}}$$

   b) Use the momentum vector $\boldsymbol{\phi}$ to update the vector $\boldsymbol{\theta}$:

$$\theta \leftarrow \boldsymbol{\theta} + \epsilon M^{-1} \boldsymbol{\phi}$$

   c) Again use the gradient of $\boldsymbol{\theta}$ to half-update $\boldsymbol{\phi}$:

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \frac{1}{2}\epsilon \frac{\partial \left(\log \left(p(\boldsymbol{\theta} \mid \Delta_{obs}\right)\right)}{\partial \boldsymbol{\theta}}$$

3. Name $\boldsymbol{\theta}^{t-1}$, $\boldsymbol{\phi}^{t-1}$ the value of the parameter and momentum vectors at the start of the leapfrog steps and $\boldsymbol{\theta}^*$, $\boldsymbol{\phi}^*$ the value after the L steps. In the accept-reject step, compute:

$$r = \frac{p\left(\boldsymbol{\theta}^* \mid \Delta_{obs}\right) p(\boldsymbol{\phi}^*)}{p\left(\boldsymbol{\theta}^{t-1} \mid \Delta_{obs}\right) p(\boldsymbol{\phi}^{t-1})}$$

4. Set:

$$\boldsymbol{\theta}^t = \begin{cases} \boldsymbol{\theta}^* & \text{with probability } \min(r, 1) \\ \boldsymbol{\theta}^{t-1} & \text{otherwise.} \end{cases}$$

Since $\boldsymbol{\phi}$ is only an auxiliary variable, there is no need to keep track of it after the accept/reject step.

The two main tuning parameters of HMC are the scaling factor ϵ and the number of leapfrog steps L. If ϵ is too large, the simulation will be inaccurate with low acceptance rates. If taken too small, many small steps will waste computation time. If L is too small, successive samples will be too close, resulting in an undesirable random walk. If too large, HMC trajectories will loop back and retrace their steps.

For many HMC problems, the tuning parameters should vary as the algorithm moves through the posterior distribution, with the matrix M scaling to the curvature of the log density, the step size ϵ getting smaller when the curvature is high, and the number of steps L being large enough for the trajectory to move far through the posterior distribution. However, adaptation performed during sampling can cause the algorithm to converge to the wrong distribution. To that end, researchers have developed algorithms such as the No-U-Turn Sampler (NUTS) that can adapt and converge to the target distribution. The NUTS sampler implemented in PyMC4 eliminates the need to pre-specify the number of leapfrog steps L and the scaling factor ϵ. For this reason, NUTS will be the preferred algorithm for the following applications.

3.2.4 Assessing Convergence of the MCMC Sampling

As mentioned earlier, to get an accurate estimate of the posterior distribution, the sampling must be long enough to draw from the stationary distribution of the Markov chain. Hence, the convergence of the samples needs to be monitored.

Convergence is diagnosed by checking stationarity and mixing. That is to say, two different chains must individually reach stationarity but also trace out a joint distribution. An easy way to test this is to split each chain in half and check if all the resulting chains are mixing. This tests

stationarity because if the chains are stationary, their first and second halves are also stationary.

Mixing can be assessed using between- and within-chains variances B and W of scalar estimands (i.e., the parameters or any other scalar quantity of interest). Using B and W, the marginal posterior variance of the estimand $\boldsymbol{\theta}$ can be estimated by

$$\widehat{\mathrm{var}}^{+}(\boldsymbol{\theta} \mid \Delta_{obs}) = \frac{n-1}{n}W + \frac{1}{n}B \tag{3.10}$$

with n the length of the different chains. Convergence is monitored by estimating the factor by which the scale of the current distribution for ψ might be reduced if the simulations were continued in the limit $n \to \infty$. This potential scale reduction is estimated by:

$$\hat{R} = \sqrt{\frac{\widehat{\mathrm{var}}^{+}(\boldsymbol{\theta} \mid \Delta_{obs})}{W}} \tag{3.11}$$

and it tends to 1 as $n \to \infty$. If the potential scale reduction is high, further simulations may improve inference about the target distribution of the estimand.

Once the chains have mixed, an 'effective number of independent simulation draws' for any estimand of interest can be estimated. In the presence of correlation, this 'effective sample size' can be defined as:

$$n_{\mathrm{eff}} = \frac{n\,m}{1 + 2\sum_{t=1}^{T}\rho_t} \tag{3.12}$$

where n is the number of samples in the m different chains, ρ_t the auto-correlation of the sequence at draw t, and T is the first odd positive integer for which $\rho_{T+1} + \rho_{T+2}$ is negative.

As a guideline, Gelman [47] suggest that the sampling has converged to the posterior distribution when $\hat{R}$ value is less than 1.1 and n_{eff} is more than 10 per chain for all parameters. The $m\,n$ samples can then be treated as a sample from the target distribution.

3.3 Model aggregation methods

3.3.1 Model selection and model plausibilities

When many models are available to predict a QoI, the question of which model to choose arises. The different types of models can be for example the same RANS model with different stochastic modelling (model-form and/or experimental uncertainty models), different RANS

models with the same stochastic model or different RANS models with different stochastic models [77].

To select the model that fits best the data, Raftery *et al.* [54] suggests using Bayes Factors. Considering two models M_1 and M_2, model plausibility is defined, according to Bayes's law, as :

$$p(M_1|\Delta_{obs}) = \frac{p(\Delta_{obs}|M_1)p(M_1)}{p(\Delta_{obs}|M_1)p(M_1) + p(\Delta_{obs}|M_2)p(M_2)} \tag{3.13}$$

where $p(\Delta_{obs}|M_m) = \int p(\Delta_{obs}|\boldsymbol{\theta}, M_m)p(\boldsymbol{\theta}|M_m)d\boldsymbol{\theta}$ is referred to as the marginal likelihood.

Raftery *et al.* suggest that the extent to which the data support M_1 over M_2 is measured in terms of posterior odds :

$$\frac{p(M_1|\Delta_{obs})}{p(M_2|\Delta_{obs})} = \frac{p(\Delta_{obs}|M_1)}{p(\Delta_{obs}|M_2)}\frac{p(M_1)}{p(M_2)} = B_{12}\frac{p(M_1)}{p(M_2)} \tag{3.14}$$

where B_{12} is called the Bayes factor between the two models. These authors also suggest the following rule of thumb to interpret the Bayes Factor: when $1 \leq B_{12} \leq 3$ there is weak evidence that the data favours M_1, when $3 \leq B_{12} \leq 20$ the evidence is positive, strong if $20 \leq B_{12} \leq 150$ and decisive if $B_{12} \geq 150$.

When confronted with more than two models, Yu *et al.* [112] suggest extending the Bayes factor to the geometric mean. For N_M models $M_1, ...M_{N_M}$,

$$b_i = \left(\prod_{j=1}^{N_M} B_{ij}\right)^{1/N_M} \tag{3.15}$$

Note that when all models are *a priori* equally likely, model selection can be based on model plausibility alone.

3.3.2 Bayesian Model Averaging

When predicting physical phenomena such as turbulence, it is often difficult to choose a single model from a set of competing models M_m, where $1 \leq m \leq N_M$, especially when limited data are available. Different models may be better suited to predicting specific physical phenomena or regions of interest, and no universal model can be a priori trusted to perform optimally in

every situation. Therefore, it is reasonable to assume that each model captures a portion of the available information, necessitating predictions that incorporate multiple model outputs. Bayesian Model Averaging (BMA) offers a coherent framework for combining model predictive inferences within a Bayesian context.

Consider a set $\mathcal{M} = \{M_1, \ldots, M_m, \ldots, M_{N_M}\}$ of N_M models, each characterised by a vector of parameters $\boldsymbol{\theta}_m$. If Δ^* is the unobserved quantity of interest, then its posterior distribution given the observed data Δ_{obs} is, in the BMA framework [27, 69, 49, 17] :

$$p(\Delta^* \mid \Delta_{obs}) = \sum_{m=1}^{N_m} p(M_m \mid \Delta_{obs}) p(\Delta^* \mid M_m, \Delta_{obs}) \tag{3.16}$$

The BMA formulation is an average of the posterior distributions under each of the models considered, weighted by their posterior model probability. In equation 3.16, N_M are considered. For model $M_m, m \in [0, N_M]$, the model posterior plausibility $p(M_m \mid \Delta_{obs})$ is given by :

$$p(M_m \mid \Delta_{obs}) = \frac{p(\Delta_{obs} \mid M_m) p(M_m)}{\sum_{k=1}^{N_m} p(\Delta_{obs} \mid M_k) p(M_k)} \tag{3.17}$$

in which $p(M_m)$ is the prior probability that M_m is the true model (given that one of the models considered is true), and

$$p(\Delta_{obs} \mid M_m) = \int p(\Delta_{obs} \mid \boldsymbol{\theta}_m, M_m) p(\boldsymbol{\theta}_m \mid M_m) d\boldsymbol{\theta}_m \tag{3.18}$$

is the marginal likelihood of model M_m, $p(\boldsymbol{\theta}_m \mid M_m)$ is the prior density of $\boldsymbol{\theta}_m$ under model M_m and $p(\Delta_{obs} \mid \boldsymbol{\theta}_m, M_m)$ is the likelihood. Note that, all probabilities are implicitly conditional on the set of all models being considered.

The posterior mean and variance of Δ^* are as follows :

$$\mathrm{E}[\Delta^* \mid \Delta_{obs}] = \sum_{m=1}^{N_M} \mathrm{E}\left[\Delta^* \mid \Delta_{obs}, M_m\right] p\left(M_m \mid \Delta_{obs}\right) \tag{3.19}$$

$$\begin{aligned}\mathrm{Var}[\Delta^* \mid \Delta_{obs}] = &\sum_{m=1}^{N_M} \mathrm{Var}\left[\Delta^* \mid \Delta_{obs}, M_m\right] p\left(M_m \mid \Delta_{obs}\right) \\ &+ \sum_{m=1}^{N_M} p\left(M_m \mid \Delta_{obs}\right) \left(\mathrm{E}\left[\Delta^* \mid \Delta_{obs}, M_m\right] - \mathrm{E}[\Delta^* \mid \Delta_{obs}]\right)^2\end{aligned} \tag{3.20}$$

where $\mathrm{E}\left[\Delta^* \mid \Delta_{obs}, M_m\right]$ and $\mathrm{Var}\left[\Delta^* \mid \Delta_{obs}, M_m\right]$ are the mean and the variance of the posterior predictive distribution of Δ^* using model M_m. The variance is decomposed into two contributions: the first term on the right-hand side of equation 3.20 represents the variance induced by uncertainty in the model parameters, while the second term accounts for the variance arising from discrepancies in the predictions made by the competing models in the model set. Within the BMA framework, the former is referred to as within-model variance, and the latter as between-model variance.

Bayesian Model Averaging can be seen as the application of Bayes' rule to a "hyper-model" that consists of a distribution over models and a distribution over the parameters within each model [113]. It generates predictions by taking a weighted average of multiple models, similar to other model mixing methods. However, BMA uniquely employs the posterior model probability of each model as its weighting factor. These posterior probabilities, derived from the likelihoods of the models based on the training data, offer a direct measure of the model's predictive accuracy. Consequently, BMA favours the most accurate models for the calibration scenario, while incorporating the uncertainty associated with model selection. BMA has been shown to deliver better predictive performance and more consistent results than selecting a single model [113].

The determination of the prior model probability has been acknowledged as a challenge [49]. Hoeting et al. suggest that, in the absence of substantial information regarding the relative plausibility of the models, it is reasonable to assume that all models are equally likely. Adopting this approach, several authors [35, 73, 20] have employed uninformative uniform priors for the models.

To determine the posterior predictive distribution of any quantity of interest, it is essential to propagate the posterior distributions obtained from calibrations through the model. This step necessitates the use of an Uncertainty Quantification (UQ) method. Typically, these methods rely on sampling strategies that require an impractically large number of RANS computations,

even with the most efficient sampling techniques. Uncertainty propagation to full-flow fields has been mostly performed for relatively low-dimensional problems using intrusive or non-intrusive polynomial chaos methods [75, 81] and their sparse variants (Compressed Sensing, Sparse Bayesian Learning...) [87, 90] or through surrogate modelling [100, 68]. Some improvement is obtained by combining dimensionality reduction techniques, as in [67], and a similar approach is used later in this work.

However, surrogate modelling of full fields remains a challenging and costly task for moderately to high-dimensional problems. The advent of advanced machine learning algorithms, such as Deep Neural Networks, is anticipated to soon enable the creation of surrogate models for a comprehensive CFD solution, although this remains an active area of research [100, 68].

A numerically convenient approach consists of approximating the full posterior distributions as Dirac distributions centred on their Maximum A Posteriori (MAP) value, as proposed in [36]. While this approximation neglects the posterior parametric uncertainty, it substantially reduces the computational cost to that of a single deterministic CFD computation. With this method, a BMA prediction requires only as many computations as there are models in the model set, rather than the vast number of calculations typically required for Monte Carlo UQ. Furthermore, since the parameters are not treated as random variables during the propagation step, the BMA formula can be applied to any quantity of interest, whether or not a surrogate model has been previously constructed. This approach ultimately facilitates the reconstruction of the full stochastic solution associated with the ensemble of concurrent models.

3.3.3 Bayesian Model-Scenario Averaging

Predicting a new scenario using a single model calibrated for a different scenario introduces bias due to model inadequacy. Bayesian Model Averaging (BMA), discussed in Section 3.3.2, mitigates bias from the model structure by averaging predictions across multiple models. However, these models are typically calibrated for specific scenarios, which may differ significantly from the scenario being predicted. While BMA reduces bias from model choice, it may still be biased if the calibration scenario is inadequate, leading to substantial errors when applied to a scenario too different from the calibration one. One approach to mitigate this bias is to simultaneously calibrate models across multiple scenarios, as proposed in [17]. However, this method tends to select coefficients that provide the best overall fit across scenarios, without necessarily yielding highly accurate predictions for individual scenarios. This issue has been explored in [73] in

the context of compressible flow predictions using real-gas equations of state, where Bayesian Model-Scenario Averaging (BMSA), which calibrates models individually across multiple scenarios, outperforms BMA in terms of predictive accuracy.

Building upon Draper's work, Meyer *et al.* [27, 74] develop the framework of Bayesian Model-Scenario Averaging as follows. Considering N_M models and N_S calibration scenarios $\mathcal{S} = \{S_1, \ldots, S_{N_S}\}$, each with some available calibration data $\Delta_{obs,s}$, the prediction of the quantity of interest Δ^* for an unseen scenario S^* is the weighted mean of BMSA predictions across all calibration scenarios:

$$\begin{aligned} p(\Delta^*|\Delta_{obs}, S^*) &= \sum_{s=1}^{N_S} p(\Delta^*|S_s, \Delta_{obs,s}, S^*) p(S_s) \\ &= \sum_{s=1}^{N_S} \sum_{m=1}^{N_M} p(\Delta^*|M_m, S_s, \Delta_{obs,s}, S^*) p(M_m|S_s, \Delta_{obs,s}) p(S_s) \end{aligned} \tag{3.21}$$

where Δ_{obs} is the collection of all scenario-dependent observations $\Delta_{obs,s}$. In the averaging, the scenarios in $\mathcal{S}$ are assumed to be mutually exclusive and collectively exhaustive, ensuring their probabilities sum to one. This is of course an approximate view, the set of possible scenarios being infinite. The posterior mean and variance are computed from the posterior distribution as:

$$\mathrm{E}[\Delta^*|\Delta_{obs}, S^*] = \sum_{s=1}^{N_S} \sum_{m=1}^{N_M} \mathrm{E}[\Delta^*|M_m, S_s, \Delta_{obs,s}, S^*] p(M_m|S_s, \Delta_{obs,s}) p(S_s) \tag{3.22}$$

$$\begin{aligned} \mathrm{Var}[\Delta^*|\Delta_{obs}, S^*] = &\sum_{s=1}^{N_S} \sum_{m=1}^{N_M} \mathrm{Var}[\Delta^*|M_m, S_s, \Delta_{obs,s}, S^*] p(M_m|S_i, \Delta_{obs,s}) p(S_s) \\ &+ \sum_{s=1}^{N_S} \sum_{m=1}^{N_M} \left(\mathrm{E}[\Delta^*|M_m, S_s, \Delta_{obs,s}, S^*] - \mathrm{E}[\Delta^*|\mathcal{M}, S_s, \Delta_{obs}, S^*]\right)^2 p(M_m|S_s, \Delta_{obs,s}) p(S_s) \\ &+ \sum_{i=1}^{N_S} \left(\mathrm{E}[\Delta^*|\mathcal{M}, S_s, \Delta_{obs}, S^*] - \mathrm{E}[\Delta^*|\Delta_{obs}, S^*]\right)^2 p(S_s) \end{aligned} \tag{3.23}$$

In equation 3.23, the variance is decomposed into three components:

1. The first component arises from uncertainty in model parameters, reflecting the propagation of the posterior pdf $\boldsymbol{\theta}_{m,s}$ from model M_m calibrated for scenario S_s. This term is termed

the within-model, within-scenario variance.

2. The second term represents the between-model, within-scenario variance, which accounts for discrepancies in predictions among competing models in the model set $\mathcal{M}$, all calibrated for the same scenario S_s and applied to S^*.
3. The third term, known as the between-scenario variance, accounts for differences in the posterior distributions $\boldsymbol{\theta}_{m,s}$ and model probabilities $p(M_m \mid S_s, \Delta_{obs,s})$ when different calibration scenarios S_s are used. This variation leads to different predictions for the new scenario S^*.

BMSA needs *a priori* specification of the scenario probabilities $p(S_s)$, whose role is to assign a higher probability to calibration scenarios that are more likely to provide an accurate estimate of model coefficients with respect to the new predicted scenario S^*. Various choices are possible, including Bayesian criteria such as Bayesian Information Criterion (BIC) or Akaike criterion (see [47]). Such criteria use no information about the prediction scenario. Ref. [36] proposed an empirical criterion based on the agreement among models. The rationale behind it is that if S^* is close enough to one of the calibration scenarios and if the models are well calibrated for such a scenario, then the set of coefficients inferred from the data available for that scenario is also the best suited to predict the closest one. Note that the criterion may fail in cases for which all models strongly agree with the wrong solution. Additionally, such a criterion requires propagating all models through the new case before scenario weights can be determined. For that reason, alternative empirical criteria (namely, based on the a posteriori model uncertainty for the calibration scenarios or an empirical distance measure between scenarios) were also tested in [22] for a compressor cascade flow problem. Overall, the study showed that, BMSA predictions were not too sensitive to the chosen criterion, and that the hierarchy of scenarios and models was similar for all considered criteria.

3.3.4 Clustered Bayesian Model Averaging

Both BMA and BMSA use constant weights for any quantity of interest to be predicted at any space location in the flow. This contrasts with the expert belief that some RANS models are more accurate at capturing some flow processes, and that model hierarchy may vary according to the specific features of the flow at stake. In statistical terms, this comes to say that model probabilities are expected to vary across the computational domain. However, BMA or BMSA model weights are determined once and for all from the model evidence estimated for each

calibration scenario, which measures how good the model is at capturing the data for that scenario. In the following, we present some methodologies that can relax the constraint of constant model weights.

Clustered Bayesian Model Averaging (CBA), proposed by Yu et al. [113], extends standard BMA by accommodating varying model behaviours across different regions of the covariate space. In CBA, the covariate space is partitioned into regions where model performances are similar within each region but differ between regions, leading to models being assigned different weights across these regions. The covariate space indicates the space associated with the model explanatory variable, i.e. the variables that are considered as known. For instance, in CFD problems we may want to predict a flow field as a function of space so that the covariate space is the 3D geometrical space. We may also imagine a dependency on other variables, e.g. associated with the geometry of the domain or with the boundary conditions.

To account for local differences in model performance and subsequently adjust model weights, Yu et al. introduce the concept of local Bayes factor.

Definition: *Consider two models M_1 and M_2, and suppose that the covariate space Λ can be partitioned into K disjoint partitions such that $\Lambda = \Lambda_1 \cup \ldots \cup \Lambda_K$. The local Bayes factor associated with the subspace Λ_k for model M_1 over M_2 is defined as the ratio of their marginal likelihoods calculated within Λ_k:*

$$B_{k12} = \frac{p(\Delta_{obs,k} \mid M_1)}{p(\Delta_{obs,k} \mid M_2)} \tag{3.24}$$

To avoid solely local model updates which may lead to instability (overfitting), the marginal likelihood is updated through partial information, where a subset of data $\Delta^*_{obs} \in \Delta_{obs}$, not necessarily all belonging to Λ_k, is used for model updating but not included in the local Bayes factor computation. Therefore, $p(\Delta_{obs,k} \mid M_m)$ is replaced by $p(\Delta_{obs,k} \mid M_m, \Delta^*_{obs}) = \int p(\Delta_{obs,k} \mid M_m, \Delta^*_{obs}, \boldsymbol{\theta}_m) p(\boldsymbol{\theta}_m \mid M_m, \Delta^*_{obs})\, d\boldsymbol{\theta}_m$. Local updating addresses two potential issues: it stabilises the likelihood surface, ensuring more reliable predictions, and it mitigates the reduction in information within each region (the smaller the region, the lesser data are contained in it), thereby maintaining the discriminative power of the Bayes factor between competing models. Observations from different regions of the covariate space are supposed to

be independent. Then,

$$p(\Delta_{obs}|M_m) = \prod_{k=1}^{K} p(\Delta_{obs,k}|M_m) \tag{3.25}$$

This hypothesis allows computing the global Bayes factor using the local Bayes factor in each subset Λ_k:

$$B_{12} = \frac{p(\Delta_{obs}|M_1)}{p(\Delta_{obs}|M_2)} = \prod_{k=1}^{K} \frac{p(\Delta_{obs,k}|M_1)}{p(\Delta_{obs,k}|M_2)} = \prod_{k=1}^{K} B_{k12} \tag{3.26}$$

While in practice observations from different regions are not independent, they are often conditionally independent given the model parameters. The authors suggest that "under a concentrated prior distribution, the dependence among observations is weak and, in the limiting case of a prior that concentrates at a given parameter vector, conditional independence is independence."

The local Bayes factors are then used to measure relative model performance and to assign different weights to models based on their performance in different regions.

In practice, the CBA method involves partitioning the covariate space using clustering techniques like Classification And Regression Trees (CART) to identify regions where models have similar performances. In each of these regions, the models are averaged using local Bayes factors. In the basic CMA version, the available data are randomly split into two subsets, of which one is used to update the model parameters, and the other to compute the Bayes factor and generate clusters of homogeneous performance. However, several variants to the basic CBA algorithm are also proposed, depending on which data are used for which task (model updating or weighting and clustering) to improve the robustness and computational efficiency of the method. Additionally, the approach can be extended to any number of concurrent models. The reader is referred to the original paper [113] for more details.

The above-mentioned features make CBA a flexible and powerful tool for scenarios where model performance is not uniform across all data, improving predictive accuracy compared to traditional methods for statistical problems where inexpensive models are involved, hence model updating can be performed without a too large computational effort. Unfortunately, the original formulation of CBA is not easily applicable to costly computer models such as those encountered in CFD, due to the need for repeated model updates and a large amount of data. For that reason, taking inspiration from the CBA algorithm as well as from non-Bayesian model aggregation techniques such as those discussed in [25], De Zordo-Banliat et al. [19] proposed a model aggregation technique that does not require model updates. Such methodology constitutes the

starting point of the present work, and it is described in the following Section.

3.3.5 Space-dependent Model Aggregation

In his PhD work, De Zordo-Banliat [19] proposed a space-dependent Model Aggregation (XMA) algorithm to generate mixtures of different RANS model solutions. Unlike the BMA or BMSA approaches presented in the preceding section and used in the previous works [35, 36, 20], XMA enables locally variable model weights. The algorithm is inspired by both CBA and BMA, yet redesigned for computationally expensive CFD applications. More precisely, the CBA algorithm is simplified to account for the high computational cost of CFD runs, and no model calibration is performed, meaning that local Bayes factors are no longer available. As a consequence, in the XMA methodology, a loss function is used instead to compute local model weights, partly inspired by that used in [25]. The latter can be viewed as a point-wise likelihood, given the model used (with a fixed set of parameters). To allow model generalisation to flows different from those considered for training the model weights, the cost function, which depends on the space location where the data is observed, is reformulated as a function of a set of well-chosen flow features, instead of the local geometrical coordinates.

Considering N_M RANS models, the prediction of the QoI is written as the weighted average of the individual model predictions:

$$\Delta_{XMA} = \sum_{m=1}^{N_M} w_m \Delta_m \tag{3.27}$$

where Δ_m is the prediction of the QoI using model M_m, and $w_m = w_m(\mathbf{x})$ is a space-dependent weighting function associated with the m-th model, subject to:

$$0 \leq w_m(\mathbf{x}) \leq 1, \quad \text{and} \quad \sum_{m=1}^{N_M} w_m(\mathbf{x}) = 1 \quad \forall \mathbf{x} \tag{3.28}$$

Spatial coordinates are specific to a given flow configuration and do not possess the due invariance properties for ensuring model generalisation. For that reason, the spatial fields are instead transformed into a well-chosen space of features $\boldsymbol{\eta} = \boldsymbol{\eta}_m(\mathbf{x})$, i.e. flow properties representative of the local flow physics.

In this methodology, the weighting function from the Exponentially Weighted Average, initially

introduced by Deswarte *et al.* [25], is used:

$$w_m\left(\boldsymbol{\eta}_m, \sigma\right) = \frac{g_m\left(\boldsymbol{\eta}_m, \Delta_{obs}, \sigma\right)}{\sum_{j=1}^{N_M} g_j\left(\boldsymbol{\eta}_j, \Delta_{obs}, \sigma\right)}, \quad m \in \{1, \ldots, N_M\} \tag{3.29}$$

where g_m is a cost function defined by

$$g_m = \exp\left(-\frac{1}{2}\frac{\|\Delta_m\left(\boldsymbol{\eta}_m\right) - \Delta_{obs}\|^2}{\sigma^2}\right) \tag{3.30}$$

and the data Δ_{obs} are now used only to inform the model weights.

The function g_m is reminiscent of a Gaussian likelihood function used in Bayesian approaches, which amplifies or damps the discrepancies between the output of the m-th model and the data. The cost function equals 1 when the model perfectly matches the data and it tends to 0 for large discrepancies. The squared exponential ensures a smooth variation of g_m with $\left(\Delta_m(\boldsymbol{\eta}_m) - \Delta\right)$. The parameter σ can be seen as a learning rate, controlling how fast the departure of the model prediction Δ_m from the observed data Δ is penalised by the cost function g_m. When $\sigma \to \infty$, the models are assigned uniform weights, when $\sigma \to 0$, the worst-performing models get weights closer to 0, while the weight of the best-performing model gets closer to 1. In other terms, σ governs model selection. In [25], a grid search procedure is used to determine the optimal value of the parameter σ. The cost functions are evaluated using high-fidelity data and then interpolated in the feature space using a regressor model. In [19], a Random Forest regressor (RFR) was chosen, because of its flexibility and robustness. A similar approach will be adopted also in the present study.

In practice, we may wish to use XMA for predicting different QoI than the observed vector Δ_{obs}, either by direct extraction from the model or through post-processing of the solution. Examples of such QoI are given by flow properties at given locations in the flow, like velocity profiles, pressure or skin friction distributions. We denote $\Delta^* = (\Delta_1^*, \ldots, \Delta_N^*)^T$ the vector of such QoI. Since no data is available to inform the weights for Δ^*, we use the same weights as those learned for Δ :

$$\Delta^*(\mathbf{x}^*) = \sum_{m=1}^{N_M} w_m[\boldsymbol{\eta}^*(\mathbf{x}^*)]\Delta_m^*(\mathbf{x}^*) \tag{3.31}$$

Of note, in the preceding expressions, the weighting function of the m-th model is expressed as a function of the features $\boldsymbol{\eta}^*$ computed according to the same model at the prediction location $\mathbf{x}^*$. The idea is that, based on the local flow features predicted by model m and the local model performance measured by the cost function in equation (3.30), the model is assigned a weighted representative of the likelihood of the model to capture the data when the flow exhibits certain characteristics.

Besides, we estimate the predictive variance as :

$$\mathrm{Var}[\Delta] = \sum_{m=1}^{N_M} w_m \Big(\Delta_m - \mathrm{E}[\Delta_m]\Big)^2 \tag{3.32}$$

In the above, Var must be understood as an indicator of the consensus among the models: small variances result from a strong agreement of individual model predictions while large variances reveal a divergence. Furthermore, as the weights are better informed, they get closer to 1 for the best models and to 0 for the worst models, and the variance decreases. This indicates that the uncertainty about model choice has been reduced.

While the XMA approach has been successfully demonstrated in [19] for the flow through a compressor cascade (a test case that is also considered later in this manuscript), by aggregating four on-the-shelf RANS models from the literature with their nominal closure coefficients, one may wonder whether its accuracy can be further improved by also updating the model parameters for the class of flows at stake. Such investigations will make the object of Chapter 5.1.

3.4 Surrogate modelling

Sampling from the posterior distribution of the parameters during the calibration step, as described in the previous sections, requires the use of sampling methods, hence many runs of the CFD solver, even with the more efficient samplers. To that end, surrogate modelling of the CFD solver is mandatory to make calibration of the RANS models computationally tractable. The goal is to model the whole field of the quantity used for calibration. To reduce the computational cost of making high-dimensional surrogate models or many independent surrogate modelling, we employ a surrogate modelling approach based on output dimension reduction methods like

Proper Orthogonal Decomposition to train surrogate model with fewer dimensions using Gaussian Process Regression - Kriging [67, 10, 57].

Proper Orthogonal Decomposition

The original idea behind Proper Orthogonal Decomposition (POD) [61, 7, 16, 9] is to decompose a vector field $u(x,t)$ into a set of deterministic spatial functions $\{\phi_k(x)\}$ modulated by time coefficients $\{a_k(t)\}$ so that:

$$u(x,t) = \sum_{k=1}^{\infty} a_k(t)\phi_k(x)$$

In RANS modelling, the quantity of interest is not time-dependent as the equations are time-averaged. However, this method can be applied here considering that the RANS model output depends on the RANS parameters' values $\boldsymbol{\theta}$. To that end, the idea is to decompose a vector field Δ_{RANS} into a set of deterministic spatial functions $\{\phi_k(x)\}$ modulated by coefficients dependent on the RANS parameter values $\{a_k(\boldsymbol{\theta})\}$ so that:

$$\Delta_{RANS}(x,\boldsymbol{\theta}) = \sum_{k=1}^{\infty} a_k(\boldsymbol{\theta})\phi_k(x) \tag{3.33}$$

Generally, only a few POD modes are enough to capture most of the information so the field can be appreciated by

$$\Delta_{RANS}(x,\boldsymbol{\theta}) \approx \sum_{k=1}^{N_k} a_k(\boldsymbol{\theta})\phi_k(x) \tag{3.34}$$

Only N_k coefficients dependent on the RANS parameter value $\{a_k(\boldsymbol{\theta})\}$ need to be regressed using a Gaussian Process Regressor to predict the quantity of interest.

Gaussian Process Regression - Kriging

Gaussian Process Regression (GPR) is a flexible, non-parametric method used for predicting complex, non-linear functions. It relies on the concept of a Gaussian Process (GP), which is a collection of random variables, any subset of which follows a multivariate Gaussian distribution. In GPR, we model our data $a_k(\boldsymbol{\theta})$ using a GP with a mean function $m(\boldsymbol{\theta})$ and a covariance function $k(\boldsymbol{\theta},\boldsymbol{\theta}')$. Formally, this can be expressed as:

$$a_k(\boldsymbol{\theta}) \sim \mathcal{GP}(m(\boldsymbol{\theta}), k(\boldsymbol{\theta}, \boldsymbol{\theta}')) \tag{3.35}$$

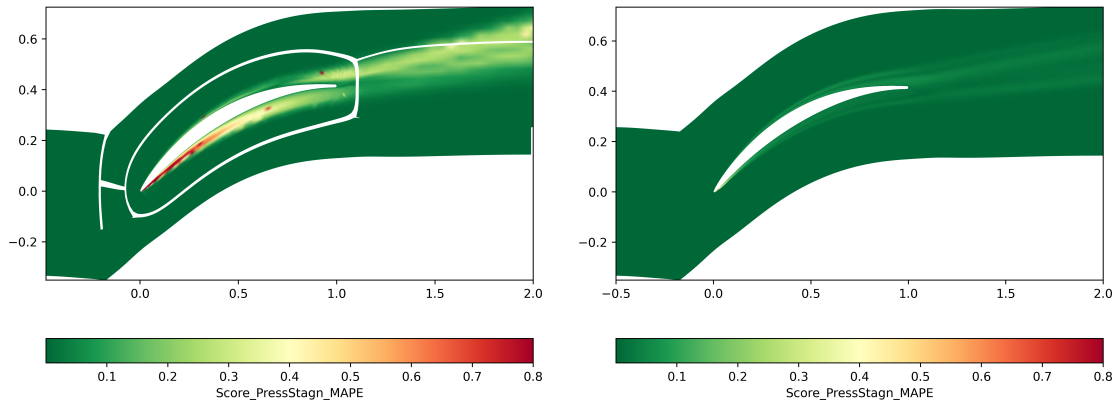
The GP formulation provides predictions in the form of a full predictive distribution. Hence, in a sense it also includes uncertainty estimates, making GPR a powerful tool for regression [12, 104]. The GP formulation is also analytic and differentiable which allow for an analytic and differentiable in $\boldsymbol{\theta}$ reconstruction of the quantity used for calibration $\Delta_{RANS}(x, \boldsymbol{\theta}) \approx \sum_{k=1}^{N_k} a_k(\boldsymbol{\theta})\phi_k(x)$. This significantly eases the implementation of the sampling using *pymc* in our study. [85, 71]

The GP model distributions are defined by the covariance function $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ a priori defined by a kernel function. This kernel function will measure the similarity between points and if points are similar, their respective outputs should be similar too. The selection of the kernel function is then critical, as it impacts the model's generalisability. There is a wealth of kernel available in the *scikit-learn* [80] library used for GPR in this study and the choice of kernel function and hyper-parameter tuning will be made using a grid search, exploring which option allows the best regression of the POD modes.

The comparison of the method is presented for the Wilcox $k-\omega$ model on the NACA65-V103 configuration. Either for the one-point kriging or the POD kriging, the Gaussian process regressor needs data points to interpolate hence several runs of the CFD solver with varying values of the model parameters. 100 RANS calculations are made using a Latin Hypercube Sampling [53] of the parameter space, optimised under the Maximum Projection Design criterion. This criterion ensures an optimal space filling by maximising the minimal distance between points of the hypercube, for every projection in parameter sub-spaces.

Figures 3.1a and 3.1b represent the mean percentage error of the two surrogate models created. The POD kriging method induces a relatively small MAPE compared to the one induced by the One-point kriging method. The dimension reduction seems to act as a low-pass filter on the spatial variability of the error produced by the independent regressions. It also has to be noted that the training using grid search and leave-one-out cross-validation of the One-point kriging method takes multiple hours meanwhile the POD-Kriging method takes only a few minutes as only the 4 most important modes need to be modelled to capture 99% of the variance in every configuration we worked on in the study.

In the rest of this study, the surrogate models used are designed using the POD-kriging technique to model the 4 most important modes and will be trained and validated using grid search to optimise the kernel and associated hyper-parameters and validated via leave-one-out



(a) Independent GPRs

(b) POD+kriging technique

Figure 3.1 – Mean absolute percentage error on the total pressure field obtained different techniques.

cross-validation.

3.5 Chapter summary

In this chapter, we introduce the main statistical methods that will be further investigated in the following chapters. Specifically:

1. As previously done in the works of Edeling [33, 35] and De Zordo-Banliat [20, 22], we use Bayesian inference for updating the closure coefficients of a set of RANS models for a given class of flows, called a set of calibration scenarios;
2. we account for model-form uncertainties using model aggregation methods that use convex linear combinations of a set of competing (alternative) RANS models to predict a new case, alongside a confidence interval estimated as the consensus among the competing models;
3. given that RANS model performance is known to vary according to the physical processes at stake, we consider aggregation methods that assign model weights depending on a set of local model features

The goal is to develop a prediction methodology that provides improved predictions of turbulent flows alongside an estimate of the predictive uncertainty, the following questions are yet to be answered:

1. how much improvement can be obtained by using data for updating the model parameters and how much by training the weighting functions to optimally combine the models?
2. Which data are better for calibrating the models?
3. To which extent the aggregated model can generalise to unseen flows?

In the following of the manuscript, we will try to provide answers to these questions.

4

Flow Configurations

This chapter discusses the different flow configurations analysed in this work. These configurations are described here for clarity, as they are used later in the manuscript. The aim is to describe the key configurations, their operating conditions, and the experimental or simulation data used. This data helps calibrate RANS models, train model aggregates, and validate the results.

Chapter contents

4.1 Building-block Configurations

We consider four "building-block" configurations for calibrating RANS models, as detailed in Part 7. These configurations represent shear flows, which are typically used for RANS model calibrations. Additionally, they include more complex flows (e.g., with separation or 3D effects) but are still dominated by a single physical phenomenon, such as recirculation or corner vortices. These configurations are described in the following subsections.

4.1.1 Zero Pressure Gradient Turbulent Boundary Layer over a Flat Plate

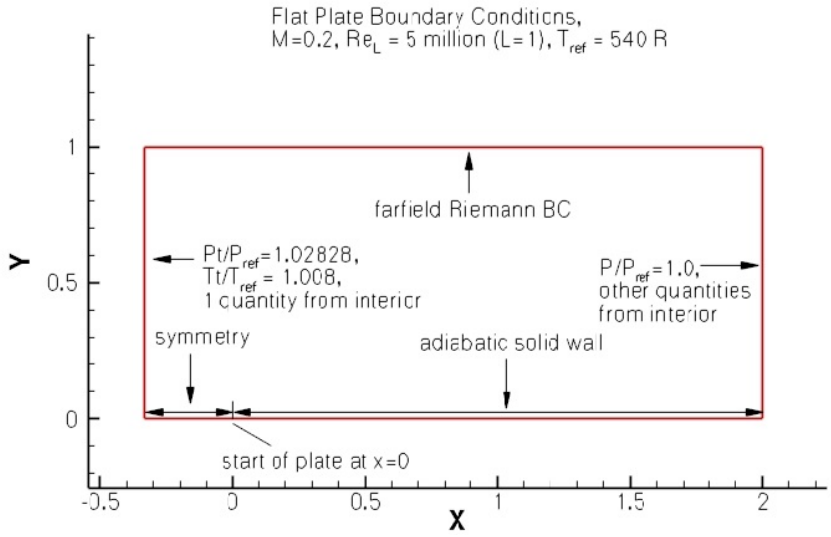


Figure 4.1 – Flat plate configuration and boundary conditions for the turbulent boundary layer with zero pressure gradient.

The turbulent boundary layer over a flat plate under a zero pressure gradient is a fundamental test case for calibrating baseline RANS models. This configuration has been widely studied in the literature and is often used for validating CFD solvers [107, 56]. Bayesian calibration has already been conducted on this configuration and is presented in [33, 21].

The experimental data for calibration includes longitudinal velocity profiles, taken at four

various locations defined by their Reynolds number based on the momentum thickness, respectively at $Re_\theta \in \{2400, 4858, 8172, 11472\}$. These four Re_θ values were chosen from five available options, but the first profile, located at the very beginning of the flat plate, was not used for calibration as it was deemed unsuitable. The friction coefficient was deduced empirically from these velocity measurements along the flat plate. These data were obtained from the study by Wieghardt and Tillman [107] and are accessible through the NASA Glenn Research Center archive. Altogether, the experimental dataset used for calibration amounts to 66 data points.

The simulations were performed using the *elsA* solver. The boundary conditions are illustrated in Figure 4.1. The mesh used was sourced from the NASA Langley Research Center and consists of 273x193 nodes.

4.1.2 Wake Flow Behind a Flat Plate

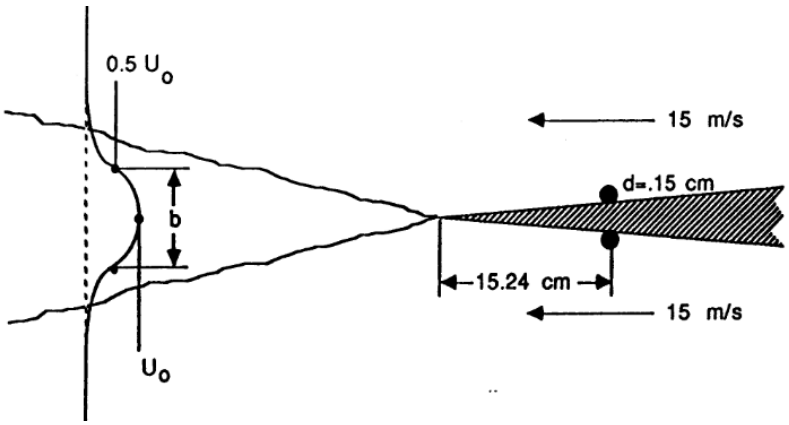


Figure 4.2 – Flat plate wake experiment: Configuration and boundary conditions. Taken from [106]

This configuration studies the self-similar properties of the wake flow behind a flat plate, generated at the trailing edge of a splitter plate in a Mixing Layer Wind Tunnel. The study was conducted by Weyghand and Mehta [106], with measurements taken at various stream-wise locations for a free-stream velocity of 15 m/s, corresponding to a Reynolds number of approximately 1750.

The calibration data includes longitudinal velocity profiles measured at various stream-wise locations. The measurements were taken beyond a streamwise distance equivalent to about 350 wake momentum thicknesses θ, a distance after which the mean flow data indicated

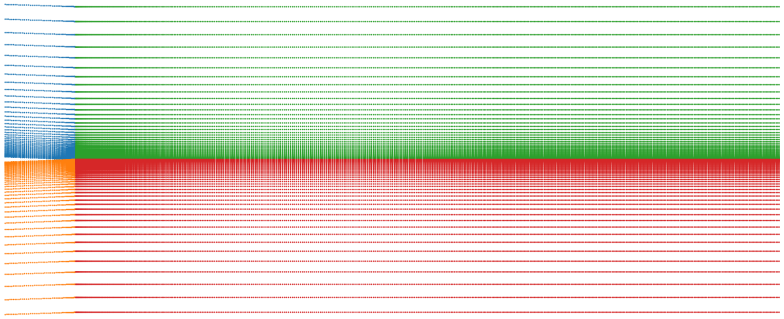


Figure 4.3 – Mesh used for the flat plate wake experiment.

a self-similar behaviour. The streamwise positions used for the measurements were $X/\theta \in \{350, 464, 750, 1093\}$, where X is the streamwise position after the flat plate.

Simulations were carried out using the *elsA* solver with Mach=0.1 and free stream velocity of 15 m/s. Figure 4.3 shows the mesh, which was provided by Camille Matar who used the meshing software Pointwise. It consists of 4 blocks and contains 2 blocks containing 50x100 nodes and 2 blocks containing 500x100 nodes.

4.1.3 Backward Facing Step

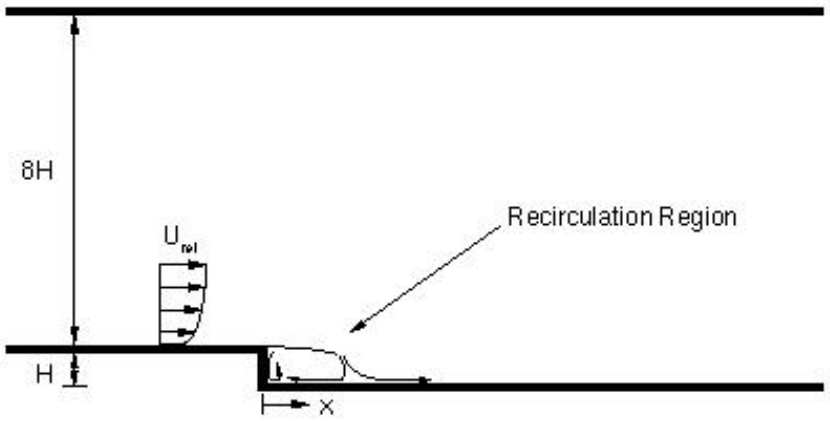


Figure 4.4 – Configuration for the backwards-facing step.

The incompressible flow over a backwards-facing step is a widely tested configuration. The data used here are from Driver and Seegmiller [28]. This is also a test case given in the

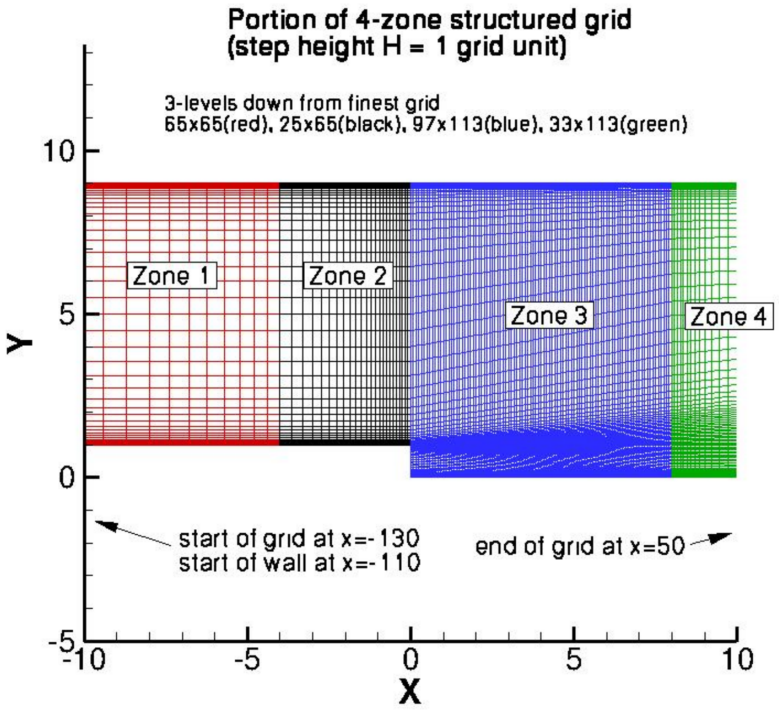


Figure 4.5 – Mesh for the backwards-facing step simulation.

ERCOFTAC Database (Classic Collection) C.30 and has been used in turbulence modelling workshops [39, 38, 40]. A turbulent boundary layer encounters a sudden back step, causing flow separation. The flow reattaches downstream of the step. The Reynolds number based on boundary layer momentum thickness before the step is 5000, corresponding to a Reynolds number of approximately 36,000 based on step height H. The boundary layer thickness before the step is approximately 1.5H. The step height to tunnel exit height ratio is 1:9, designed to minimise freestream pressure gradients, and the step height to tunnel width ratio is 1:12 to reduce three-dimensional effects.

Calibration data includes longitudinal velocity profiles measured at various stream-wise locations within the recirculation zone.

Simulations were performed with the *elsA* solver under incompressible conditions (Mach=0.128), following the boundary conditions shown in Figure 4.5. The mesh was sourced from the NASA Langley Research Center.

4.1.4 Square Duct Corner Flow

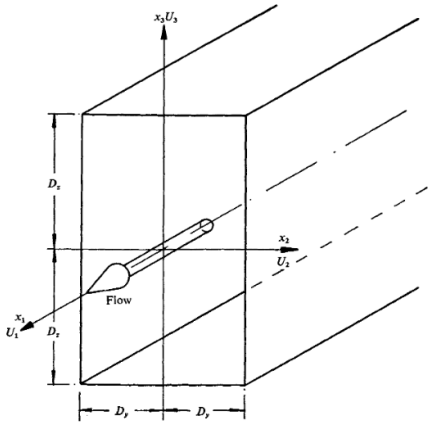


Figure 4.6 – Configuration of the square duct experiment. Taken from [72]

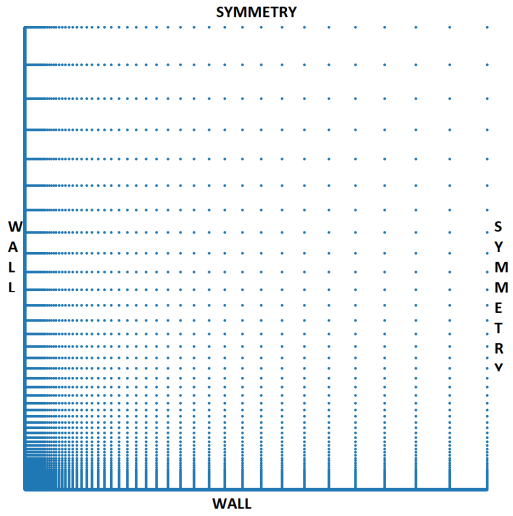


Figure 4.7 – Mesh for the square duct experiment.

The study of turbulent flow in a 3D square duct configuration is particularly interesting due to the presence of secondary motions, which have a significant impact on flow behaviour. These secondary flows, though often only about 1% of the axial velocity, can greatly affect both local and overall flow properties, such as pressure drop and heat transfer, making them important for applications in engineering systems. In square ducts, secondary motions lead to complex flow patterns, including vortex structures that affect wall shear stress distribution and can influence flow efficiency. These effects become critical in improving predictions for pressure loss and thermal transport in non-circular cross-sections, which are key to optimising designs in various industries. Moreover, secondary motions introduce challenges for turbulence modelling The turbulent flow inside a square duct was studied experimentally by Melling and Whitelaw [72]. This configuration provides a controlled environment for capturing detailed velocity profiles, turbulence statistics, and pressure measurements at various sections of the duct.

The calibration data for the RANS models includes mean secondary velocity contours at a specific section of the flow ($x/D = 36.8$). Note that classical linear eddy viscosity models struggle to capture secondary velocities in the corner flow, and the QCR correction, presented earlier, is needed to resolve this accurately. This building block is essential for wall-bounded 3D

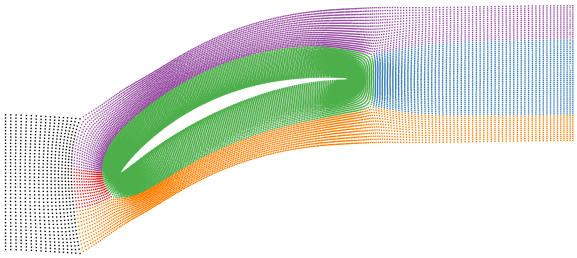


Figure 4.8 – Mesh for the NACA65-V103 compressor cascade

Scenario	S_1	S_2	S_3	S_4
Angle of attack (°)	36.99	39.97	44.09	49.2
$Mach_{in}$	0.654	0.674	0.666	0.65
Re_{in}	302000	302000	298000	289000
Tu(%)	2.9	3.3	3.2	3.5

Table 4.1 – Experimental conditions of the different scenarii

flows, as corner vortices appear frequently at the foot of a compressor blade in turbomachinery. The mesh, provided by Camille Matar, was an extrusion of a 2D mesh shown in Figure 4.7 and is composed of 50x50x30 nodes.

4.2 Turbomachinery Configurations

4.2.1 2D NACA65-V103 Linear Compressor Cascade

This study also examines the NACA65-V103 linear compressor cascade under various operating conditions already studied in [20, 22, 21, 19]. Although high-fidelity simulations exist, reference data for this configuration were generated using an EARSM $k-k\ell$ RANS model, as high-fidelity data fields are not publicly available. Note that the EARSM model, while not high-fidelity, uses a nonlinear approach that is expected to perform better than linear models in complex configurations such as this one, where boundary layer separation can occur. Synthetic reference data generated using an EARSM $k-k\ell$ model are provided for four scenarios, as shown in Table 4.1.

The RANS simulations used periodic boundary conditions to represent the entire cascade, requiring only a single blade passage. The computational grid, shown in Figure 4.8, was created using the Autogrid software for the *elsA* solver. The domain spans from 0.5 axial chord lengths

upstream of the leading edge to 1.0 chord lengths downstream of the trailing edge.

4.2.2 3D NACA65 Linear Compressor Cascade

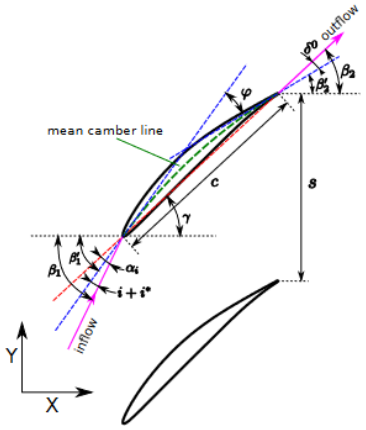


Figure 4.9 – NACA65 blade geometry

Symbol	Name	Magnitude	Formula
c	Chord	150.0 mm	
φ	Camber angle	$23.22°$	$\varphi = \beta_1' - \beta_2'$
γ	Stagger angle	$42.7°$	
s	Pitch	134.0 mm	
σ	Solidity	1.12	$\sigma = c/s$
h	Blade span	370.0 mm	
AR	Aspect ratio	2.47	$AR = h/c$
i	Incidence angle	$-5° \sim 7°$	
i^*	Optimum incidence	$0.18°$	
α_i	Attack angle		$\alpha_i = \varphi/2 + i + i^*$
β_1'	Design inflow angle	$54.31°$	$\beta_1' = \gamma + \varphi/2$
β_1	Actual inflow angle		$\beta_1 = \beta_1' + i + i^*$
β_2'	Design outflow angle	$31.09°$	$\beta_2' = \gamma - \varphi/2$
β_2	Actual outflow angle		$\beta_2 = \beta_2' + \delta^0$
δ^0	Flow deviation angle		
θ	Flow turning angle		$\theta = \beta_1 - \beta_2$ $= \varphi + i + i^* - \delta^0$

Figure 4.10 – Geometrical parameters of the cascade.

The more complex configuration under investigation is a 3d linear compressor cascade comprising modified NACA 65 blades. The original thickness distribution of the blade is derived from the NACA 65-009 blade. These blades are installed periodically with a stagger angle γ of $42.7°$ and a pitch s of $0.89c$, while the blade span is $2.47c$. Consequently, the solidity of the present cascade is $\sigma = 1.12$, and the aspect ratio is $AR = 2.47$. The design inlet and outlet angles are $\beta_1' = 54.31°$ and $\beta_2' = 31.09°$, respectively. The two-dimensional geometry of the blade is depicted in Figure 4.9 and Table 4.10.

The high-fidelity data employed for this configuration consists of time-averaged results from a wall-resolved Large Eddy Simulation (LES) for two different incidence angles: $4°$ and $7°$. The wall-resolved LES computations were performed by Dimitrios Papadogiannis, a research engineer at Safran Tech [29]. This configuration was also previously experimentally and numerically by Gao *et al.* [44].

The RANS simulation geometry includes one half-blade and an end-wall. Only a half-blade over the span has been computed due to the area of interest being close to the hub, with the

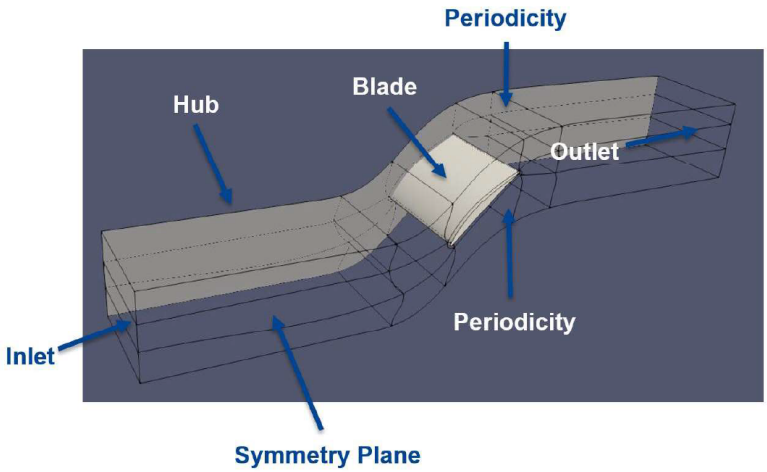


Figure 4.11 – NACA65 3D configuration for RANS simulations

second half simulated by symmetry. No tripping bands were added to the geometry as they are not required in RANS to trigger turbulence transition. However, it should be noted that tripping bands are present on the blades used for the experiments and the LES, which must be considered during comparison. Gao [45] investigated the influence of these bands on the boundary layer and highlighted some discrepancies in velocity profiles. Nevertheless, the global effect on losses has not been quantified and would likely depend heavily on the height of the tripping bands.

The computational domain extends three blade chords upstream and includes a sufficiently long downstream section to prevent any influence of the outlet condition on the blade wake. The complete linear cascade is simulated by applying a periodic condition, allowing the simulation of only one of the blades. The domain meshing was performed using the software ICEM. Initially, a two-dimensional mesh was meticulously created using a blocking method and subsequently extruded in the third direction (perpendicular to the hub) with an appropriate growth rate, ranging between 1.05 and 1.20 depending on the mesh resolution. The minimum cell height is 5×10^{-6} m for cells adjacent to walls, including the hub and blade surfaces. Several meshes with varying levels of refinement were generated, with the fine mesh used in this study consisting of approximately 20 million cells. The quality of the mesh was verified using the determinant $3 \times 3 \times 3$ method, which evaluates the ratio between the determinant of the Jacobian matrix for each element. This is a common quality metric, with a value greater than 0.2 being considered acceptable for any solver. For the mesh employed in this study, the determinant was greater

than 0.85, indicating a sufficiently high-quality mesh suitable for accurate computational fluid dynamics (CFD) simulations.

This chapter has detailed the various flow configurations employed in this thesis, ranging from fundamental test cases to more complex scenarios, such as the square duct corner flow and the NACA65 compressor cascades. The methodologies developed in the subsequent chapters will leverage these configurations to advance Bayesian calibration techniques and model aggregation methods to enhance the reliability of turbulence predictions across diverse flow conditions.

5

Space-dependent updated model aggregation

In this chapter, we investigate a model aggregation methodology inspired by the XMA algorithm of De Zordo-Banliat *et al.* [19] presented before to quantify and reduce the uncertainty associated with a prediction using RANS models. Differently from that work, which aggregates on-the-shelf RANS models with their nominal closure parameters, we seek to improve the accuracy of the model mixture by first calibrating the component models to enhance their intrinsic performance before the aggregation step.

This methodology will be applied to the NACA65 V103 linear cascade and will consider four widely used RANS models, i.e. the Spalart-Allmaras model [97], the 2006 Wilcox $k-\omega$ model [108], the Smith $k-\ell$ model [95], and the Launder-Sharma $k-\epsilon$ model [58].

Chapter contents

5.1 Methodology

The proposed methodology is based on the Bayesian framework for model calibration and averaging. It consists of the following steps:

1. Offline calibration and model weighting,
2. Regression of space-dependent weights,
3. Online prediction,

described in detail in the following.

5.1.1 Offline calibration and model weighting

Consider a set $\mathcal{M} = \{M_1, \ldots, M_m, \ldots, M_{N_M}\}$ of N_M models, each characterised by a vector of parameters $\boldsymbol{\theta}_m$, and consider a vector of N_D observed data $\Delta_{obs} = (d_1, d_2, ..., d_i, ..., d_{N_D})$ used to calibrate all the different models, where d_i corresponds to the observation of some flow quantity at a spatial location $\boldsymbol{x}_i$ of the computational domain in the present framework.

We calibrate the N_M baseline models by using the Bayesian framework presented in Chapter 3 to obtain N_M updated models. This involves three phases: i) stochastic modelling of the problem, ii) sampling from the posterior distribution of each vector of parameters $\boldsymbol{\theta}_m$, and iii) propagation of the posterior distribution within the CFD solver for future predictions. In the aim of reducing the computational burden of propagating the full posterior distributions through the CFD solver, we approximate them as multidimensional Dirac distribution, centred on the joint Maximum A Posteriori (MAP) estimate $\boldsymbol{\theta}_m^{MAP}$ of the parameter vector:

$$p(\boldsymbol{\theta}_m|\Delta_{obs}, M_m) \approx \delta(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^{MAP}) \tag{5.1}$$

as proposed in [36]. While this approximation neglects the posterior parametric uncertainty, it substantially reduces the computational cost to that of a single deterministic CFD computation.

The Bayesian Model Averaging (BMA) framework presented in Chapter 3, uses the posterior model plausibility as weight. Nonetheless, the model plausibilities are thus scalar and constant throughout the flow.

The proposed method aims to weight models using space-dependent weights, i.e., spatially varying weight for each observation $d_i \in \Delta_{obs}$ in location x_i of the (x, y) coordinate space. To that extent, we can consider each observation independent from one another and define, for all $d_i \in \Delta_{obs}$, local model plausibilities $p(M_m \mid x_i, \Delta_{obs})$.

Based on the BMA framework, for all $d_i \in \Delta_{obs}$, we define the local posterior weight plausibility :

$$p(M_m \mid x_i, \Delta_{obs}) = \frac{p(\Delta_{obs}|M_m, x_i)p(M_m)}{\sum_{k=1}^{K} p(\Delta_{obs}|M_k, x_i)p(M_k)} \tag{5.2}$$

where $p(M_m)$ is the prior model plausibility and where $p(\Delta_{obs}|M_m, x_i) = \int p(\Delta_{obs}|\boldsymbol{\theta}_m, M_m, x_i)p(\boldsymbol{\theta}_m|M_m)d\boldsymbol{\theta}_m$ is the marginal likelihood in x_i. Let's further assume that the MAP approximation in equation 5.1 holds in every point of the flow, i.e.:

$$\forall d_i \in \Delta_{obs}, \quad p(\boldsymbol{\theta}_m|x_i, \Delta_{obs}, M_m) \approx \delta(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^{MAP}) \tag{5.3}$$

This facilitates the computation of the local marginal likelihood, as it simplifies as $p(\Delta_{obs}|M_m, x_i) = p(\Delta_{obs}|\boldsymbol{\theta}_m^{MAP}, M_m, x_i)$, which is the likelihood at x_i using parameter values $\boldsymbol{\theta}^{MAP}$. This hypothesis suggests that the MAP estimates of the model's parameters would still be $\boldsymbol{\theta}_m^{MAP}$ if estimated individually on each observation in the flow. This is certainly not the case

as the model parameters would be influenced differently by the calibration using points from different regions of the flow (the boundary layer and the free stream for example). However, this is a working hypothesis necessary to define the space-dependent weight within the Bayesian framework.

Such locally variable plausibilities $p(M_m|x_i, \Delta_{obs})$ are then used as weights to aggregate the N_M component models into a space-dependent model mixture where weights are locally assigned to each model. For clarity, let us denote

$$W_{im} = p(M_m|x_i, \Delta_{obs}) \tag{5.4}$$

the weight (or plausibility) of model M_m at x_i after calibration against Δ_{obs}.

5.1.2 Regression of space-dependent weights

The weights W_{im} are space-dependent, i.e. they can be expressed as $W_{im}(x_i)$. This coordinate-weight relationship is geometry and scenario-dependent, meaning that the weights are known only at the locations with observed data, and cannot be used to predict a new point of the calibration case or a new case.

Following [19], the computed weights are thus regressed onto a feature space using a Random Forest Regressor (RFR). The chosen features, denoted $\boldsymbol{\eta}$, similar to those initially proposed in [62] are gathered in table 5.1.

The goal is to learn the relationship between the feature space and the space-dependent weight, to be able to predict for any feature set $\boldsymbol{\eta}$, the weight for model M_m:

$$\boldsymbol{\eta} \xrightarrow{RFR} w_m(\boldsymbol{\eta}) \tag{5.5}$$

Regression falls in the supervised learning category of Machine Learning (ML), which is a broad and well-explored topic in the literature. Many well-known model algorithms fall into this category: from the simplest least squares linear models to complex non-linear regression models, Support Vector Machine (SVM) and kernel methods, Gaussian Process Regression, Ensemble Methods based on trees [46], and Neural Networks are a few examples.

In the remainder of this study, supervised regression is based on Random Forests for three primary reasons. First, they are well-suited to handle large datasets and numerous input features.

#	Description	Formula
1	Non-dimensionalised Q criterion	$\frac{\|\mathbf{\Omega}\|^2-\|\mathbf{S}\|^2}{\|\mathbf{\Omega}\|^2+\|\mathbf{S}\|^2}$
2	Turbulence intensity	$\frac{k_t}{0.5\overline{u}_i\overline{u}_i+k_t}$
3	Turbulence Reynolds number	$\min\left(\sqrt{\frac{k_t d}{50\nu}}, 2\right)$
4	Pressure gradient along streamline	$\frac{\overline{u}_k\frac{\partial\overline{p}}{\partial x_k}}{\sqrt{\frac{\partial\overline{p}}{\partial x_j}\frac{\partial\overline{p}}{\partial x_j}\overline{u}_i\overline{u}_i+\left\|\frac{\partial\overline{p}}{\partial x_l}\right\|}}$
5	Ratio of turbulent time scale to mean strain time scale	$\frac{\|\mathbf{S}\|k}{\|\mathbf{S}\|k_t+\varepsilon}$
6	Viscosity ratio	$\frac{\nu_T}{100\nu+\nu_T}$
7	Ratio of pressure normal stresses to normal shear stresses	$\frac{\sqrt{\frac{\partial\overline{p}}{\partial x_i}\frac{\partial\overline{p}}{\partial x_i}}}{\sqrt{\frac{\partial\overline{p}}{\partial x_i}\frac{\partial\overline{p}}{\partial x_i}}+0.5\rho\frac{\partial\overline{u}_k^2}{\partial x_k}}$
8	Marker of Gorle et al., deviation from parallel shear flow	$\frac{\|\overline{u}_k\overline{u}_l\frac{\partial\overline{u}_k}{\partial\overline{u}_l}\|}{\sqrt{\overline{u}_n\overline{u}_n\overline{u}_i\frac{\partial\overline{u}_i}{\partial x_j}\overline{u}_m\frac{\partial\overline{u}_m}{\partial x_j}}+\|\overline{u}_i\overline{u}_j\frac{\partial\overline{u}_i}{\partial x_j}\|}$
9	Ratio of convection to production of k_t	$\frac{\overline{u}_i\frac{\partial k}{\partial x_i}}{\|\overline{u_i'u_j'}S_{jl}\|+\overline{u}_l\frac{\partial k_t}{\partial x_l}}$
10	Ratio of total Reynolds stresses to normal Reynolds stresses	$\frac{\|\overline{u_i'u_j'}\|}{k_t+\|\overline{u_i'u_j'}\|}$

Table 5.1 – Features and their descriptions

Given our aim to evaluate the influence of extensive data sets on prediction, it is essential to select an appropriate regressor. Secondly, we anticipate non-linear behaviour in the regression, with possibly sharp variations and significant gradients. Random Forests, being based on regression

trees, are well-equipped to capture such behaviour. Lastly, Random Forests have demonstrated good generalisation performance in the context of CFD [63, 103] and are a natural evolution of the ensemble of CART trees employed by Yu et al. in their original algorithm [113].

To summarise, Random Forests are an ensemble method that combines the predictions of multiple base models or weak learners—in this case, decision trees—to generate a robust prediction. Ensemble methods are particularly beneficial for decision trees as they tend to overfit the data, leading to biased predictions. More specifically, Random Forests utilises bagging (sampling with replacement from the original training dataset) to create a randomised dataset for each tree in the forest and randomly select a fixed number of input variables for consideration at each node split. This diversity in both input variables and training examples significantly reduces prediction variance, while the combination of multiple predictions inherently reduces bias. Random Forests are also efficient to construct, as they do not require pruning to prevent overfitting (unlike decision trees) and can be easily parallelised. Additionally, the Random Forest Regressor (RFR) is particularly advantageous for interpolating model weights onto a feature set due to its straightforward implementation using the *scikit-learn* library in Python [80], simplifying deployment and integration within complex workflows. By leveraging the collective power of numerous trees, the RFR enhances predictive performance and generalisation, which is crucial for deriving reliable and precise model weights across diverse feature landscapes. Finally, four Random Forest hyper-parameters are considered in this study: the number of trees, the number of input features, the criterion for node split selection, and the minimum number of elements in each terminal leaf. These parameters' values are optimised via cross-validation.

5.1.3 Online prediction

Let us now present the workflow for the prediction step. The first step consists of predicting the flow with each of the N_M updated RANS models. The solutions are used to compute the prediction quantity of interest Δ^* and the set of features $\boldsymbol{\eta}^*$, in each coordinate point of the prediction configuration. Note that the prediction configuration and the prediction QoI can be similar or different from the calibration ones.

The features $\boldsymbol{\eta}^*$ are then supplied to the Random Forest interpolator to evaluate the weight of each updated model at each prediction location $w_m(\boldsymbol{\eta}^*)$.

Once the weights are known, the aggregated solution can be reconstructed similarly as in the BMA expression from chapter 3. A more detailed demonstration of this formulation is given at

the end of this section.

For each Δ^* the prediction QoI at location $\boldsymbol{\eta}^*$ in the feature space, the predictive posterior probability after training using Δ_{obs} is :

$$p\left(\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*\right) = \sum_{m=1}^{N_M} p\left(\Delta^* \mid M_m, \Delta_{obs}, \boldsymbol{\eta}^*\right) w_m(\boldsymbol{\eta}^*) \tag{5.6}$$

which is the average of the prediction using the updated models weighted by the predicted weights of each updated model at the prediction location $\boldsymbol{\eta}^*$.

The latter can be used to estimate the aggregated expectancy and variance:

$$\mathbb{E}\left(\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*\right) = \sum_{m=1}^{N_M} \mathbb{E}\left(\Delta^* \mid M_m, \Delta_{obs}, \boldsymbol{\eta}^*\right) w_m(\boldsymbol{\eta}^*) \tag{5.7}$$

$$\mathrm{Var}\left(\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*\right) = \sum_{m=1}^{N_M} \left(\mathbb{E}\left(\Delta^* \mid M_m, \Delta_{obs}, \boldsymbol{\eta}^*\right) - \mathbb{E}\left(\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*\right)\right)^2 w_m(\boldsymbol{\eta}^*) \tag{5.8}$$

The variance expresses simply the consensus in the values of Δ^* predicted by the competing models. Note that the within-model variance, i.e. variance due to model parametric uncertainty, is zero because the posterior distribution of the model parameters is approximated as a Dirac distribution (MAP approximation).

In the next section, the preceding framework is applied to the NACA65-V103 compressor cascade presented in Chapter 4.2.1.

5.1.4 Summary of the method

The method consists of 3 steps :

1. Offline calibration and model weighting

   — **Calibrate** a set of N_M models, characterised by parameters $\boldsymbol{\theta}_m$ against a vector of N_D observed data Δ_{obs} using the Bayesian framework and **approximate** the posterior distributions as Dirac distributions centred around the MAP estimate $\boldsymbol{\theta}_m^{MAP}$.

   — **Suppose** that each $d_i \in \Delta_{obs}$ is independent to **compute** the posterior model plausibility (weights) at the location x_i that will be used to the model aggregation step : $W_{mi} = p(M_m|x_i, \Delta_{obs})$

2. Regression of space-dependent weights using a Random Forest Regressor to map the weights in the feature space.
3. Online prediction
   - — **Predict** the quantity of interest Δ^* with the N_M updated models using the MAP estimates of the parameters $\boldsymbol{\theta}_m^{MAP}$.
   - — **Compute** the features $\boldsymbol{\eta}^*$ to **evaluate** the weights of each updated model $w_m(\boldsymbol{\eta}^*)$ using the RFR .
   - — **Aggregate** each prediction to compute the predictive posterior distribution at every prediction location: $p\left(\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*\right) = \sum_{m=1}^{N_M} p\left(\Delta^* \mid M_m, \Delta_{obs}, \boldsymbol{\eta}^*\right) w_m(\boldsymbol{\eta}^*)$. From this posterior predictive distribution, the expectancy and variance can be computed.

Demonstration of the XUMA prediction formula

Consider a set $\mathcal{M} = \{M_1, \ldots, M_m, \ldots, M_{N_M}\}$ of N_M models, each characterised by a vector of parameters $\boldsymbol{\theta}_m$, and consider a vector of N_D observed data $\Delta_{obs} = (d_1, d_2, ..., d_i, ..., d_{N_D})$ used to calibrate all the different models, where d_i corresponds to the observation of some flow quantity at a spatial location x_i of the computational domain in the present framework.

Finally, let us also consider $\boldsymbol{\eta}^*$ and $\Delta^*(\boldsymbol{\eta}^*)$, where $\boldsymbol{\eta}^*$ is a local set of flow features, and Δ^* is a local piece of information about the flow available at location $\boldsymbol{\eta}^*$ in the feature space. Application of the BMA leads to equation 5.9:

$$p(\Delta^*|\Delta_{obs}, \boldsymbol{\eta}^*) = \sum_{m=1}^{N_M} p(M_m|\Delta_{obs}, \boldsymbol{\eta}^*) \int p(\Delta^*|\boldsymbol{\eta}^*, \Delta_{obs}, M_m, \boldsymbol{\theta}_m) p(\boldsymbol{\theta}_m|\boldsymbol{\eta}^*, \Delta_{obs}, M_m) d\boldsymbol{\theta}_m \quad (5.9)$$

where the well-known data conditional independence hypothesis is involved such that $p(\Delta^*|\boldsymbol{\eta}^*, \Delta_{obs}, M_m, \boldsymbol{\theta}_m) = p(\Delta^*|\boldsymbol{\eta}^*, M_m, \boldsymbol{\theta}_m)$. Now assume that $p(\boldsymbol{\theta}_m|\boldsymbol{\eta}^*, \Delta_{obs}, M_m) = p(\boldsymbol{\theta}_m|\Delta_{obs}, M_m)$ - this means that the posterior distribution of $\boldsymbol{\theta}_m$ is independent of the points to be predicted. Then 5.9 becomes:

$$p(\Delta^*|\boldsymbol{\eta}^*, \Delta_{obs}) = \sum_{m=1}^{N_M} p(M_m|\Delta_{obs}, \boldsymbol{\eta}^*) \underbrace{\int p(\Delta^*|\boldsymbol{\eta}^*, M_m, \boldsymbol{\theta}_m) p(\boldsymbol{\theta}_m|\Delta_{obs}, \boldsymbol{\eta}^* M_m) d\boldsymbol{\theta}_m}_{p(\Delta^*|\Delta_{obs}, \boldsymbol{\eta}^*, M_m)} \quad (5.10)$$

where $p(\Delta^*|\boldsymbol{\eta}^*, M_m, \boldsymbol{\theta}_m)$ is the likelihood function[1] - as Δ^* involves $M_m(\boldsymbol{\eta}^*)$ in the stochastic model -

1. See comments in Section 3.1.3

and $p(\boldsymbol{\theta}_m|\Delta_{obs}, \boldsymbol{\eta}^*, M_m)$ is the parameter posterior distribution at location $\boldsymbol{\eta}^*$ after calibration on Δ_{obs}. Using hypothesis in equation 5.3, $p(\boldsymbol{\theta}_m|\Delta_{obs}, \boldsymbol{\eta}^*, M_m) = \delta(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^{MAP})$

Equation 5.10 can thus be written:

$$p(\Delta^*|\boldsymbol{\eta}^*, \Delta_{obs}) = \sum_{m=1}^{N_M} p(M_m|\Delta_{obs}, \boldsymbol{\eta}^*) p(\Delta^*|\boldsymbol{\eta}^*, M_m, \boldsymbol{\theta}_m^{(MAP)}) \tag{5.11}$$

Applied to model probabilities, the Bayes' rule writes:

$$\begin{aligned} p(M_m|\Delta_{obs}, \boldsymbol{\eta}^*) &= \frac{p(\Delta_{obs}|M_m, \boldsymbol{\eta}^*) p(M_m, \boldsymbol{\eta}^*)}{\sum_{k=1}^{N_M} p(\Delta_{obs}|M_k, \boldsymbol{\eta}^*) p(M_k, \boldsymbol{\eta}^*)} \\ &= \frac{p(\Delta_{obs}|M_m, \boldsymbol{\eta}^*) p(M_m|\boldsymbol{\eta}^*) p(\boldsymbol{\eta}^*)}{\sum_{k=1}^{N_M} p(\Delta_{obs}|M_k, \boldsymbol{\eta}^*) p(M_k|\boldsymbol{\eta}^*) p(\boldsymbol{\eta}^*)} \\ &= \frac{p(\Delta_{obs}|M_m, \boldsymbol{\eta}^*) p(M_m|\boldsymbol{\eta}^*)}{\sum_{k=1}^{N_M} p(\Delta_{obs}|M_k, \boldsymbol{\eta}^*) p(M_k|\boldsymbol{\eta}^*)} \end{aligned} \tag{5.12}$$

Traditionally, $p(\Delta_{obs}|M_m, \boldsymbol{\eta}^*)$ [2] is computed by integration while $p(M_m|\boldsymbol{\eta}^*)$ is explicitly user-defined (by taking a discrete uniform prior over models, most of the time). Nonetheless, in this approach, $p(\Delta_{obs}|M_m, \boldsymbol{\eta}^*)$ is instead directly given as a function of $\boldsymbol{\eta}^*$ along with local evaluations of model ranking at each point of the dataset:

$$p(M_m|\Delta_{obs}, \boldsymbol{\eta}^*) = w_m\left(M_1, \cdots, M_{N_M}, \Delta_{obs}, \boldsymbol{\eta}^*\right) \tag{5.13}$$

where w_m is a weighting function. Hence the prior model probabilities are implicitly given and a strong assumption is that for any posterior model probability $p(M_m|\Delta_{obs}, \boldsymbol{\eta}^*)$, there exists a prior $p(M_m|\boldsymbol{\eta}^*)$ such that equation 5.12 holds. Ultimately, the function w_m is involved in equation 5.13 is the RFR.

2. Note that we could also assume that $p(\Delta_{obs}|M_m, \boldsymbol{\eta}^*) = p(\Delta_{obs}|M_m)$ to make equation 5.12 look more similar as equation 3.17 though it is unnecessary in the context of this demonstration.

5.2 Calibration of RANS Models for a compressor cascade

We consider the NACA 65-V103 compressor cascade configuration. Data are available for four different flow scenarios, previously presented in Section 4.2.1.

We calibrate four competing RANS models *i.e.* the Spalart-Allmaras model [97], the 2006 Wilcox $k-\omega$ model [108], the Smith $k-\ell$ model [95], and the Launder-Sharma $k-\epsilon$ model [58], whose descriptions can be found in Chapter 2.

The following stochastic model is used to build the likelihood function:

$$\Delta_{obs}(\mathbf{x}) = \Delta_{true}(\mathbf{x}) + \boldsymbol{\varepsilon}(\mathbf{x}) = \boldsymbol{\eta} \odot \Delta(\mathbf{x}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}(\mathbf{x}) \tag{5.14}$$

where the random vector $\boldsymbol{\varepsilon}$ represents the experimental noise and the random vector $\boldsymbol{\eta}$ is the model-form uncertainty. A Hadamard (component-wise) product is assumed between $\boldsymbol{\eta}$ and the RANS model output Δ. In this work, the chosen calibration quantity Δ is the total pressure.

We model the experimental noise as $\varepsilon \sim \mathcal{N}(0, (0.01\Delta_{obs})^2)$. For the present synthetic data, the noise is not due to experimental uncertainties but rather to the errors introduced by surrogate modelling.

The model-form uncertainty is modelled as $\eta \sim \mathcal{N}(1, \sigma_\eta^2)$. This approach assumes that the larger the QoI, the greater the uncertainty. The hyper-parameter σ_η is calibrated alongside the model parameters and is assigned a uniform prior distribution between 0 and 1. This model-form error assumes that model errors at each point in our dataset are independent. Although rather simplistic (the data is correlated to some extent because they are extracted from the same flow field), this formulation greatly reduces the computational cost of the calibration step as it leads to an easily invertible diagonal covariance matrix.

Under the preceding assumptions, the likelihood can finally be expressed as:

$$\Delta_{obs} \sim \mathcal{N}(\Delta, (\sigma_\eta \Delta)^2 + (0.01\Delta_{obs})^2) \tag{5.15}$$

The preceding inference framework is applied to the calibration of uncertain closure parameters of the chosen RANS turbulence models. For each model, some parameters are selected to be updated using Bayesian calibration. Prior distributions then need to be specified before the calibration. Margheri *et al.* [70] investigated the parametric uncertainty of two widely employed turbulence models and used physical constraints and data collected from the literature to construct

Param	BV	LB	UB
β	0.0708	0.03	0.08
σ	0.5	0.35	0.55
σ_s	0.6	0.30	1.00
σ_{do}	0.125	0.06	0.20
κ	0.41	0.40	0.60

(a) $k-\omega$ model

Param	BV	LB	UB
$c_{\epsilon 2}$	1.92	1.90	2.50
σ_k	1.0	0.55	1.20
κ	0.41	0.30	0.65

(b) $k-\epsilon$ model

Param	BV	LB	UB
B1	18	1	36
E2	1.20	0.01	1.60
σ_l	1.43	0.65	3.60
σ_k	1.43	0.20	2.50
κ	0.41	0.36	0.95

(c) $k-l$ model

Param	BV	LB	UB
$cb1$	0.1355	0.07	0.15
$cb2$	0.622	0.50	1.00
$cv1$	7.1	6.00	15.0
$cw2$	0.3	0.10	0.90
$cw3$	2.0	0.80	2.40
σ	2./3.	0.60	1.50
κ	0.41	0.36	0.55

(d) S-A model

Table 5.2 – Baseline parameter values (BV), lower (LB) and upper (UB) prior bounds for all models

prior probability distributions for the coefficients. Although constructing such informative prior distributions for various turbulence models using such an approach is feasible, it could strongly constrain the posterior, especially if relatively few observed data are available. Therefore, we follow common practice in Bayesian inference [3, 73] and choose uninformative priors, *i.e.* uniform priors, for each component of $\boldsymbol{\theta}$ (assumed independent). Given the careful design of RANS models, we confidently assume that the nominal values of the parameters should be included in the range of the prior. Consequently, we select uniform priors that encompass the nominal values of closure coefficients found in the literature, as previously done in [17, 33].

Furthermore, there is no evidence to suggest that model predictions would improve by selecting closure coefficients with significant deviations from the standard values. The prior intervals are thus chosen to be sufficiently large to allow comprehensive exploration of the parameter space while avoiding values that are too extreme, which could prevent the CFD solver from converging or result in non-physical outcomes. It is also important to note that excessively large prior distributions may lead to overfitting, producing posterior coefficients that fit the calibration data well but degrade predictions of unobserved QoIs.

The parameters selected for calibration are gathered in Table 5.2 where we also report their baseline values and the lower and upper bound of the corresponding prior uniform distribution.

5.2.1 Posterior distributions

Four independent chains are launched for each model using the NUTS sampler [50]. Each chain samples 30,000 iterations. The convergence of each chain is assessed using the $\hat{R}$ indicator and by checking the visual appearance of the sampling traces. Figure 5.1 shows the posterior probability marginal distributions after the calibration of the $k-\ell$ model parameters and the hyper-parameter σ_η for scenario S_4. Several observations are in order.

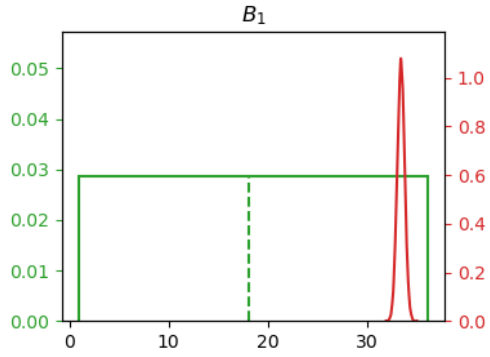
Firstly, the posterior distributions are not centred around the parameter's baseline values. This suggests that the baseline values do not maximise the likelihood of the model predictions matching the reference data and that adjusting these values can improve the likelihood. Secondly, the hyper-parameter posterior distribution is centred around a value lesser than 0.01. This indicates that the observed data can be matched by adjusting the model parameters, with no need for a large model-inadequacy parameter σ_η. Thirdly, the value of the hyper-parameter σ_η can be interpreted as a degree of confidence that the data attributes to the newly updated model. It serves as an indicator of confidence in the updated models. A model with a large hyper-parameter should be used with greater caution compared to a model with a smaller hyper-parameter because the former is more uncertain. Typical plots of the posterior distributions of σ_η are shown in Figure 5.2 and suggest that the $k-\varepsilon$ and the $k-\ell$ models should be trusted less than the Spalart-Allmaras or the $k-\omega$ models.

Finally, we note that the posterior distributions are well peaked, indicating that the parameter posterior distributions are well-informed by the data. This result justifies the approximation of the posterior distributions as Dirac distributions centred around the MAP. Also, note that the MAP estimates for all parameters in all clusters are available in the Appendix A.

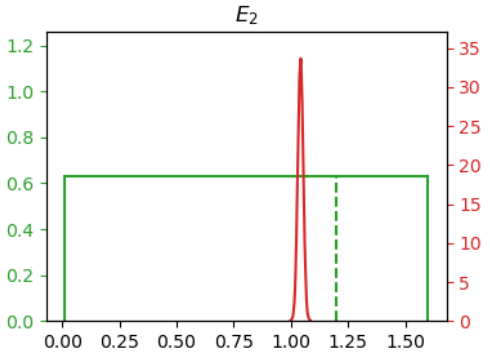
5.2.2 Influence of the dataset size

In the previous section, one point out of two in the mesh is used as a reference data point for the calibration. This means that the probes used as reference data points are one point out $N_{\text{thin}} = 2$. The probes used are depicted in orange in Figure 5.3. Using every other point of the mesh for calibration, i.e. $N_{\text{thin}} = 2$ represents an amount of data that is only available for simulation results. To test the possibility of using sparser data like experimental data, it is important to investigate the impact of the thinning number values N_{thin} on the calibration results.

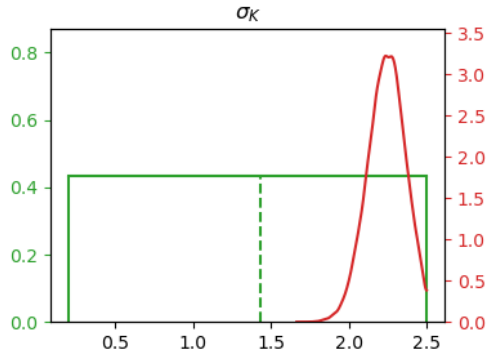
Figure 5.4 shows the posterior distributions obtained while increasing N_{thin}. The figures



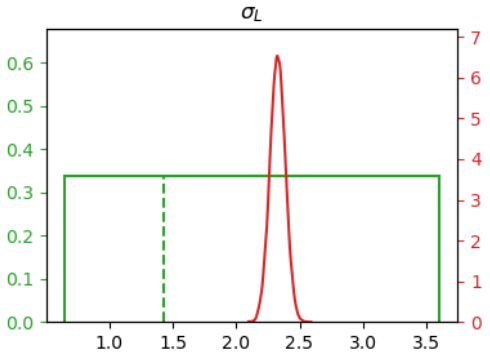
(a) RANS parameter B_1



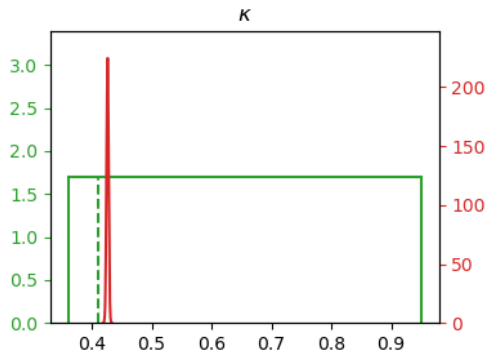
(b) RANS parameter E_2



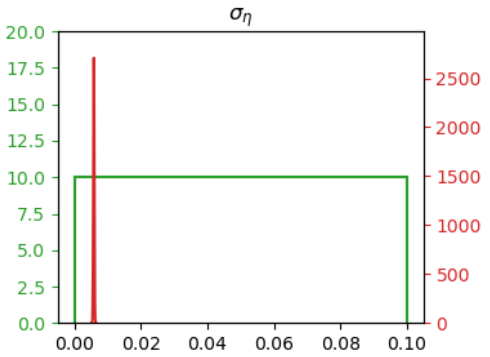
(c) RANS parameter σ_K



(d) RANS parameter σ_L

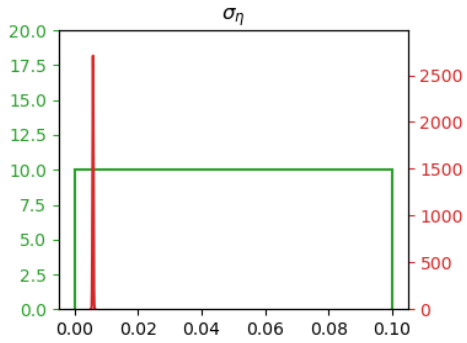


(e) RANS parameter κ

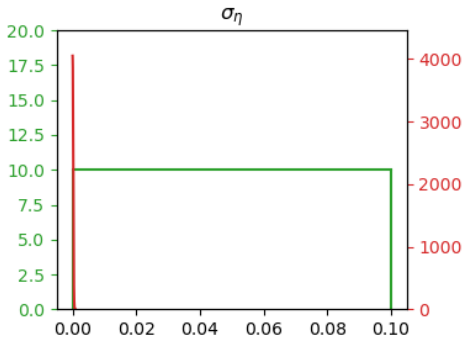


(f) Hyper-parameter σ_η

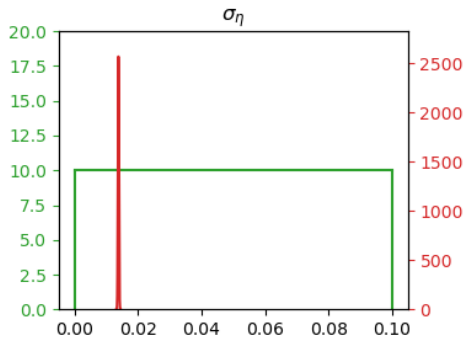
Figure 5.1 – Prior and posterior marginal distributions for the $k - \ell$ parameters and hyper-parameter on scenario S_4. The dashed line is the baseline value



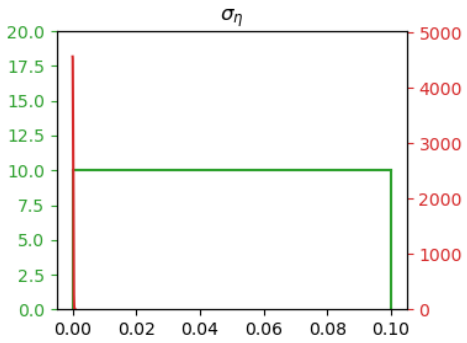
(a) $k - \ell$ model



(b) $k - \omega$ model



(c) $k - \varepsilon$ model



(d) S-A model

Figure 5.2 – Prior and posterior marginal distributions for the σ_η hyper-parameter on scenario S_4 for the 4 RANS models

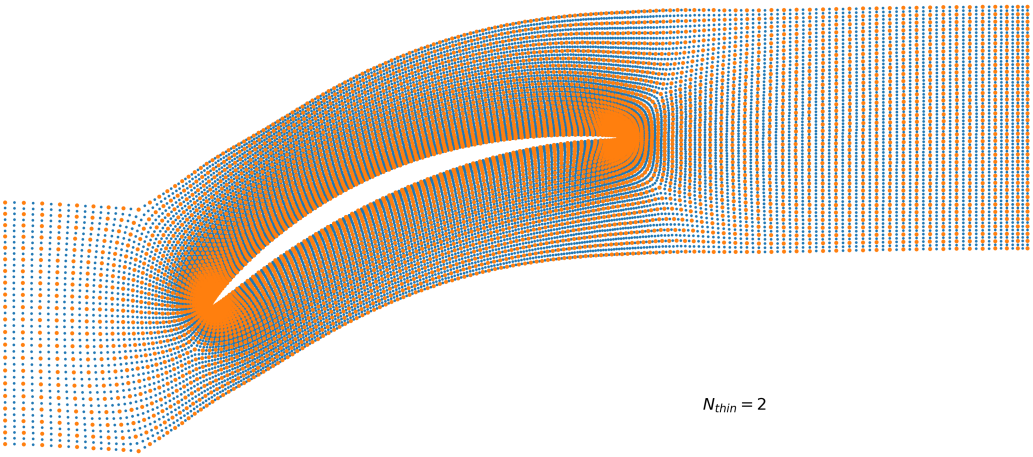


Figure 5.3 – Probes used for calibration and mesh points

suggest that the posterior modes do not vary with the thinning number values N_{thin}. However, the variance of the distributions increases with the increase of N_{thin}, suggesting that the more data the calibration step gets, the more informative it is. To consider the posterior distributions as Dirac distributions, it is then important to calibrate using a larger number of probes. However, this can be computationally expensive. A compromise may be required between the computational cost and the validity of the Dirac hypothesis. In the rest of this chapter, the thinning number considered will be fixed $N_{\text{thin}} = 2$.

5.2.3 Prediction using the updated models

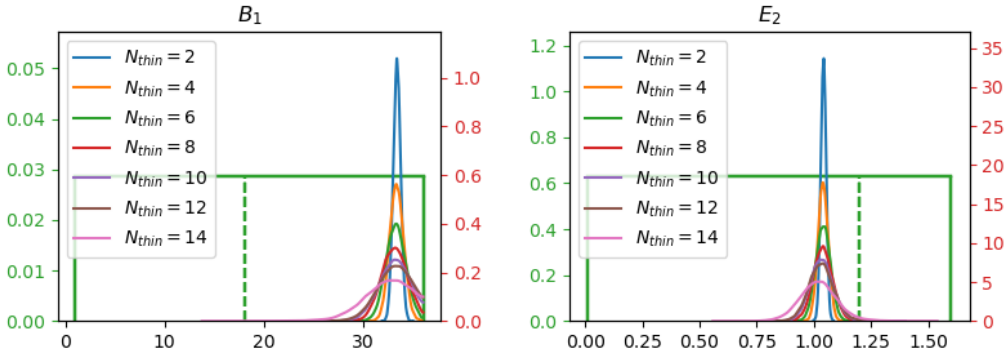
To test whether the models benefit from the calibration or not, updated models are then applied to two different configurations:

— the calibration scenario, for which a significant improvement is expected

— an unseen scenario (different geometry, different operating conditions...), to evaluate the generalisation capabilities of the approach.

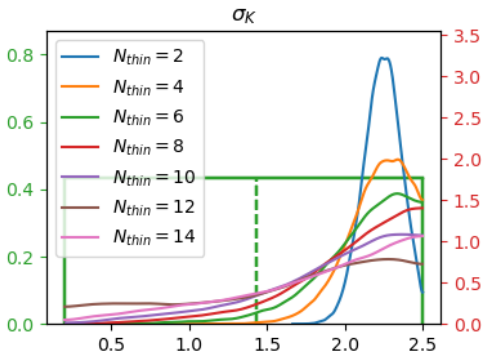
Prediction on the calibration scenario

Figure 5.5 displays the mean squared error (MSE) between predictions of the updated models and observations for the total pressure, the pressure and the velocity magnitude. These MSEs

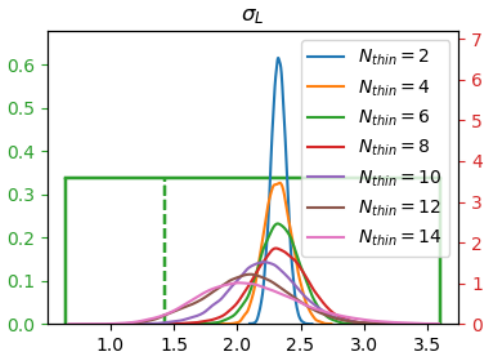


(a) RANS parameter B_1

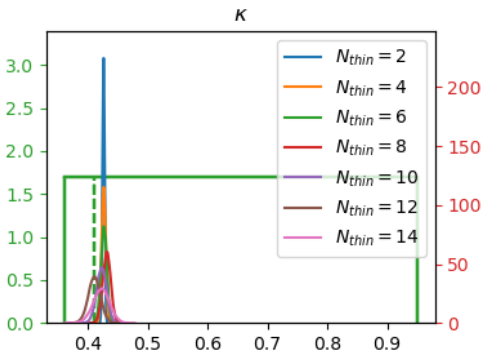
(b) RANS parameter E_2



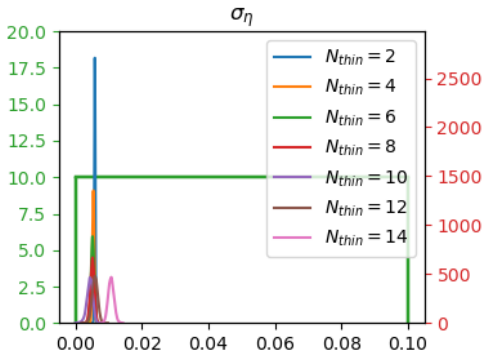
(c) RANS parameter σ_K



(d) RANS parameter σ_L



(e) RANS parameter κ



(f) Hyper-parameter σ_η

Figure 5.4 – Posteriors and prior marginal distributions for the $k - \ell$ RANS parameters and hyper-parameter on scenario S_4. The dashed lineb is the baseline value

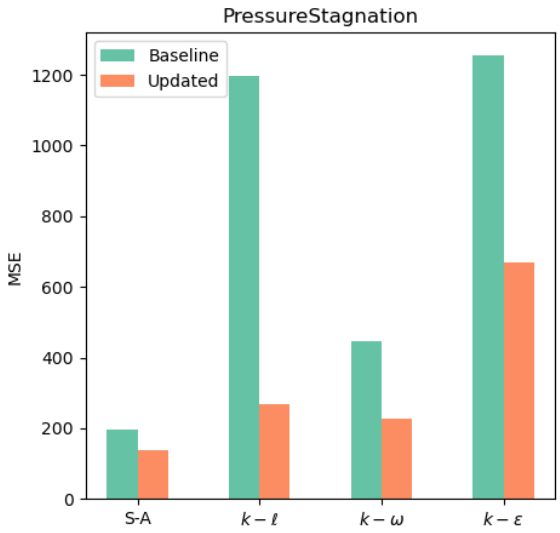
are presented in bar plots. Regarding the calibration quantity (total pressure), for all models, the calibration step significantly reduces the MSE. Notably, the performance ranking of the models remains unchanged after calibration. On the other hand, for other QoI, the impact of the calibration varies significantly between models. For instance, while the errors in both the pressure and velocity are reduced after calibration for the $k-\ell$ and $k-\omega$ models, the latter increases the S-A and $k-\varepsilon$ models. To try to understand where this phenomenon arises from, consider figure 5.6 that displays the relative error distributions of total pressure fields for the updated and baseline models in the training scenario S_4. The figures demonstrate that error reduction varies significantly across different spatial regions. In some areas, the calibration process can even increase local errors. This suggests that the calibration process allows for a decrease in local performance to achieve a better overall improvement in model performance. This indicates that the model might over-fit certain regions to minimise overall error and increase likelihood. Notably, the S-A and $k-\varepsilon$ models are the ones that specialise locally to reduce errors, even at the expense of performance elsewhere. The sub-optimal performance of these two models on quantities unseen during calibration as suggested in figure 5.5, although related to the quantity of interest, is likely due to this local specialisation phenomenon.

Prediction on an unseen scenario

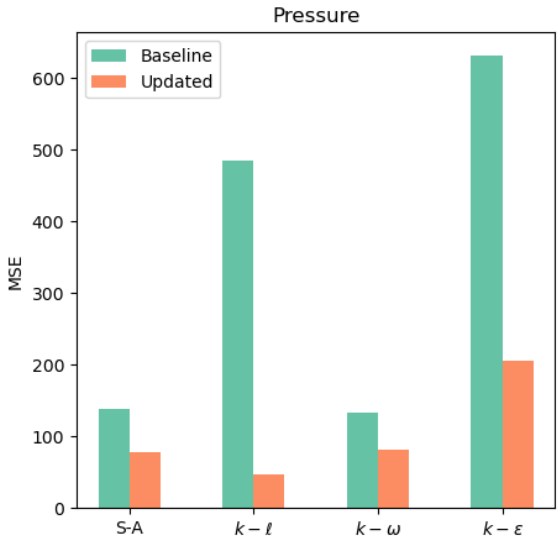
To evaluate the generalisability of the calibration, the updated models are then used to predict the flow of scenario S_3. The MSEs between the predictions and the reference are displayed in Figure 5.7 for the total pressure, the pressure and the velocity magnitude. It turns out that whatever the QoI (used for the calibration or not), the error is decreased for only two out of the four models, namely the S-A and $k-\omega$ models. One explanation could be that during the calibration step the parameters specialise to improve the global performance by improving drastically one area of the flow, the parameters do not perform well on a different quantity of interest or a different scenario. Figures 5.7b to 5.7c suggest a similar conclusion for the different quantities of interest.

5.2.4 Section summary

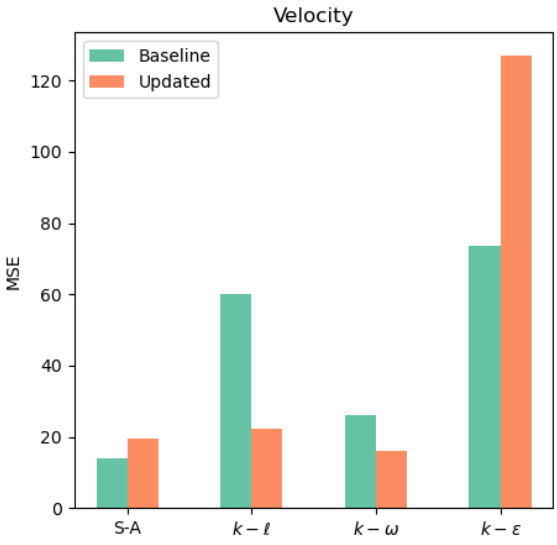
This section describes the Bayesian calibration of closure parameters for four RANS models—Spalart-Allmaras, $k-\ell$, $k-\omega$, and $k-\epsilon$—to better align their predictions with observed data, particularly the total pressure field. The calibration aimed to minimise discrepan-



(a) Total pressure

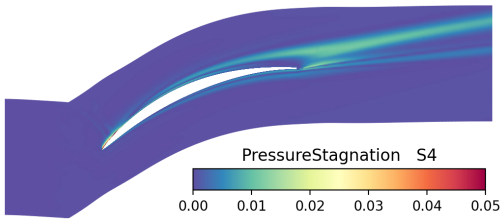


(b) Pressure

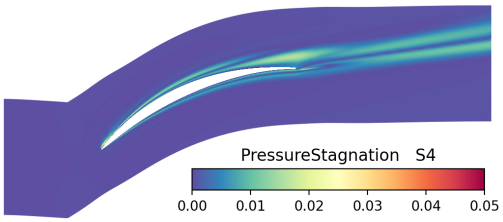


(c) Velocity

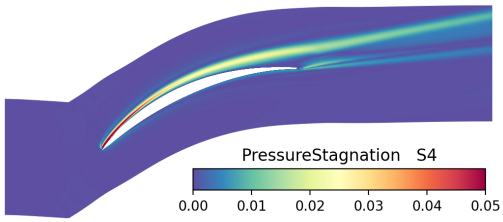
Figure 5.5 – MSE of various QoI w.r.t the reference data for the baseline and updated models on scenario S_4 after calibration on the total pressure on S_4



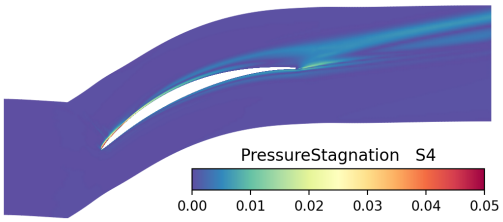
(a) Baseline S-A model



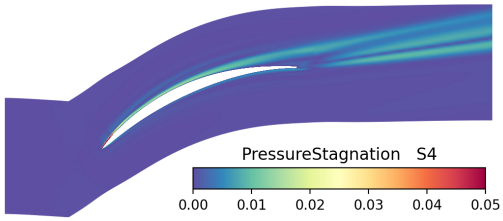
(b) Updated S-A model



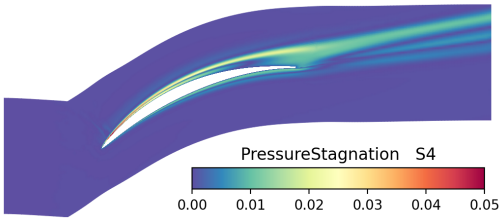
(c) Baseline $k - \ell$ model



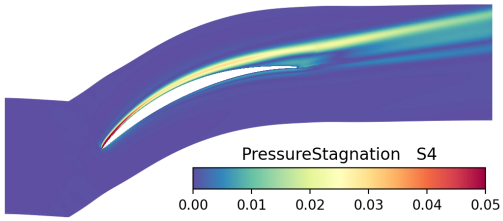
(d) Updated $k - \ell$ model



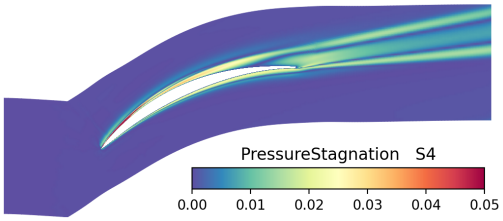
(e) Baseline $k - \omega$ model



(f) Updated $k - \omega$ model

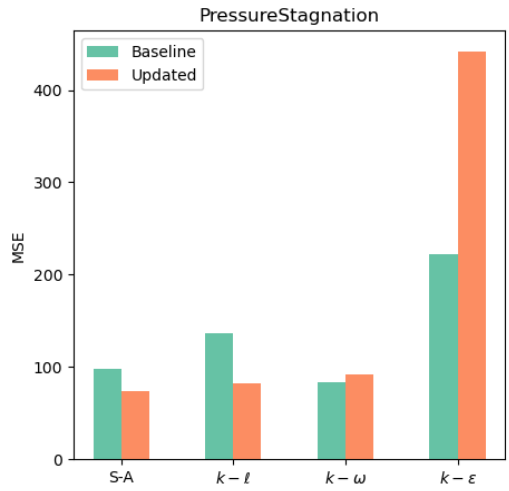


(g) Baseline $k - \varepsilon$ model

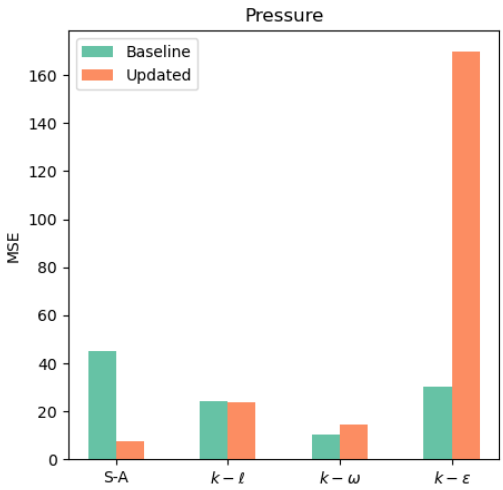


(h) Updated $k - \varepsilon$ model

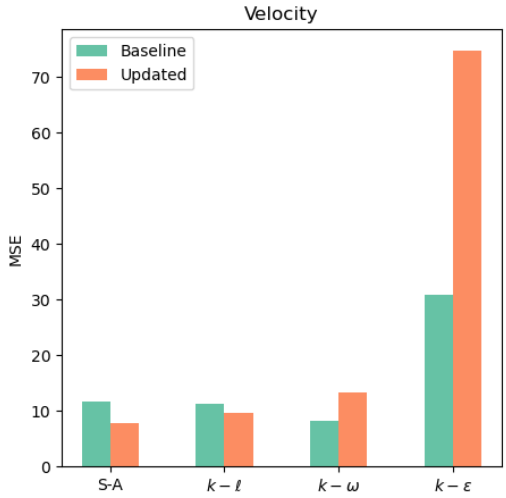
Figure 5.6 – Relative error on total pressure with regards to the reference data on S_4 for the baseline and the updated models



(a) Total pressure



(b) Pressure



(c) Velocity

Figure 5.7 – MSE of various fields w.r.t the reference data for the baseline and updated models on scenario S_3 after calibration on the total pressure field on S_4

cies between model outputs and reference data by calibrating model parameters.

Key findings include:

1. The modes of the parameter posterior distributions are not centred around the parameters baseline values indicating that the parameter updated would be beneficial to better match the reference data.
2. The modes of the posterior distributions were stable across different dataset sizes, but larger datasets led to more peaked distributions. This supports the approximation of posterior distributions as Dirac delta functions centred around the mode, indicating that the calibration process effectively captures parameter uncertainty.
3. The hyper-parameter σ_η, introduced during calibration, served as a measure of confidence in the recalibrated models. Posterior distributions of σ_η were well-defined and centred around small values, suggesting a high level of reliability in the updated models.
4. The MAP values of the calibrated parameters were used in the CFD solver for further predictions. The calibration process generally improved model performance for the calibration quantity (total pressure field). However, this improvement was uneven across spatial regions, with some areas experiencing increased local errors. Additionally, the calibration did not necessarily reduce errors for other related quantities.
5. The generalisability of the calibrated parameters was tested by applying them to an unseen scenario. Results were mixed, with some models maintaining reduced error levels, while others did not generalise well, indicating that calibration might lead to over-specialisation for specific scenarios, limiting the broader applicability of the parameters.

Overall, while the calibration improved the RANS models' agreement with the observed data, the results underscored the trade-offs involved, particularly in terms of local versus global performance and the generalisability of the calibrated parameters.

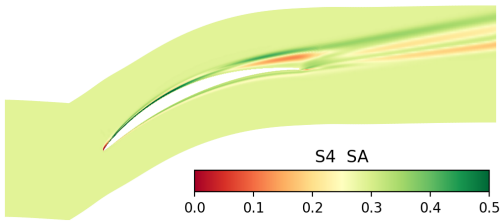
5.3 Space-dependent aggregation of models

5.3.1 Space-dependent weighting

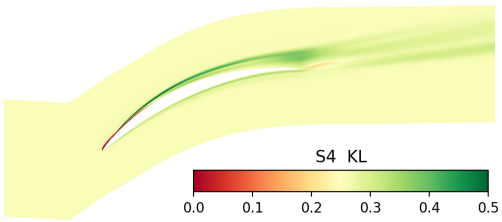
Prediction of weights using the Random Forest Regressor

The model weights defined in Equation 5.4, are regressed in the feature space composed of the features proposed by Ling and Templeton [62] using a Random Forest Regressor.

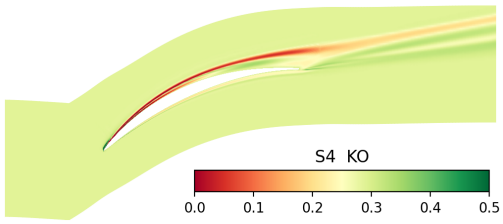
The weights obtained are displayed in Figures 5.8a to 5.8d. These figures suggest that the weights exhibit significant spatial variability, with each model being favoured in certain areas and disadvantaged in others. Notably, in the potential flow region, where all models yield the



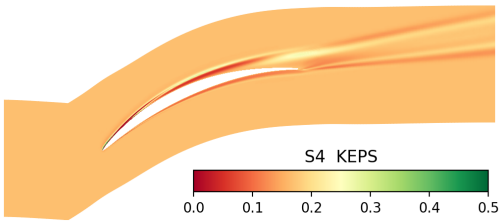
(a) Spalart-Allmaras model



(b) Smith $k - \ell$ model



(c) Wilcox $k - \omega$



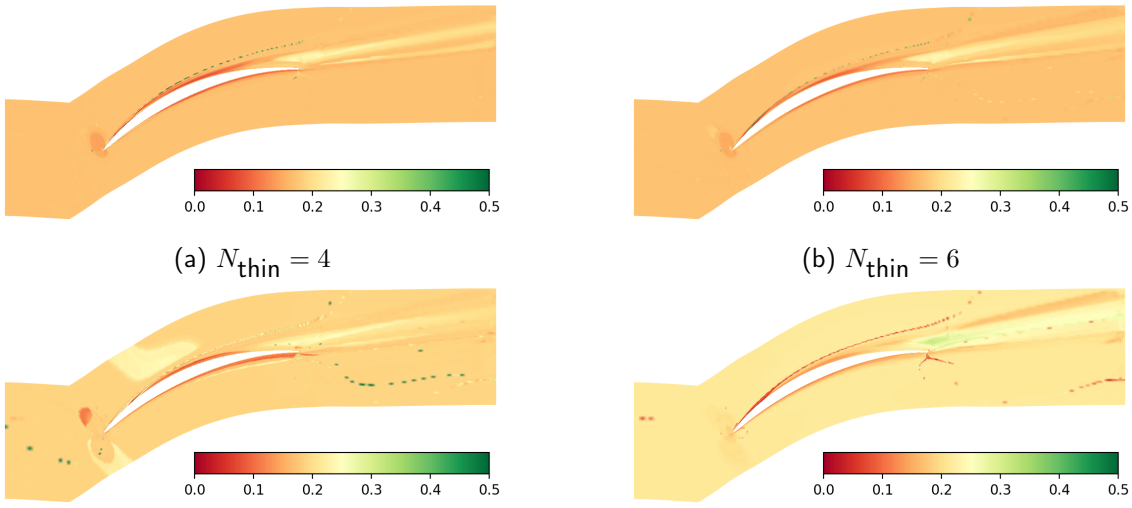
(d) Launder-Sharma $k - \epsilon$ model

Figure 5.8 – Model posterior probabilities on scenario S_4 after calibration on scenario S_4

same results due to the turbulence model not affecting the total pressure field far from the blade, the model weights are constant but differ between models. This discrepancy arises because the models have different inferred values for the hyper-parameter σ_η, which represents the confidence in modifying the model parameters to match the reference data. A larger σ_η indicates less trust in the updated model. These varying values of σ_η not only affect the weight in the potential flow but also disadvantage the model throughout the flow, suggesting that the calibration step had less confidence in this model and that it should be used with caution.

Influence of the training dataset size

It has been previously shown that in this study, the size of the training dataset does not impact the maxima a posteriori (MAP) of the parameter distributions. However, Figures 5.9a to 5.9d indicate that the interpolated weights are highly sensitive to the number of data points employed to train the Random Forest Regressor. Notably, minor interpolation artefacts are observed even with thinning numbers as small as 4 or 6. These artefacts are unlikely to significantly affect the reconstructed results. Nevertheless, they could introduce discrepancies when predicting weights in a different scenario with a varying thinning number. As the thinning number increases further, these artefacts become more pronounced and can even alter the overall trend. As a general



(a) $N_{\text{thin}} = 4$ (b) $N_{\text{thin}} = 6$

(c) $N_{\text{thin}} = 10$ (d) $N_{\text{thin}} = 14$

Figure 5.9 – $k - \epsilon$ model posterior probabilities after calibration on scenario S_4 with varying N_{thin} numbers

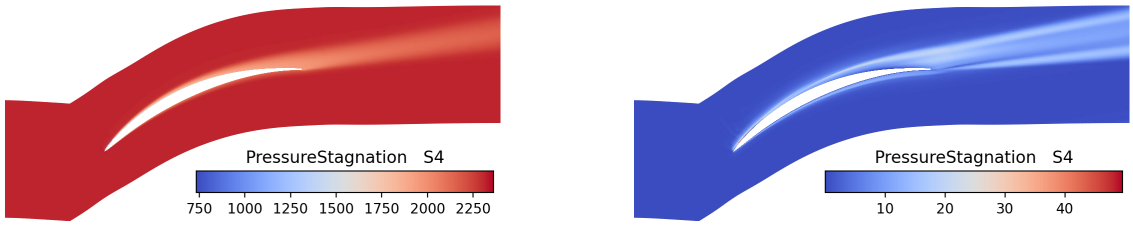
guideline, it is advisable to use the largest dataset available for interpolating the weights, while balancing this against reasonable computational costs. In the remainder of this chapter, the calibration and weighting steps will utilise a thinning number of $N_{\text{thin}} = 2$.

5.3.2 Reconstruction and predictions

Using Equations 5.6 to 5.8, it is possible to reconstruct a prediction and the variance associated with the prediction, either on the training scenario or a different one to test the

generalisability of the proposed method.

On the training scenario



(a) Expectancy (b) Standard deviation

Figure 5.10 – Expectancy and standard deviation of the reconstructed total pressure field after training and prediction on scenario S_4

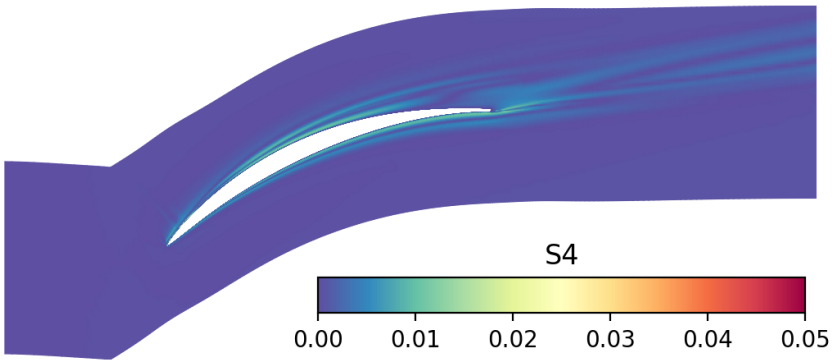
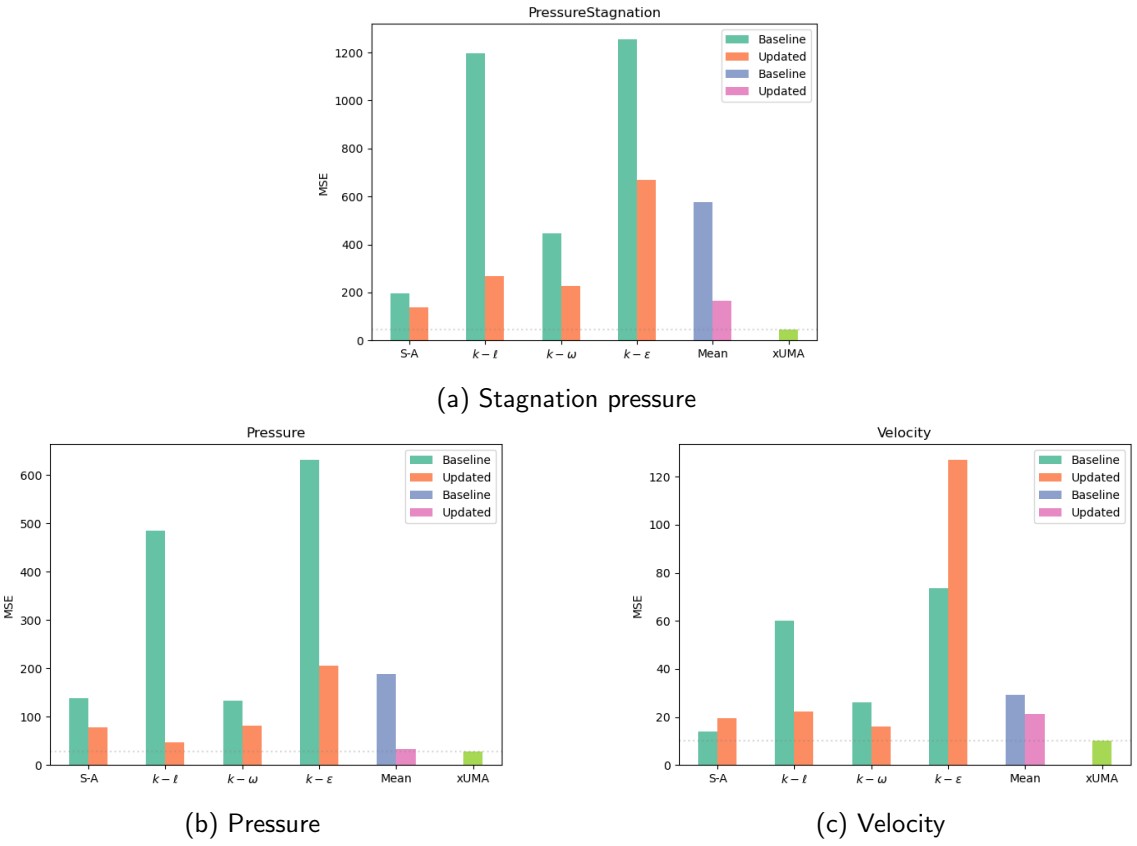


Figure 5.11 – Relative error w.r.t the observed data of the reconstructed total pressure field after training and prediction on scenario S_4

The predicted total pressure field and the variance associated with the prediction are presented in Figures 5.10a and 5.10b. The latter is particularly noteworthy as it illustrates the standard deviation among the models. It is evident that in the potential flow region, the variance is zero, indicating that all models yield the same results as expected. Outside the potential flow region, the standard deviation is not null, which is reassuring because, in these areas, the turbulence model significantly influences the resulting total pressure field. However, the relatively small variance compared to the average value of the pressure field suggests that all the calibrated

models provide predictions close to the expected value of the xUMA, indicating strong agreement among the models.

The relative error to the observed data for the same reconstructed total pressure field is depicted in Figure 5.11. This figure suggests that the error seems reduced across the flow compared to each of the individual baseline or updated models shown in the previous subsection.



(a) Stagnation pressure

(b) Pressure

(c) Velocity

Figure 5.12 – MSE of the baseline and updated models, the arithmetic mean of baseline and updated models and the space-dependent mixture of updated models after training and prediction on scenario S_4

Figure 5.12a shows the mean squared errors (MSEs) for baseline RANS models, individually updated models, the arithmetic mean of these models, and the space-dependent mixtures of updated models (XUMA) after training and predicting on scenario S_4 for the calibration and weighting QoI. The results demonstrate that the XUMA approach outperforms all individual

baseline and updated models. Not only does the XUMA mixture achieve better performance than any single model, but it also reduces uncertainty in model selection and surpasses the performance of a simple aggregation of baseline or updated models. This suggests that strategic weighting of models based on their performance in the feature space significantly enhances overall accuracy.

Figures 5.12b and 5.12c illustrate the MSEs for both baseline and updated RANS models, including their arithmetic mean and the XUMA results for quantities not seen during the offline training phase. These figures confirm that XUMA continues to deliver superior performance compared to any single baseline or updated model, even though, in some cases, the updated models may underperform relative to the baseline models for velocity fields.

On an unseen scenario

To predict a new scenario, the online prediction methodology needs to be followed. Once the RANS simulation using the updated parameters is complete, the features on the new configurations can be computed. Using those, the Random Forest Regressor (RFR) can predict the new model weights for the unseen configuration.

Figures 5.13a to 5.13g display the predicted model weights for scenario S_3 using the RFR training on scenario S_4. The consistency in the general trend of these weights compared to those obtained in the training scenario is reassuring. Given that this study employs synthetic data for validation purposes, observed data is available for scenario S_3, allowing for the computation of exact weights. These exact weights are depicted in Figures 5.13b to 5.13h.

The discrepancy between the exact model weights and the predicted weights highlights potential limitations in the Random Forest Regressor's (RFR) predictive performance. Several factors may contribute to this issue. First, the feature set used may not fully capture the spatial variability in model weights. Some features from Ling's set might be missing because they are irrelevant in a 2D configuration or because they are undefined for certain models—such as the lack of turbulent kinetic energy data for the SA model. This non-bijective relationship suggests that identical feature sets in the training data might correspond to different model weights. Adding more features could address this issue. Second, variations in physical flow processes within different scenarios can significantly affect the weights and their interpolation in the feature space. For example, scenario S_4 features a recirculation bubble near the trailing edge, which is absent in scenario S_3.

The predicted total pressure field and the variance associated with the prediction are presented in Figures 5.14a and 5.14b. The relative error to the observed data for the same reconstructed total pressure field is depicted in Figure 5.15.

These figures indicate a reduction in error compared to the error plots of the baseline and updated models discussed in the previous chapter. Figure 5.14b is particularly notable for showing the standard deviation among the models. In the potential flow region, the standard deviation remains zero, while outside this region, it is non-zero but relatively small. This low standard deviation is attributed to the highly discriminative model weights. According to equation 5.8, the variance is the sum of squared discrepancies between the global and model-specific expected values, weighted by the model weights. Models with poor performance, and thus smaller weights, contribute less to this variance. Despite this, the small variance relative to the average pressure field value suggests that the calibrated models' predictions are closely aligned with the xUMA's expected value, reflecting strong agreement among the models.

Figure 5.16a illustrates the MSEs for total pressure from the baseline and updated RANS models, the arithmetic mean of these models, and the space-dependent mixtures of updated (XUMA) models after training on scenario S_4 and predicting on scenario S_3. For reference, it also shows the MSE of the XUMA if the predicted weights were replaced with the exact weights derived from observed data for scenario S_3. The results reveal that the updated models do not uniformly outperform the baseline models, as demonstrated by the deterioration in the $k-\omega$ model's performance.

The space-dependent aggregation of models (XUMA) for the unseen scenario S_3 still performs better than the best-performing updated model, though it falls short compared to XUMA using true weights from the observed data. This underscores that discrepancies between predicted and exact weights significantly impact XUMA performance, and maintaining consistent physical processes between training and prediction scenarios would likely enhance prediction accuracy.

Importantly, the space-dependent model mixture outperforms a naive mixture based on the arithmetic mean of all models. This improvement is due to the weights being influenced by the hyper-parameter σ_η, which reflects the reliability of each updated model. For instance, the significant performance drop of the $k-\varepsilon$ model does not heavily affect the overall mixture performance due to its lower trust level, as indicated by the higher σ_η value assigned during calibration.

Figures 5.16b to 5.16c presents the MSEs of the baseline and updated RANS models, the arithmetic mean of baseline and updated models, and the space-dependent mixtures of updated

(XUMA) models after training on scenario S_4 and predicting on scenario S_3 for various quantities. For comparison, the figures also display the MSE of the XUMA if the predicted weights were the "true" weights computed using the observed data on scenario S_3.

The XUMA performs less or as well as the best-performing updated model and is still less effective than XUMA using true weight values computed from the observed data available in the prediction scenario. Also, it can be noted that the XUMA using the "true" weights computed using the observed data on the prediction scenario performs as well as the best-updated model.

5.3.3 Section summary

In this section, the space-dependent aggregation of the previously updated models was investigated to enhance the predictive performance of quantities in both training and unseen scenarios. The approach combined Bayesian-inspired model weighting with a Random Forest Regressor (RFR) to interpolate these weights, resulting in several key insights:

1. The study demonstrated that space-dependent weights, computed from model plausibilities, effectively discriminate between models. This discrimination occurs globally, via the hyper-parameter σ_η that indicates overall model trustworthiness, and locally, as weights vary spatially to reflect different model performances across the flow.
2. The RFR's capability to manage extensive datasets and capture complex, non-linear relationships within the data has been crucial for accurately predicting model weights across scenarios. However, the predictive capabilities of the RFR appeared limited, as the weights predicted for an unseen scenario can differ from the true values obtained using the observed data. Adding more informative features is an interesting perspective to ensure that the relationship between the weights and the features is well captured and is as bijective as possible to ensure the best possible prediction of weight and the best possible predictive performances.
3. The reliability of the interpolated weights was strongly dependent on the size of the training dataset. Larger datasets led to more accurate weight predictions, highlighting the need to balance dataset size with computational resources to achieve optimal results.
4. The aggregated XUMA approach consistently outperformed individual models and naive model averaging by effectively reducing both bias and variance in predictions. However, while XUMA performed well in unseen scenarios, it was less effective than using true weights

derived from observed data, underscoring the importance of accurate weight interpolation between training and prediction scenarios.

In conclusion, the space-dependent aggregation of updated models offers a means of enhancing the reliability and accuracy of CFD predictions, particularly when combined with the calibration step. However, the performance of the RFR for weight prediction in unseen scenarios suggests that further refinement is necessary. Future research should explore additional machine learning methodologies, improve feature sets, or develop new frameworks for computing weights to further advance the predictive capabilities of aggregated CFD models.

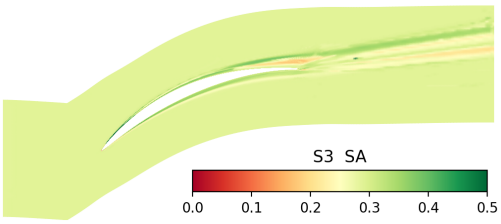
5.4 Chapter summary

In this chapter, we developed a new methodology for aggregating RANS models by first calibrating them before aggregation. This method, inspired by the XMA algorithm, aimed to improve the accuracy of predictions for the NACA65 V103 linear cascade. We calibrated four RANS models—Spalart-Allmaras, $k-\omega$, $k-\ell$, and $k-\epsilon$—to observed data and then combined these models using space-dependent weights.

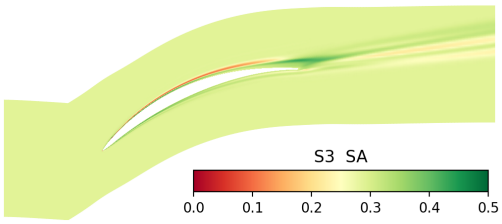
The methodology included three main steps: offline calibration, weight regression using a Random Forest Regressor (RFR), and online prediction. The offline calibration used Bayesian methods to adjust model parameters, improving their likelihood with observed data and enhancing the prediction of the total pressure field. This process was crucial for reducing discrepancies between model outputs and reference data. However, the calibration showed some trade-offs, such as potential over-specialisation to specific scenarios, which could limit the broader applicability of the parameters. Although the calibration improved accuracy for the specific quantity, performance varied across different areas and did not always reduce errors for other related quantities.

During the aggregation phase, the use of space-dependent weights—calculated from model plausibilities—was shown to improve predictive performance by reducing bias and variance compared to using individual models or simple averaging. The RFR was effective in managing the complexities of weight interpolation. However, its performance was limited for new scenarios, highlighting the need for better weight prediction methods. Larger training datasets led to more accurate predictions, but balancing dataset size with computational resources can be challenging in more complex configurations. The aggregated approach, while better than individual models, was significantly less effective on an unseen configuration than using weights derived directly from the available observed data, emphasising the importance of accurate weight interpolation.

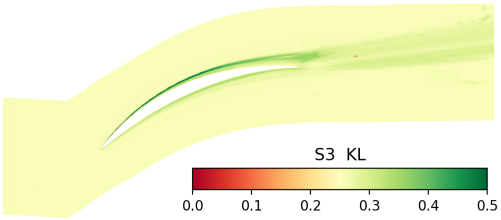
Overall, this chapter demonstrates that integrating calibration with model aggregation can significantly enhance CFD predictions. By refining RANS models through Bayesian calibration and combining them with space-dependent weights, we achieved better results compared to traditional methods. However, the aggregation technique using RFR revealed limitations in predicting weights for new scenarios. Future research should focus on developing advanced machine learning methods, expanding feature sets, and exploring new frameworks for weight computation. These improvements will be crucial for overcoming challenges in new scenarios and advancing the predictive performance of aggregated CFD models.



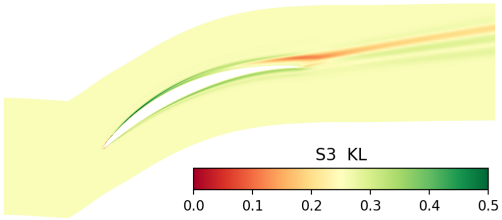
(a) Spalart-Allmaras model



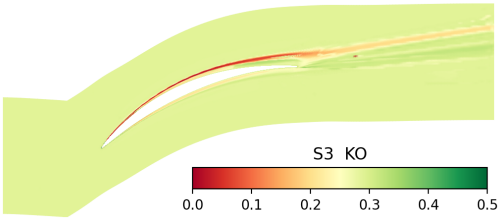
(b) Spalart-Allmaras model



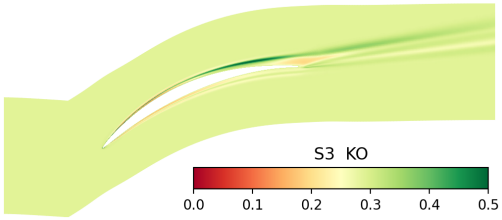
(c) Smith $k - \ell$ model



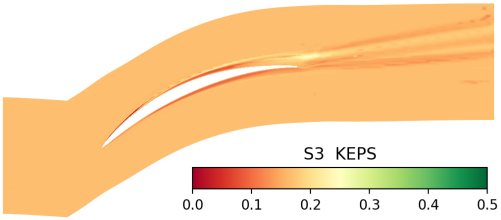
(d) Smith $k - \ell$ model



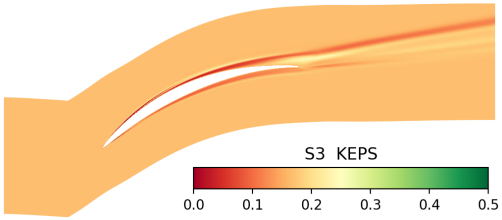
(e) Wilcox $k - \omega$ model



(f) Wilcox $k - \omega$ model

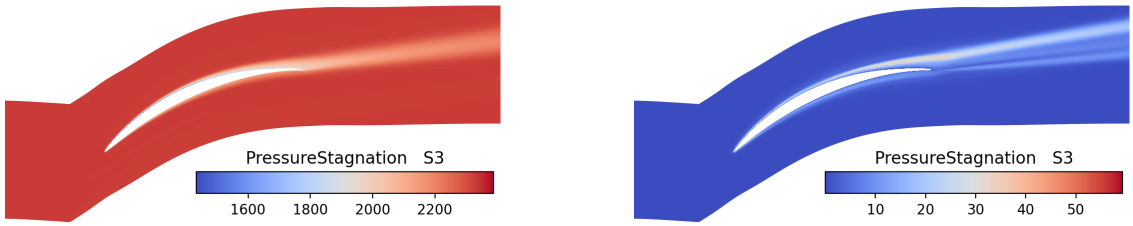


(g) Launder-Sharma $k - \epsilon$ model



(h) Launder-Sharma $k - \epsilon$ model

Figure 5.13 – Predicted (left panel) weights using the RFR on scenario S_3 after calibration on scenario S_4 and and exact (right panel) on scenario S_3 after calibration on scenario S_3.



(a) Expectancy (b) Standard deviation

Figure 5.14 – Expectancy and standard deviation of the reconstructed total pressure field after training on scenario S_4 and prediction on scenario S_3

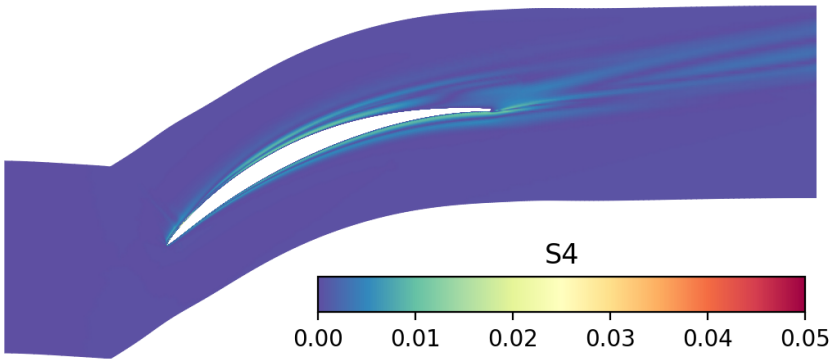
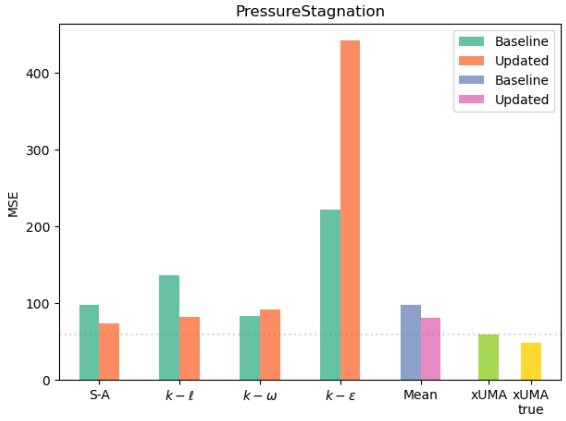
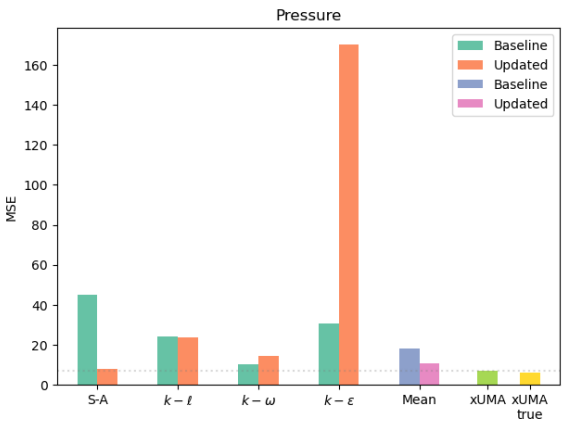


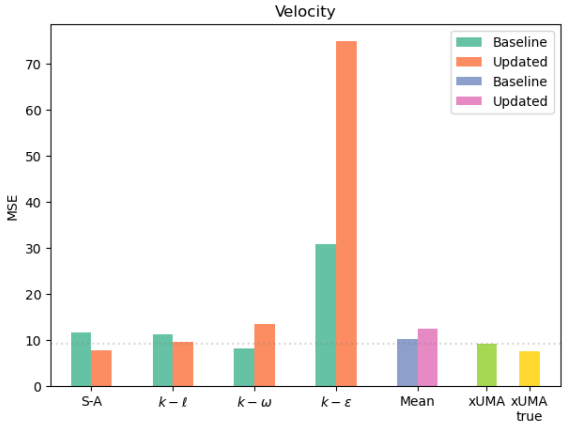
Figure 5.15 – Relative error of the reconstructed total pressure field after training on scenario S_4 and prediction on scenario S_3



(a) Stagnation pressure



(b) Pressure



(c) Velocity

Figure 5.16 – MSE of the baseline and updated models, of the arithmetic mean of baseline and updated models and of the space-dependent mixture of updated models after training on scenario S_4 and prediction on scenario S_3 w.r.t the reference data

6

Clustered model aggregation

This chapter presents a novel model calibration approach designed to address the limitations of global calibration on full-field data (discussed in Chapter 5), as well as an alternative space-dependent model aggregation approach. Recognising that the over-specialisation of models can deteriorate performance in certain flow regions, the proposed methodology leverages clustering techniques to calibrate models within clusters corresponding to similar physical dynamics, thereby creating specialised models on purpose. By generating predictions restricted to specific physical dynamics and subsequently aggregating these specialised predictions, the method aims to reconstruct a comprehensive global prediction. Implementing fuzzy clustering ensures smooth transitions between clusters, preserving the continuity of physical fields while improving overall prediction accuracy. This model aggregation approach significantly reduces errors compared to baseline RANS models and demonstrates superior generalisation across varied scenarios than the previously studied space-dependent aggregation methods.

Chapter contents

6.1 Methodology

The XUMA method presented in the preceding chapter relies on calibrating the RANS model parameters on a full field, namely, the total pressure dataset. As the previous results suggest, calibrating the whole field can compromise performance in certain areas while attempting to improve them elsewhere. Furthermore, the inferred posterior parameters do not generalise well across all considered scenarios. Using model aggregation mitigates these issues by leveraging the best-performing models according to local flow features. Still, results for a new scenario do not always perform better than the baseline models, indicating that the model is prone to overfitting.

The Clustered Model Aggregation (CMA) method proposed in this section builds on the previous approach. Still, it employs unsupervised machine learning to cluster the training data

before calibration, aiming to identify regions corresponding to different physical dynamics, which may subsequently require distinct modelling. The proposed methodology relies on calibrating the concurrent RANS models on data extracted from regions representative of dominant physical dynamics present in the flow.

6.1.1 Clustered Model Aggregation

Consider a set $\mathcal{M} = \{M_1, \ldots, M_m, \ldots, M_{N_M}\}$ of N_M models, each characterised by a vector of parameters $\boldsymbol{\theta}_m$, and consider a vector of observed data Δ_{obs}. We assume that subsets of the data are assigned to N_C clusters, according to some probability distribution. As a consequence, $\Delta_{obs} = \cup_{c=1}^{N_C} \Delta_{obs,c}$, where $\Delta_{obs,c}$ refers to the data belonging to the c-th cluster.

The data $\Delta_{obs,c}$ in each cluster $C_{c\in[1,N_C]}$ are used to calibrate the N_M competing models by using the Bayesian framework presented in Chapter 3, leading to $N_M \times N_C$ updated models. The latter can be used to estimate the posterior predictive distribution of a new quantity of interest for the same flow or an unseen flow. As done previously, to reduce the computational cost of propagating the full posterior distributions through the CFD solver, we approximate the posteriors as multidimensional Dirac distribution, centred at the joint Maximum A Posteriori (MAP) estimate $\boldsymbol{\theta}^*_{m,c}$ of the parameter vector:

$$p(\boldsymbol{\theta}_{m,c}|\Delta_{obs,c}, M_m) \approx \delta(\boldsymbol{\theta}_{m,c} - \boldsymbol{\theta}^*_{m,c}) \tag{6.1}$$

The philosophy behind the CMA is to create for each cluster an *expert* hyper-model specialised at capturing the flow dynamics within the cluster. This cluster-specific hyper-model aggregates the component models calibrated within the cluster. The prediction of the cluster hyper-model may not perform well outside of the training cluster but is not supposed to be used outside of it. As the clusters can be considered as physically coherent entities it makes sense to construct a weighting method in which the weights are constant throughout the cluster.

Based on the definition of local weights, previously introduced in chapter 5, section 5.1, from the posterior model plausibility within a cluster :

$$p(M_m|d_{c,i}) \propto p(d_{c,i}|M_m) \tag{6.2}$$

where $d_{c,i} \in \Delta_{obs,c}$ is an observation belonging to cluster C_c and $p(d_{c,i}|M_m)$ is the marginal likelihood of the observation $d_{c,i}$ under model M_m, the cluster hyper-model weights can be

defined as the average of those local weights. For model M_m in cluster C_c, the weight is defined as :

$$W_{m,c} = \frac{\sum_{i=1}^{N_{D,C}} p(M_m|d_{c,i})p(d_{c,i})}{\sum_{k=1}^{N_M}\sum_{i=1}^{N_{D,C}} p(M_m|d_{c,i})p(d_{c,i})} \tag{6.3}$$

where $p(M_m \mid d_{c,i})$ is the local model posterior plausibility and $p(d_{i,c})$ is the probability of observation d_i belonging to cluster C_c. The latter probability is obtained as an output of the clustering algorithm. The proposed weighting method allows the construction for each cluster of a hyper-model prediction defined as the aggregation of the models updated against the observed data from that cluster weighted by their respective weights. The definition of constant weights throughout the clusters has multiple advantages. The first one is that the scalar definition of in-cluster weights suppresses the need for the regression of weights, as the weights are defined for one model in one cluster, simplifying drastically the implementation of the the CMA framework. The second mentioned earlier, is to make sure of the physical coherence of the clustering step in the weighting step.

Once the models are updated and their respective weights computed, the following prediction methodology can be applied, on the same or an unseen configuration, to predict the calibration quantity or an unseen quantity.

Be $\Delta^* = \Delta^*(\boldsymbol{\eta}^*)$ a QoI to be predicted at an unseen location $\boldsymbol{\eta}^*$ of the covariate space. The $N_M \times N_C$ updated models are first propagated through the CFD solver to estimate the posterior predictive distributions of Δ^* given the model M_m and the data used for calibration within a cluster, $\Delta_{obs,c}$: $\quad p(\Delta^*|\Delta_{obs,c}, M_m, \boldsymbol{\eta}^*)$.

The clustered model aggregation prediction is defined as :

$$p(\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*) = \sum_{c=0}^{N_C} p(\Delta^*|\Delta_{obs,c}, \boldsymbol{\eta}^*)p(C_c|\boldsymbol{\eta}^*) \tag{6.4}$$

where $p(\Delta^*|\Delta_{obs,c}, \boldsymbol{\eta}^*)$ is the cluster hyper-model prediction, and $p(C_c|\boldsymbol{\eta}^*)$ is the cluster plausibility at the location $\boldsymbol{\eta}^*$ of the covariate space, i.e. the probability of the location $\boldsymbol{\eta}^*$ to belong to cluster C_c. The covariate space used for the clustering is based on a set of flow features introduced later in Section 6.2. The in-cluster prediction is estimated as the average of each updated model posterior predictive distribution weighted by their in-cluster weights. For cluster

C_c, this is defined as :

$$p(\Delta^*|\Delta_{obs,c}, \boldsymbol{\eta}^*) = \sum_{i=0}^{N_M} p(\Delta^*|M_m, \Delta_{obs,c}, \boldsymbol{\eta}^*) W_{m,c} \tag{6.5}$$

Note that the in-cluster model weight here is a scalar and does not depend on the location $\boldsymbol{\eta}^*$ of the covariate space, but only on the model and the cluster considered.

Finally, the cluster plausibility represents the probability of a prediction location $\boldsymbol{\eta}^*$ to belong to a given cluster. For the training environment, such probability is obtained as an output of the clustering algorithm applied to the observed data. For a prediction environment, however, no data is available and the clustering must rely on the model predictions. To avoid an implicit procedure, several choices are possible. Hereafter we use flow field predictions given by the N_M baseline models to generate the N_C clusters for the prediction case. Because the model predictions may differ, the discovered clusters are not the same, but they are expected to be still representative of a given physical dynamics seen at training. Given cluster C_c we then get a different cluster probability $p(C_c|M_m, \boldsymbol{\eta}^*)$, which is weighted by the model prior plausibility $p(M_m)$. For *a priori* equi-plausible models with $p(M_m) = 1/N_M$:

$$p(C_c|\boldsymbol{\eta}^*) = \sum_{m=0}^{N_M} p(C_c|M_m, \boldsymbol{\eta}^*) p(M_m) = \frac{1}{N_M} \sum_{m=0}^{N_M} p(C_c|M_m, \boldsymbol{\eta}^*). \tag{6.6}$$

This formulation accounts for the discrepancies in cluster predictions associated with model-form uncertainties and for the unavailability of observed data on the prediction scenario to discover the clusters.

Based on the posterior predictive distribution in equation 6.4, the expectancy and variance can be expressed as follows :

$$\mathbb{E}\left[\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*\right] = \sum_{c=0}^{N_C} \mathbb{E}\left[\Delta^* \mid \Delta_{obs,c}, \boldsymbol{\eta}^*\right] p(C_c|\boldsymbol{\eta}^*). \tag{6.7}$$

Expanding $\mathbb{E}\left[\Delta^* \mid \Delta_{obs,c}, \boldsymbol{\eta}^*\right]$, we get

$$\mathbb{E}\left[\Delta^* \mid \Delta_{obs,c}, \boldsymbol{\eta}^*\right] = \sum_{m=0}^{N_M} \mathbb{E}\left[\Delta^* \mid M_m, \Delta_{obs,c}, \boldsymbol{\eta}^*\right] W_{m,c}. \tag{6.8}$$

Thus, the overall expectation becomes:

$$\mathbb{E}[\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*] = \sum_{c=0}^{N_C} p(C_c|\boldsymbol{\eta}^*) \sum_{m=0}^{N_M} \mathbb{E}\left[\Delta^* \mid M_m, \Delta_{obs,c}, \boldsymbol{\eta}^*\right] W_{m,c}. \tag{6.9}$$

The total variance is given by the law of total variance:

$$\begin{aligned} \mathrm{Var}(\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*) = &\sum_{c=0}^{N_C} p(C_k|\boldsymbol{\eta}^*)\, \mathrm{Var}(\Delta^* \mid \Delta_{obs,c}, \boldsymbol{\eta}^*) \\ &+ \sum_{c=0}^{N_C} p(C_k|\boldsymbol{\eta}^*) \left(\mathbb{E}[\Delta^* \mid \Delta_{obs,c}, \boldsymbol{\eta}^*] - \mathbb{E}[\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*]\right)^2 \end{aligned} \tag{6.10}$$

Expanding the in-cluster variance $\mathrm{Var}(\Delta \mid C_k)$:

$$\begin{aligned} \mathrm{Var}(\Delta^* \mid \Delta_{obs,c}, \boldsymbol{\eta}^*) = &\sum_{m=0}^{N_M} W_{m,c}\, \mathrm{Var}(\Delta^* \mid M_m, \Delta_{obs,c}, \boldsymbol{\eta}^*) \\ &+ \sum_{m=0}^{N_M} W_{m,c} \left(\mathbb{E}[\Delta^* \mid M_m, \Delta_{obs,c}, \boldsymbol{\eta}^*] - \mathbb{E}[\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*]\right)^2 \end{aligned} \tag{6.11}$$

However, using the MAP approximates rather than the full posterior distribution implies a null within model variance, i.e. $\mathrm{Var}(\Delta^* \mid M_m, \Delta_{obs,c}, \boldsymbol{\eta}^*) = 0$ Thus, the total variance becomes:

$$\begin{aligned} \mathrm{Var}(\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*) = &\sum_{c=0}^{N_C} p(C_c|\boldsymbol{\eta}^*) \sum_{m=0}^{N_M} W_{m,c} \left(\mathbb{E}[\Delta^* \mid M_m, \Delta_{obs,c}, \boldsymbol{\eta}^*] - \mathbb{E}[\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*]\right)^2 \\ &+ \sum_{c=0}^{N_C} p(C_c|\boldsymbol{\eta}^*) \left(\mathbb{E}[\Delta^* \mid \Delta_{obs,c}, \boldsymbol{\eta}^*] - \mathbb{E}[\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*]\right)^2 \end{aligned} \tag{6.12}$$

In the next section, the preceding framework is applied to the NACA65-V103 compressor cascade presented in Chapter 4.2.1.

6.1.2 Summary of the method

In summary, the proposed methodology consists of the following steps:

1. offline identification of N_C regions (clusters) corresponding to the dominant physical dynamics by using a clustering algorithm to a covariate space described by a set of flow

features $\boldsymbol{\eta}$;

2. in-cluster model calibration and estimation of weights corresponding to cluster probabilities (offline);
   - — **Calibrate** a set of N_M models, characterised by parameters $\boldsymbol{\theta}_m$ against data $\Delta_{obs,c}$ observed within the c-th cluster using the Bayesian framework, and **approximate** the posterior distributions as Dirac distributions centred at the MAP estimate $\boldsymbol{\theta}^*_{m,c}$. This leads to $N_C \times N_M$ updated models.
   - — **Compute** the in-cluster weights defined as the average of the posterior model plausibility. They are defined as $W_{m,c} \propto \sum_{i=1}^{N_{D,C}} p(M_m|d_{c,i})p(d_{c,i})$.
3. Online prediction of a new case by cluster-dependent aggregation of specialised models.
   - — **Predict** the quantity of interest $\Delta^* = \Delta^*(\boldsymbol{\eta}^*)$ by propagating the MAP estimates of the parameters $\boldsymbol{\theta}^*_{m,c}$ through the $N_M \times N_C$ updated models.
   - — Predict the features $\boldsymbol{\eta}$ throughout the flow domain by using the N_M baseline models
   - — **Apply** the clustering algorithm to the N_M baseline features and **compute** the cluster probabilities defined as $p(C_c|\boldsymbol{\eta}^*) = \sum_{m=0}^{N_M} p(C_c|M_m, \eta^*)p(M_m)$
   - — **Compute** the in-cluster prediction defined as $p(\Delta^*|\Delta_{obs,c}, \boldsymbol{\eta}^*) = \sum_{i=0}^{N_M} p(\Delta^*|M_m, \Delta_{obs,c}, \boldsymbol{\eta}^*)W_{m,c}$
   - — **Aggregate** the N_C in-cluster prediction to compute the predictive posterior distribution: $p(\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*) = \sum_{c=0}^{N_C} p(\Delta^*|\Delta_{obs,c}, \boldsymbol{\eta}^*)p(C_c|\boldsymbol{\eta}^*)$. From this posterior predictive distribution, the expectancy and variance can be computed.

In the following, we discuss the keystone of the algorithm, i.e. the generation of clusters, and we propose and assess a clustering method to identify fluid flow regions. Then, we apply the clustering strategy to the flow through the NACA65 cascade configuration, and we use the clustered data for the local model updating. Finally, we use the CMA aggregation approach to formulate predictions of various quantities of interest for the training flow scenario, and for an unseen scenario.

6.2 Automatic detection of flow regions

With the large volume of data generated by CFD simulations, researchers are increasingly turning to Machine Learning (ML) techniques to assist them in post-processing and analysis. An interesting task consists of using ML for detecting automatically flow regions characterised by common physical dynamics [14, 43]. These detected regions can then be used for various purposes, including interpreting the underlying physical mechanisms, constructing surrogate models, detecting regions in need of grid refinement (mesh adaptation), or applying different equations in each region (e.g., distinguishing between near-wall turbulent regions and inviscid far fields).

There is substantial potential in using ML techniques for identifying flow regions, which is the primary focus of this study. Employing machine learning for region detection typically involves classification or clustering methods, which facilitate data organisation by forming clusters of points or elements with similar characteristics or values. The key challenge is determining the boundaries of these clusters, which differentiate the classified or clustered data. Classification algorithms (e.g., Logistic Regression, K-Nearest Neighbours, Decision Trees, Support Vector Machines) fall under the supervised learning paradigm [5, 80], requiring pre-labelled datasets to guide and train the algorithm. In contrast, clustering methods (e.g., Gaussian mixture models, K-means, MiniBatch K-Means, Mean Shift, Spectral Clustering) [12] are part of unsupervised learning and do not require pre-labelled data, enabling the discovery of data groupings.

The main advantage of using ML for flow region classification or clustering is its ability to improve over traditional methods and to accommodate multiple inputs. Traditional methods depend heavily on manually selected thresholds [109], and selecting a threshold can often be based on a trial-and-error process in which the final choice is strongly based on expert judgement. The automatic detection of flow regions by ML is raising more and more interest as studies already seem to reveal very promising results, either by using supervised or unsupervised learning. For instance, Li *et al.* [60] utilised a supervised ML classification method to identify turbulent and non-turbulent regions in flow past a circular cylinder at various turbulent Reynolds numbers, using inputs such as kinetic energy and vorticity to train an extreme gradient boosting XGBoost classifier. Beck *et al.* [6] developed a new indicator for the detection of shocks and strong discontinuities using supervised learning and convolutional neural networks for shock-capturing applications. Colvert *et al.* [18] used neural networks to classify various vortex wake types that have been trained using local vorticity data.

Recent studies have also highlighted the potential of unsupervised ML in this domain. For instance, Wu *et al.* [109] used an unsupervised self-organising map machine learning algorithm, as an automatic classifier to separate a transitional boundary layer in turbulent and non-turbulent regions. Otmany *et al.*[78] introduced a clustering methodology for detecting viscous and turbulent regions, based on the premise that these flows exhibit distinct characteristics, with viscous dissipation and vorticity prevalent in the viscous and turbulent regions, while the outer flow remains inviscid and irrotational. Callaham *et al.*[15] proposed learning dominant physical processes described by various differential equations by using features derived from the governing equations themselves. The approach was applied to a turbulent boundary layer dataset to detect different flow regions such as the free stream, laminar inflow, transitional region, viscous sublayer, and inertial sublayer.

While promising, the above-mentioned methods have limitations. Otmany *et al.*'s approach does not differentiate various turbulent flow types within the turbulent cluster, and Callaham *et al.*'s method, while effective in a turbulent flat plate boundary layer, lacks generalisability to more complex flows, such as flows with separated regions or curvature effects because the proposed features are dimensional and frame-dependent, i.e, the clustering algorithm trained on the considered set of data does not apply to the same data represented in a different reference frame.

In this work, we aim to distinguish not only turbulent and non-turbulent regions or viscous-dominated regions from outer inviscid regions but also different kinds of turbulent flows (e.g. attached or separated boundary layers). For that purpose, we need first of all to define a suitable set of features, which can be applied to moderately complex geometries and are, to some extent, frame-independent.

In the following, we introduce a set of features that can be used to identify dominant flow dynamics using unsupervised clustering algorithms.

The feature set is an extension of the approach initially introduced in [15]. Unlike the original work, the features are non-dimensional, Galilean-invariant, and applicable to non-Cartesian systems of coordinates. The features are fed to a Gaussian Mixture Model clustering algorithm, which outputs a set of smooth cluster probability distributions. The proposed clustering methodology is tested on different flow scenarios, including the zero pressure gradient turbulent boundary layer, the backwards-facing step, and the 2D NACA65 v103 compressor cascade configurations presented in Chapter 4. The results of the clustering algorithm are highly dependent on the chosen set of features.

6.2.1 Methodology

The clustering methodology requires i) the definition of a suitable set of flow features to discriminate different physical processes and ii) a clustering algorithm.

Feature set

We propose an ML framework to discriminate nearly parallel viscous and turbulent flow regions (e.g. attached boundary layers and wakes) from inviscid flow regions (e.g. free-stream and irrotational flow regions) and separated flow regions. The methodology is based on the assumption that the different physical dynamics present in the flow are characterised by different preponderant flow figures. For example, the pressure gradient along a streamline distinguishes a favourable or adverse pressure gradient turbulent boundary layer from a zero-pressure-gradient flat plate boundary layer or a wake. The chosen flow features must then be as representative as possible of the local flow physics.

Several sets of features have been proposed in the literature, mostly for the representation of data-driven turbulence models. The one proposed by Ling and Templeton [63], used in the previous chapter to interpolate space-dependent weights, is probably one of the most widely used in the literature. Such features were initially introduced to detect regions of potential RANS modelling inadequacies, and not to discriminate flow regions. However, they are effective for data-driven turbulence modelling and have been widely employed in the literature. Note that such features are not all frame-invariant. This is for instance the case of features η_2 (the velocity magnitude is used as a normalising factor), η_4, η_7, η_8 and η_9.

To discriminate dominant physical processes in flows instead of modelling turbulence, we introduce a new set of features by building upon Callaham 's [15] equation space, where each feature corresponds to one of the terms in the governing equation. In [15], the notion of a dominant balance regime is also introduced. The latter corresponds to a region in which the flow can be described by a subset of the original terms of the governing equation. This is used to explore the equation space and discover different physical dynamics in a transitional boundary layer using the Reynolds Averaged Navier Stokes equation as the governing equation of the equation space. For an incompressible flow, the state variables are the velocity vector $\mathbf{u} = (u, v, w)$ and the pressure p, with the fluid parameterised by a constant density ρ and viscosity ν. After performing the Reynolds decomposition of the variables into mean and fluctuating components, e.g., $u(\mathbf{x}, t) = \overline{u}(\mathbf{x}) + u^{'}(\mathbf{x}, t)$, the mean flow is determined by the

Reynolds-averaged Navier-Stokes equations. In a Cartesian coordinate system aligned with the flat plate, the stream-wise momentum equation is

$$\bar{u}\frac{\partial \bar{u}}{\partial x} + \bar{v}\frac{\partial \bar{u}}{\partial y} = \rho^{-1}\frac{\partial \bar{p}}{\partial x} + \nu\nabla^2\bar{u} - \frac{\partial}{\partial y}\overline{u'v'} - \frac{\partial}{\partial x}\overline{u'^2} \tag{6.13}$$

The features used in the clustering algorithm of [15] are simply the terms in equation 6.13. Such features are dimensional and frame-dependent. Furthermore, they account only for the x-component of momentum balance. If the reference frame is rotated so that the x coordinate is now orthogonal to the wall, the terms in the x-momentum equation would all be negligible compared to the pressure derivative, resulting in an incorrect identification of the boundary layer.

Intending to develop a more general set of coordinates with better invariance properties, we consider the RANS momentum equation in the vector form:

$$\rho\frac{D\mathbf{u}}{Dt} = -\nabla p + \nabla\cdot\boldsymbol{\tau_{tot}} \tag{6.14}$$

where we omitted the averaging symbols for the sake of simplicity. In equation (6.14), $\mathbf{u}$ is therefore the mean velocity, ρ the mean density, p the mean pressure, and $\nabla\cdot\boldsymbol{\tau}_{tot} = \nabla\cdot\boldsymbol{\tau} + \nabla\cdot\boldsymbol{\tau}^R$ the viscous forces. Furthermore, $D\bullet/Dt$ denotes the Galilean-invariant material derivative. The pressure gradient and viscous terms are also Galilean-invariant.

To account for non-Cartesian flow domains, Equation (6.14) is projected onto a local reference frame associated with the local flow direction. Restricting our attention to 2D flows, we consider a frame contained in the (x, y) plane and defined by the velocity direction and a unit vector perpendicular to it and forming a levogyre system $(\boldsymbol{e}_u, \boldsymbol{e}_\perp)$. More specifically, we project the equation along the local streamwise direction $\boldsymbol{e}_u$ and we consider as features the magnitudes of all the projected terms. The latter corresponds then to the magnitudes of the streamwise inertia, pressure, and viscous forces. For high Reynolds flows, the magnitude of the latter is expected to be negligible compared to the other terms in the potential flow regions, comparable to the other terms or dominant in viscosity-dominated regions such as shear layers. To apply the trained cluster probability to a different flow, we wish to deal with non-dimensional features. The latter could be extracted directly from the non-dimensional RANS equations, obtained by defining suitable length, mass, and time scales. However, the resulting terms would no longer be Galilean invariant, because the reference velocity by which they are normalised is not. As an alternative, we divide each term by a local normalisation factor A corresponding to the sum of

the absolute values of all terms, which are Galilean invariant. This results in the following set of non-dimensional features:

$$\eta_1 = \left| \frac{\rho \frac{D\mathbf{u}}{Dt}}{A} \cdot \boldsymbol{e}_u \right|, \quad \eta_2 = \left| \frac{\nabla p}{A} \cdot \boldsymbol{e}_u \right|, \quad \eta_3 = \left| \frac{\nabla \cdot \boldsymbol{\tau_{tot}}}{A} \cdot \boldsymbol{e}_u \right|, \quad \eta_4 = \boldsymbol{e}_u \cdot \boldsymbol{e}_{u,in} \tag{6.15}$$

with:

$$A = \left| \rho \frac{D\mathbf{u}}{Dt} \cdot \boldsymbol{e}_u \right| + |(\nabla p) \cdot \boldsymbol{e}_u| + |(\nabla \cdot \boldsymbol{\tau_{TOT}}) \cdot \boldsymbol{e}_u| \tag{6.16}$$

Using the magnitudes of the various terms also prevents the normalising factor from being zero, except in the case of a uniform flow. From a numerical point of view, a small positive constant is added to A to prevent division by zero, leading the first three features to be null. Because we use the absolute values of the various terms in the equations, we cannot distinguish adverse from favourable pressure gradients or detect flow reversal. To detect the latter one needs to define some reference flow direction (e.g. of the inlet velocity). Thus, we introduce the additional feature η_4 in equation (6.15) as the scalar product of the local flow direction with this reference direction. In summary, equation (6.15) defines a set of Galilean invariant, non-dimensional local flow features.

Clustering algorithm

Clustering is a method used to group data into distinct categories by identifying underlying patterns within the dataset [88]. Unlike supervised learning methods, which require labelled data to guide the learning process, clustering is an unsupervised learning technique that can separate data without the need for predefined labels. In this study, the Gaussian Mixture Model (GMM) is employed for clustering [12]. GMM operates under the assumption that the data are generated from a mixture of Gaussian distributions, classifying the data into sub-populations, each following a Gaussian distribution.

The model iteratively estimates the mean and covariance of these distributions to optimise these parameters. The covariance of each model can be prespecified to share the same general covariance matrix or each component can have its general covariance matrix. The mean and covariance will then be iteratively estimated using the Expectation-Maximisation (EM) algorithm [11], which optimises these parameters. The EM algorithm involves two key steps: the expectation step (E-step), where the conditional expectation of the data is calculated based on the available samples, effectively filling in missing values; and the maximisation step (M-step),

where the parameters of the Gaussian distributions (mean and covariance) are updated by maximising the conditional expectation obtained in the E-step. These steps are repeated until the changes in the estimated parameters are minimal, with the algorithm stopping when the difference in conditional probabilities between iterations falls below a threshold [23]. In GMM, each Gaussian distribution represents a cluster, and data points are assigned to clusters based on their membership probabilities. Several versions of GMM are available in open-source libraries, such as the Gaussian mixture class from the *scikit-learn* Python library [80] used in this work.

6.2.2 Clustering results

The GMM algorithm is applied to various flow configurations to discover regions with different physical dynamics. The four configurations are the zero pressure gradient (ZPG) turbulent boundary layer over a flat plate, the wake flow after a flat plate, the backwards-facing step and the 2D NACA 65 V103 linear compressor cascade (and precisely the separated scenario S_4) described in Chapter 4. The features are computed after simulating the configurations with the elsA solver and the baseline Smith $k - \ell$ model. In this section, only the clustering results using the new feature set will be presented, a similar study to present the clustering results using Ling's features can be found in the Appendix. The different configurations will first be considered independently and the clustering algorithm applied to the sets of features, then the algorithm will be used to cluster the concatenated features of the three first configurations to train the GMM and then used to cluster the NACA65 feature set to test generalisability of the clustering method.

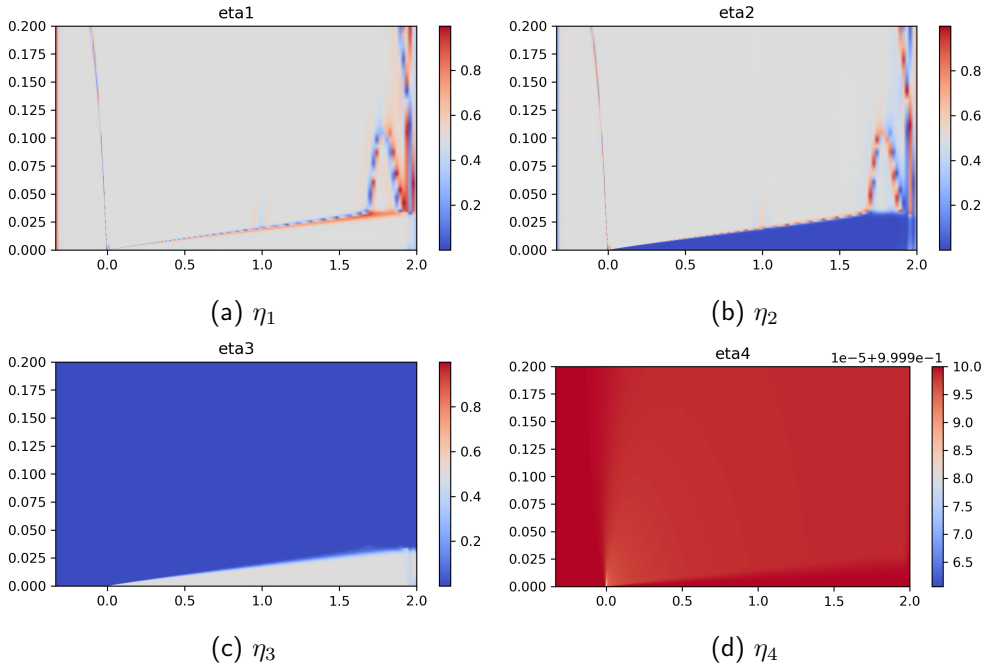
ZPG turbulent Boundary Layer (BL)

Figures 6.1a to 6.1d display the features on the BL configuration. Some *discontinuities* seem to appear notably on feature η_1. This is a numerical artefact that appears because of the normalisation and the absolute value.

Figure 6.2 suggests that the different areas separated by the discontinuity have the same absolute values, but different signs, the discontinuity appears as the frontier on which the signed values changed sign. Also, the change of sign is abrupt because of the normalisation.

In this configuration, there are only two different physical dynamics present, the boundary layer and the free stream. These seem to appear distinctively on the features' spatial representations.

Figures 6.3a to 6.11a display the clustering results using different numbers of clusters for two



(a) η_1 (b) η_2

(c) η_3 (d) η_4

Figure 6.1 – Isocontours of the features for the BL flow.

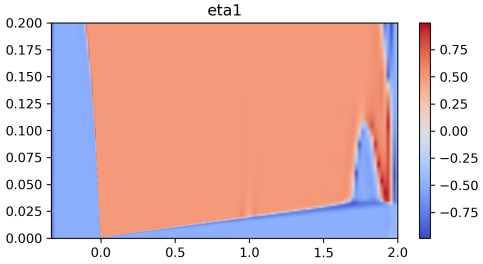
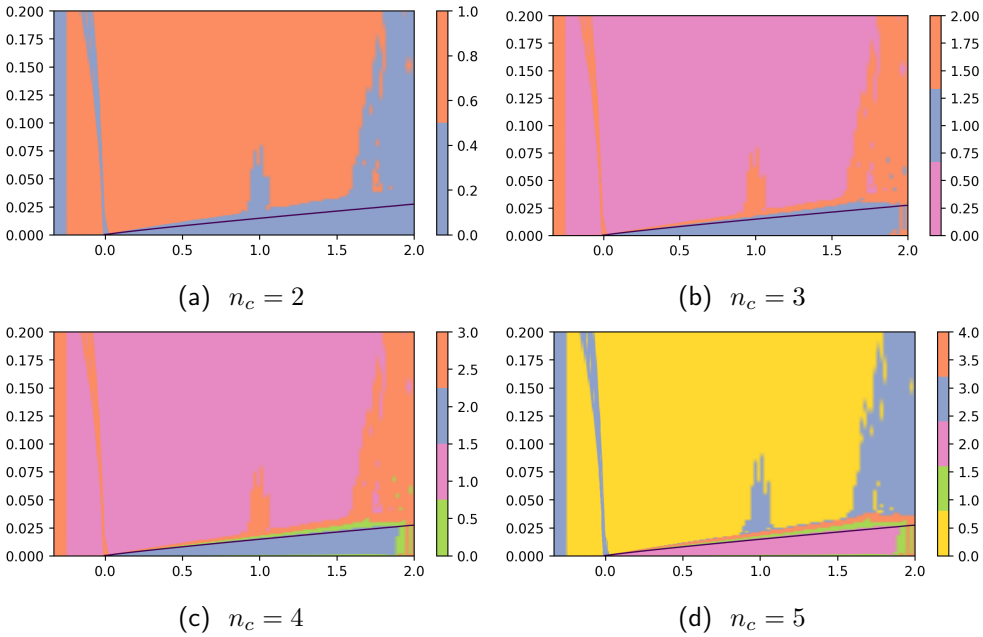


Figure 6.2 – Signed values of feature η_1 for the BL flow.



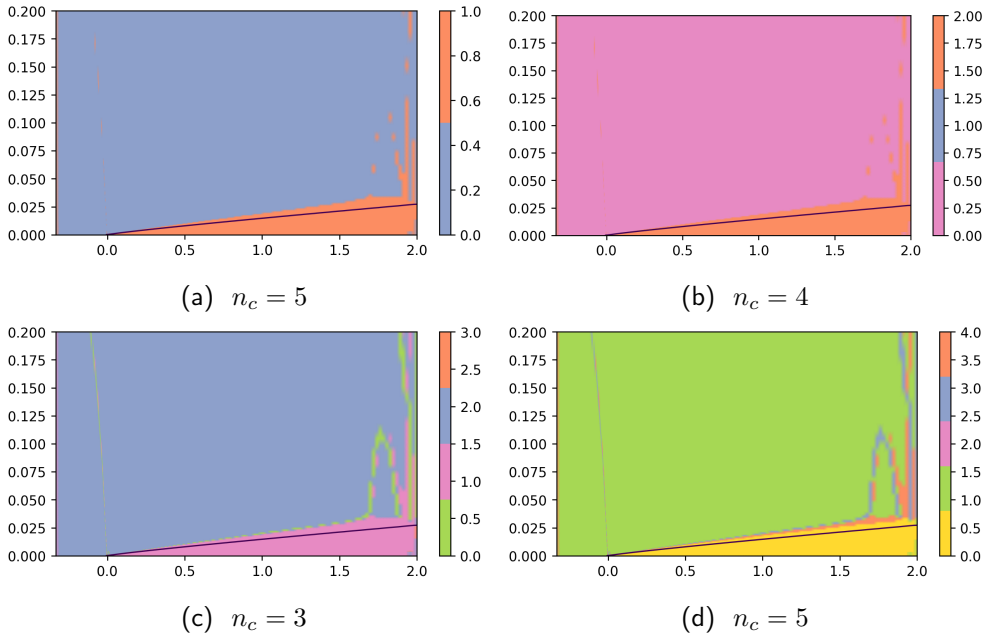
(a) $n_c = 2$ (b) $n_c = 3$

(c) $n_c = 4$ (d) $n_c = 5$

Figure 6.3 – Clustering results for the BL flow with various numbers of clusters based on GMM with full covariance matrices. The black line indicates the contour where $u = 0.99u_{\text{inf}}$.

types of GMM: using a *full* covariance type, in which each component has its general covariance matrix or a *tied* covariance type where all components share the same general covariance matrix. The black line displayed on each figure is a contour of velocity to indicate where $u = 0.99u_{\text{inf}}$, i.e. when the velocity reaches 99% of the outside velocity, indicating the boundary layer thickness. Using both types of covariances, the GMM algorithm distinguishes quite accurately the boundary layer from the inviscid flow using only two clusters and trying to find more clusters leads to transitional clusters or clusters containing the different *discontinuities* mentioned earlier. Note that using the tied covariances between components discovers a slightly thicker boundary layer than expected and adding components in the GMM will refine the BL thickness by adding transitional clusters. This suggests that this type of GMM is cautious in the transitions between clusters.

The major difference between the two types of covariance is how they handle the discontinuities that can be considered as noise. Although a full covariance can capture significantly different cluster shapes by adjusting components, the *scikit-learn* user guide suggests that it is prone to overfitting datasets and does not generalise well to held-out test data. In the following, we will



(a) $n_c = 5$ (b) $n_c = 4$

(c) $n_c = 3$ (d) $n_c = 5$

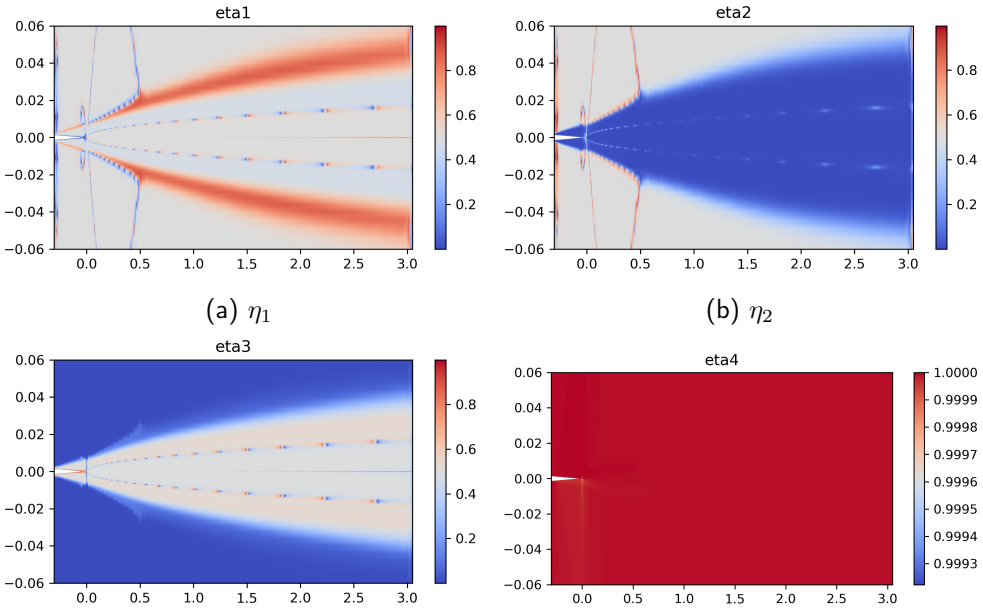
Figure 6.4 – Clustering results with various numbers of clusters using a GMM with tied covariance matrices. The black line indicates the contour where $u = 0.99u_{\text{inf}}$.

only consider GMM using a tied covariance in each component, however, the results of clustering using full covariances are reported in the appendix C.

The wake flow (WF)

Figures 6.5a to 6.5d display the features on the WF. The same *discontinuities* seem to appear notably on features η_1 and η_2. This configuration is quite simple, even if more complex than the precedent. Three physical dynamics are to be discovered in that flow: a boundary layer on the flat plate, a wake and a free stream.

Clustering results using a tied covariance type are presented in figure 6.6a to 6.6d. Using only two clusters will separate the free stream from the stream impacted by the flat plate. The black line indicates the contour where $u = 0.99u_{\text{inf}}$ and suggests that with only two clusters the GMM will be cautious in including all the points impacted by viscous forces in the turbulent cluster. Using three clusters, no difference is observed between the wake and the boundary layer; instead, the transition region between the wake and the freestream is identified. In this region, feature η_1 (the acceleration term seems to dominate the other ones). Using four clusters eventually



(a) η_1 (b) η_2

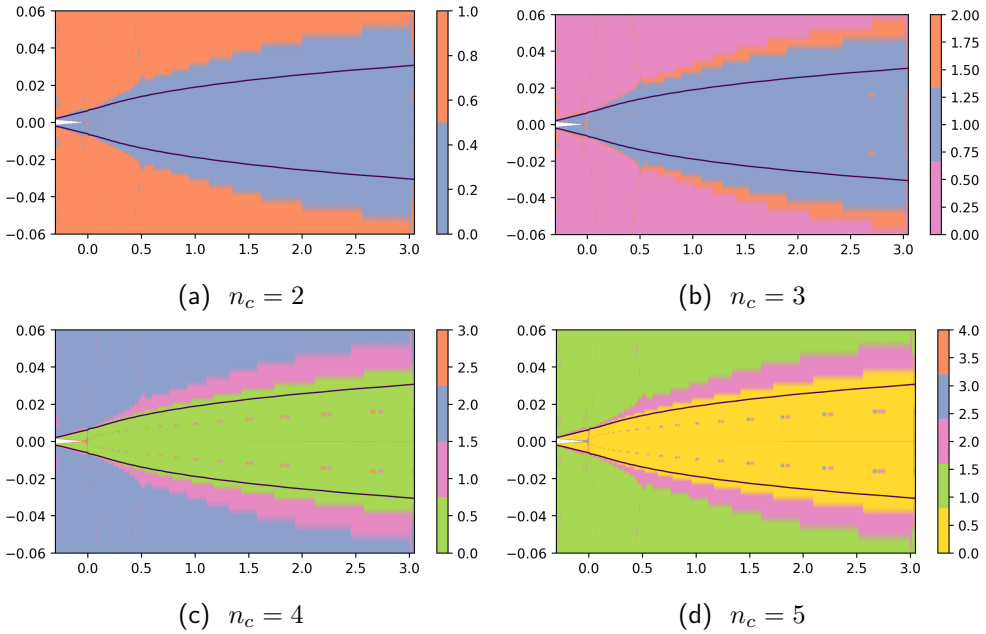
(c) η_3 (d) η_4

Figure 6.5 – Isocontours of the features for the WF flow.

separates a viscous sublayer from the boundary layer, which is dominated by the viscous forces, as suggested by the dominance of the third feature. Adding more clusters does not discover any more physical dynamics and only separates noise. Note that the clustering algorithm using tied covariance does not separate the turbulent boundary layer from the wake. In this configuration or the previous one, the flow is mostly aligned with the horizontal direction, and the last feature is useless in the clustering step as it is constant and equal to 1.

Boundary layer separating from a backward-facing step (SBL)

Figures 6.7a to 6.7d display the features obtained for the separated flow past a step. Some discontinuities seem to appear, notably on features η_1 and η_2. This configuration is more complicated as it includes a recirculation bubble at the bottom of the step. This is why, in this configuration, the fourth feature, which was constant in the previous configurations is now capable of indicating where the flow is reversed. The different physics that the clustering step should aim to discover in this configuration are the recirculation bubble, boundary layers and the potential flow region in the channel core.



(a) $n_c = 2$ (b) $n_c = 3$

(c) $n_c = 4$ (d) $n_c = 5$

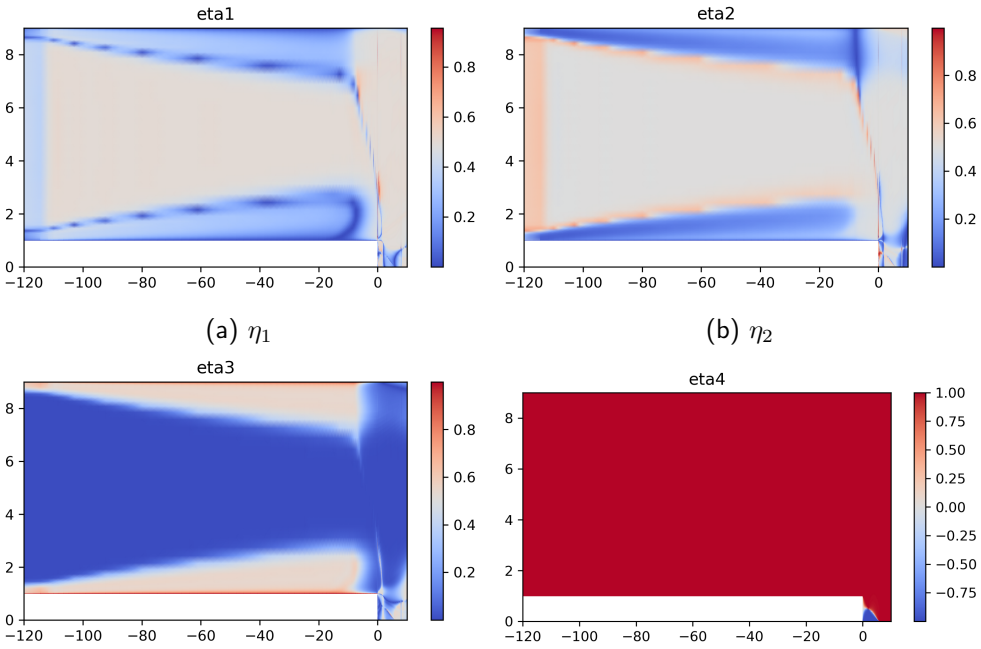
Figure 6.6 – Clustering results with various numbers of clusters using a GMM with tied covariance matrices. The black line indicates the contour where $u = 0.99u_{\text{inf}}$.

Clustering results using a tied covariance type are presented in figure 6.8a to 6.8f. Results using only two clusters would only discriminate the recirculation bubble from the rest of the flow. The clustering step will not however encapsulate the whole recirculation bubble. Using these features, the recirculation bubble discovered by the clustering step is only half of it, the part in which the flow is reversed.

Using three or four clusters separates the core flow from the boundary layer (and a transition region in between). Adding a fifth or sixth cluster allows to separate a viscous sublayer closer to the wall from the rest of the boundary layer, in yellow in 6.11c or blue in 6.12c) for example. Finally, a seventh cluster separates ZPG regions of the boundary layers from those with pressure gradients as suggested by the second feature in figure 6.7b.

The NACA65 V103 linear compressor cascade (N65)

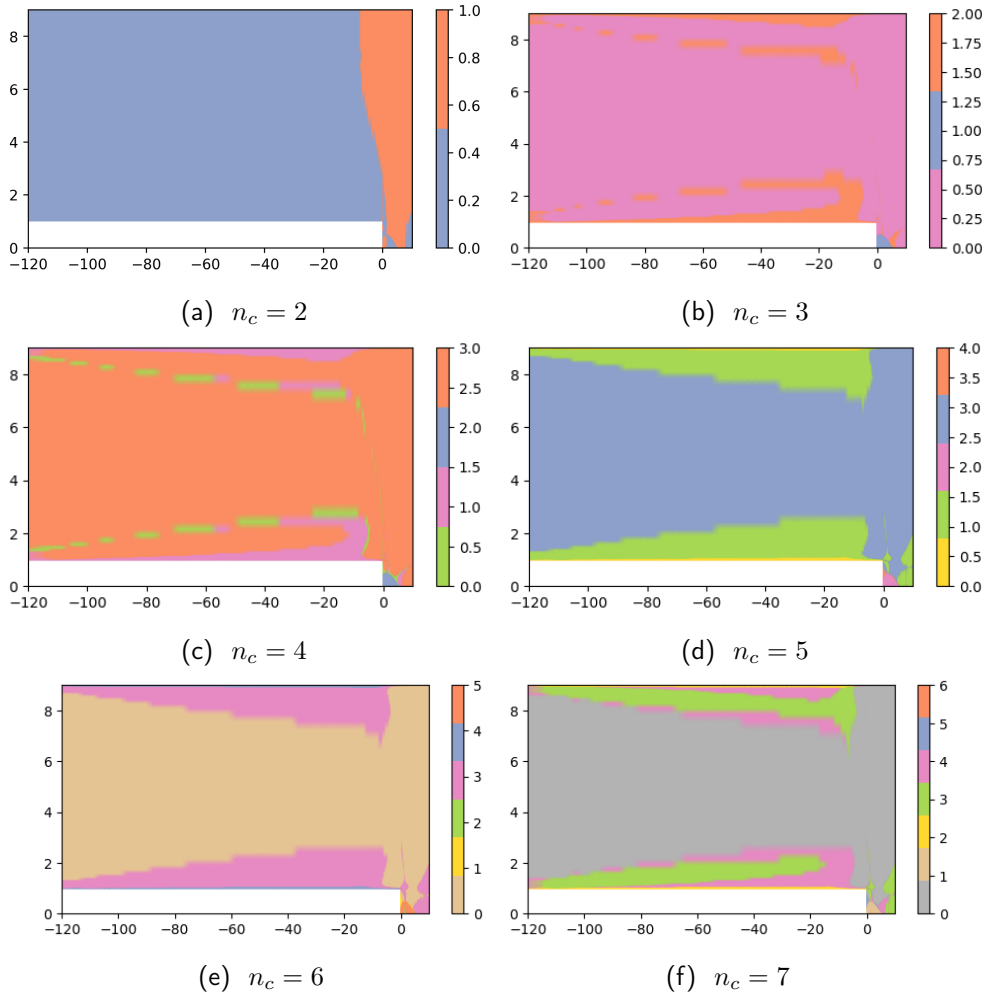
Figures 6.7a to 6.7d display the features on the NACA65 V103 linear compressor cascade. The chosen scenario for this clustering study is scenario S_4 as it includes separation of the boundary layer near the trailing edge, as indicated by the fourth feature, where the flow is reversed. In this



(a) η_1 (b) η_2

(c) η_3 (d) η_4

Figure 6.7 – Isocontours of the features for the SBL flow.

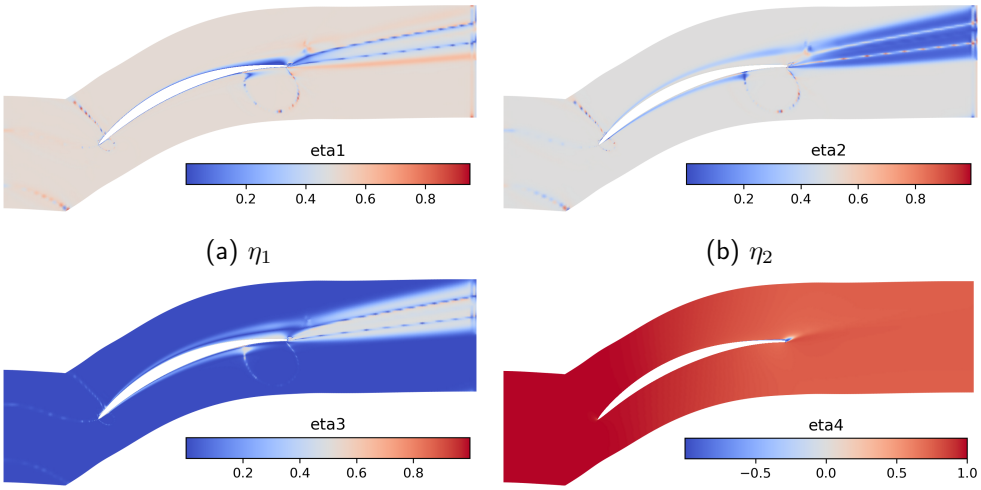


(a) $n_c = 2$ (b) $n_c = 3$

(c) $n_c = 4$ (d) $n_c = 5$

(e) $n_c = 6$ (f) $n_c = 7$

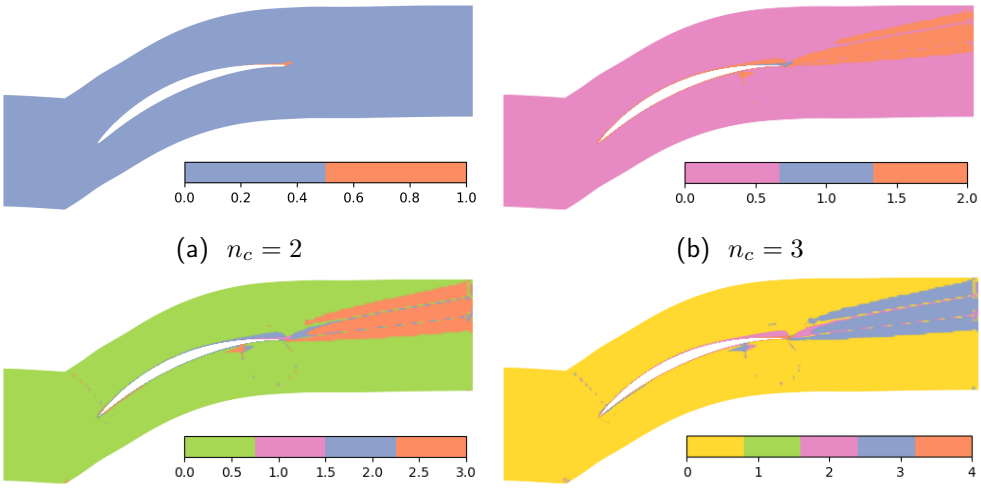
Figure 6.8 – Clustering results with various numbers of clusters using a GMM with tied covariance matrices.



(a) η_1 (b) η_2

(c) η_3 (d) η_4

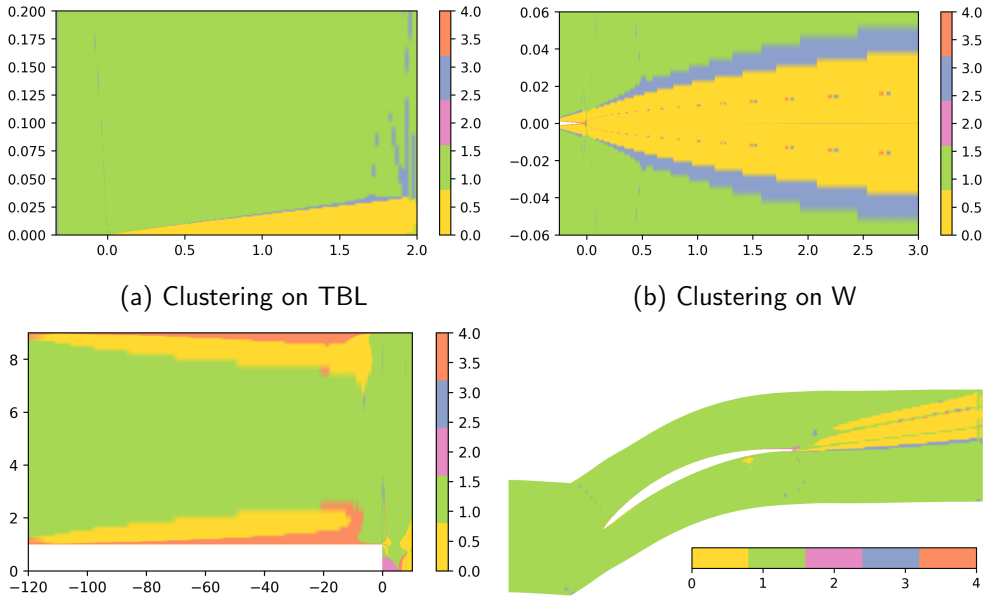
Figure 6.9 – Isocontours of the features for the N65 flow.



(a) $n_c = 2$ (b) $n_c = 3$

(c) $n_c = 4$ (d) $n_c = 5$

Figure 6.10 – Clustering results with various numbers of clusters using a GMM with tied covariance matrices.



(a) Clustering on TBL

(b) Clustering on W

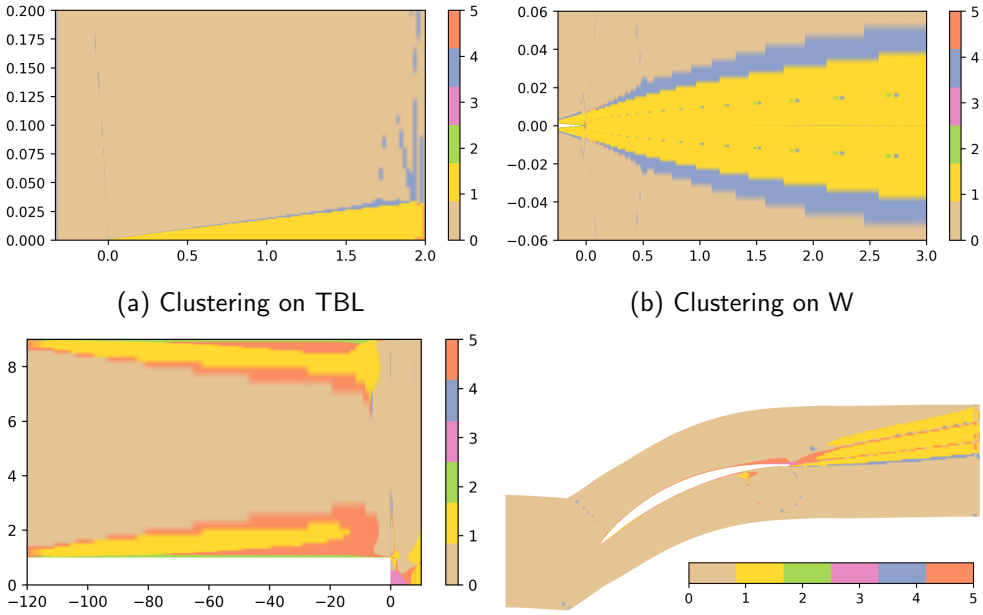
(c) Clustering on BFS

(d) Clustering on N65

Figure 6.11 – Clustering results after training on the TBL, WF and BFS configurations using 5 clusters.

configuration, four physical dynamics are expected: the boundary layers, the separated boundary layer near the trailing edge, the wake and the free stream. The present dynamics are similar to the ones observed for the flat plate boundary layer and the wake cases since the compressor blades are thin bodies. However, the flow exhibits some curvature. Furthermore, the main flow direction is not aligned with the Cartesian axes.

Clustering results using a tied covariance type are presented in figure 6.10a to 6.10d. In this case, using two clusters allows only to discriminate the reverse flow region from the rest of the flow. Using three clusters, a cluster corresponding to the boundary layer and wake is distinguished from the separated region and the free stream. Distinct clusters for the boundary layer and wake are discovered by using four clusters. Adding more clusters will not discover new dynamics or separate further the previously found ones, but rather cluster transition from one cluster to the others.



(a) Clustering on TBL

(b) Clustering on W

(c) Clustering on BFS

(d) Clustering on N65

Figure 6.12 – Clustering results after training on the TBL, WF and BFS configurations using 6 clusters.

Clustering simultaneously on multiple configurations

One of the motivations for developing the new set of features is to improve the generalisability of the discovery process to more complex or multiple configurations. Using the normalised features allows the assembly of different configurations in a single database to be used for clustering. The GMM is then trained on multiple configurations and can then be used to cluster the different physical dynamics present in an unseen configuration. For example, multiple simple flow configurations such as the BL, the WF or the SBL can be used to train the GMM to discover boundary layers, separation or wake and be used to predict the different dynamics present in the NACA65 linear cascade.

In this section, the GMM is trained using a tied covariance using the feature data from the turbulent boundary layer, the wake flow and the backwards-facing step configurations simultaneously. The obtained GMM is used to predict the different physical dynamics in the training configurations and applied to an unseen configuration: the NACA65.

Figures 6.11a to 6.11c depict the clustering results on the training configuration using five clusters. The clustering trained simultaneously for the three elementary configurations predicts the main clusters obtained in single-flow training when applied to only one case in the training set. For example, it separates the TBL in figure 6.11a, the wake and viscous sublayer (in orange) from the free stream in green. The core flow is clustered in the free stream cluster in green on 6.11c, and the separation in pink, the boundary layer in yellow and the sublayer in orange are also discovered in the BFS configuration.

Furthermore, application to the NACA65 cascade predicts correctly the free stream, the wake and the separation region as suggested in figure 6.11d. Note that no boundary layer seems to be detected near the blade. This is surely due to only considering boundary layers under no pressure gradient in the training configurations whereas in the case of the NACA65 compressor blades the boundary layers experience significant boundary layers.

As suggested by the second feature of the BFS configuration in figure 6.7b, a small pressure gradient seem to appear near the step. Adding a sixth cluster to the prediction seems to capture a new dynamic in the BFS configuration, a boundary layer under a pressure gradient in orange as suggested in figure 6.12c. Using this newly trained GMM with six components to cluster the NACA65 configuration seems to capture a turbulent boundary layer close to the blade as suggested in figure 6.12d. These results confirm that the algorithm cannot be used to predict dynamics that were not present in the training set.

6.2.3 Section Summary

The growing application of Machine Learning (ML) in Computational Fluid Dynamics (CFD) post-processing offers promising avenues for the automatic detection of flow regions characterised by specific physical dynamics. This study utilises an unsupervised clustering approach to calibrate and consolidate models within each identified physical dynamic, aiming to develop *expert* hyper-models for these dynamics. The feature set for clustering is built on Callaham's prior work [15], with refinements to ensure the features are non-dimensional, Galilean invariant, and applicable to non-Cartesian domains.

The Gaussian Mixture Model (GMM) is employed for clustering various flow configurations, demonstrating its capability to differentiate between distinct flow regions, such as boundary layers, wakes, and free streams, in multiple scenarios, from simpler configurations such as the turbulent Boundary Layer (BL) configuration, the Wake Flow (WF) configuration and the backwards-facing step (SBL) configuration to more complex, industrial configuration like the 2D NACA65 V103 Compressor Cascade configuration.

The development of a new set of features, as opposed to relying solely on Callaham's features, enhances the generalisability of the clustering process to more complex flow configurations. Using normalised features allows the integration of diverse flow configurations into a unified dataset, allowing the GMM to predict physical dynamics in unseen configurations based on training from simpler flows. This was validated by training the GMM with features from three simple configurations (BL, WF, and SBL). The GMM accurately distinguishes different flow regions in these training configurations and correctly predicts the free stream, wake, boundary layer and separation regions in the unseen NACA65 configuration after fine-tuning. The results suggest that to predict all physical dynamics in an unseen flow configuration effectively, similar dynamics must be present in the training dataset. Including all relevant dynamics in the training set is required to ensure the model's ability to generalise to new scenarios.

Overall, this approach appears promising as it enables the detection of various physical dynamics and demonstrates generalisability. The method is also advantageous due to its straightforward implementation and minimal computational and storage requirements. However, further development is needed to address some limitations identified. Specifically, the current features and tied covariances in the GMM do not fully distinguish between turbulent boundary layers and wakes, though using full covariances seems to, even if it raises other issues. Additionally, the use of the flow-inversion feature captures only half of the separation regions, detecting only

the reversed flow. Exploring the use of signed feature values, adding more features (such as projections onto orthogonal directions), or incorporating features from other methods could offer solutions to these challenges.

6.3 Calibration of RANS models within clusters

The global calibration presented in the previous chapter suffers from an over-specialisation of the model in certain areas of the flow, which enhances predictions in those areas at the risk of diminishing performance in others. To leverage the model's ability to specialise in predicting specific regions of the flow, the method presented in this chapter calibrates models within clusters, i.e., within similar physical dynamics, as the clustering step seeks to differentiate between the various dynamics present in the flow.

6.3.1 Observed Data, Models, and Prior Distributions

As in the previous chapter, four linear RANS models are calibrated on the NACA65-V103 compressor cascade using the synthetic data generated by the EARSM $k - k\ell$ model. Although this is not a truly high-fidelity model, the generated data will allow for a proof-of-concept of the method in such a configuration.

We recall that the linear eddy viscosity at stake are the Spalart-Allmaras [97], Wilcox 2006 $k - \omega$ [108], Smith $k - \ell$ [95], and Launder-Sharma $k - \epsilon$ models [58].

The Bayesian calibration methodology follows the same approach outlined in the preceding chapter. The prior distributions of model parameters are the same we used in Chapter 5, and gathered in Tables A.5 to A.12. While the stochastic models are the same, the reference data used for calibration are now drawn from the various clusters, leading to a different set of updated models for each cluster discovered in the previous section for the NACA65 configuration and the off-design scenario S_4 (characterised by flow separation upstream of the trailing edge, at the blade suction side).

The clusters and their abbreviations are displayed in figure 6.13, and are obtained by clustering the reference data using a GMM with four components and tied covariance matrices. More details about the clustering step were provided in the previous section. Due to turbulence modelling inaccuracies, clusters based on each competing model do not coincide with the reference. Thus, the calibration data are extracted only at points contained in the reference clusters. This ensures that even if the predicted clusters of each LEVM can differ from the EARSM ones, all LEVMs are calibrated on the same subsets of data. The clusters obtained do not contain the same number of data. As suggested by the results from the previous chapter, having more data for calibration can lead to more peaked posterior distributions. Therefore, every point assigned to a cluster by

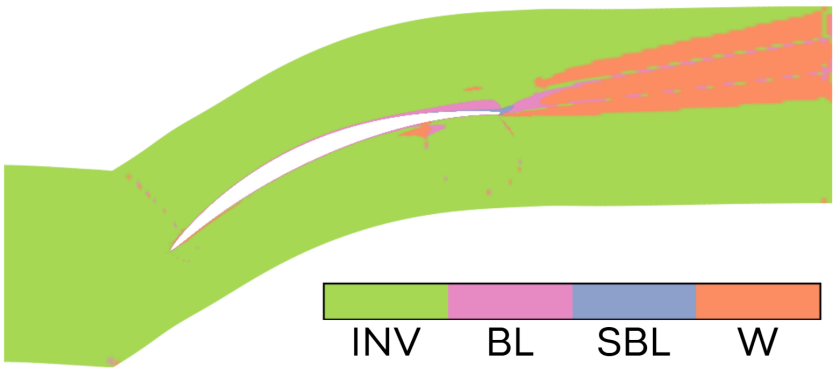


Figure 6.13 – Wake (WF), Separated Boundary Layer (SBL), Boundary Layer (BL) and INViscid flow (INV) clusters on scenario S_4 of the NACA65 compressor cascade.

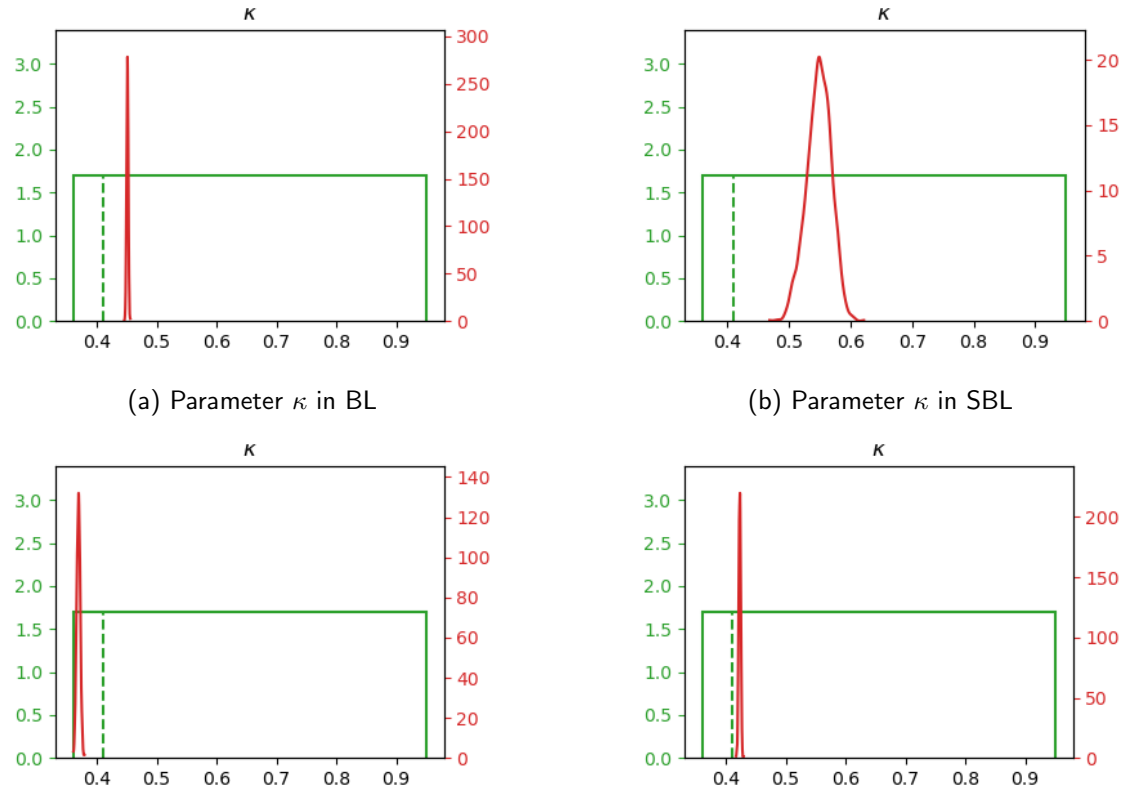
the clustering algorithm is used for calibration.

6.3.2 Cluster-dependent posterior distributions

In Figures 6.14a to 6.14d we show selected posterior probability distributions of the κ coefficient obtained after calibration of the $k-\ell$ model on the four different clusters, to exemplify the results. The MAP estimates for all parameters in all clusters are available in the Appendix A. The posterior distributions are not centred around the parameter's baseline values. This suggests that the models need corrections to better match the observed data. Note that the models were calibrated against synthetic data using an EARSM model, with parameters optimised to match the EARSM data. This context may explain why the calibrated κ for the boundary layer (BL) cluster, is not centred on the baseline value, even if the baseline model was carefully calibrated for boundary layer regions. One might also argue that the parameters in the inviscid flow region should remain uninformed. However, as discussed previously, the choice of the number of clusters is somewhat arbitrary and can affect the calibration process. Furthermore, the flow in the inviscid region is still affected by the development of the boundary layers, and thus it indirectly depends on the turbulence model predictions.

It is also worth noting that the posterior distributions of the same parameter, such as the κ parameter of the $k-\ell$ model depicted in figures 6.14a to 6.14d, are not centred around the same values. This indicates that the specialised models are significantly different from one another.

Then, the posterior distributions are highly peaked for most clusters, indicating that the



(a) Parameter κ in BL

(b) Parameter κ in SBL

(c) Parameter κ in W

(d) Parameter κ in INV

Figure 6.14 – Prior and posterior distributions for the $k - \ell$ κ parameter on scenario S_4 for the four different clusters. The dashed line is the baseline value.

parameter posterior distributions are well-informed. The only exception is the SBL cluster, which shows less peaked posterior distributions. This is attributed to the fewer data contained in this cluster. Despite this, the modes of the posterior distributions remain distinctly defined, and the posterior distributions can be safely approximated as Dirac distributions centred around the mode.

Also, as suggested in table 6.1 gathering the Maximum A Posteriori (MAP) values of the hyper-parameter σ_η, for each cluster, the hyper-parameter posterior distribution is centred around a value smaller than 0.01. This suggests that the updated RANS models can accurately represent the observed data by only adjusting the model parameters. If the data could not be matched by merely varying the model parameters, the stochastic model would have required a higher

	S-A	$k-\ell$	$k-\omega$	$k-\varepsilon$
BL	1.50e-4	1.20e-2	7.50e-3	3.84e-3
SBL	4.65e-5	6.92e-7	7.72e-7	6.60e-7
WF	8.48e-3	8.75e-3	3.96e-3	9.22e-3
INV	3.84e-7	2.50e-3	1.40e-5	1.43e-2

Table 6.1 – MAP values of the σ_η hyper-parameter on the different clusters for the 4 RANS models calibrated.

value for the σ_η hyper-parameter to include the data in the posterior predictive distribution. This indicates that the models are effectively calibrated to fit the observed data within each cluster.

Finally, as previously mentioned, the value of the hyper-parameter σ_η is an indicator of trust regarding the use of recalibrated models relative to each other. For example, we found that the k_ε and the $k-\ell$ models should be trusted less than the Spalart-Allmaras or the $k-\omega$ models on cluster BL, because their associated model-form uncertainty parameter (reported in table 6.1 for all clusters and models) is higher.

6.3.3 Predictions using the updated models

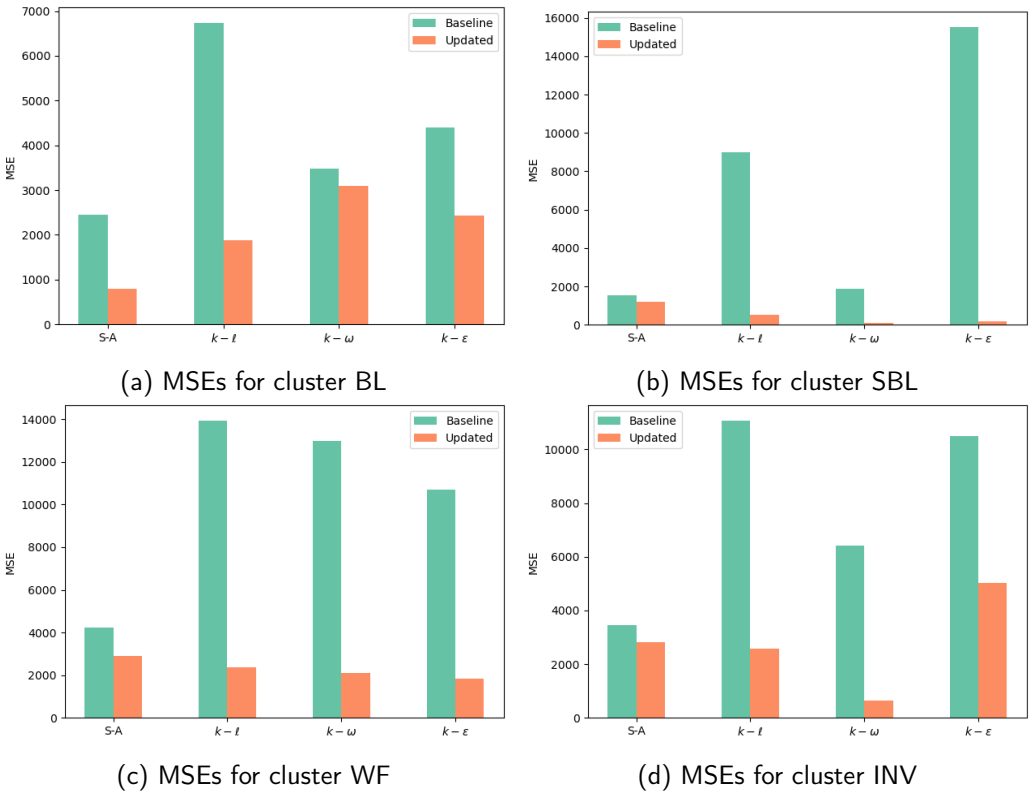
Based on the MAP estimates of the model parameters, predictions of the total pressure field can be generated by propagating these MAP values into the CFD solver. The calibrated parameters can then be used to either simulate the calibration scenario or to predict previously unseen scenarios or configurations. Consequently, the updated model can be used to predict the total pressure field or any other quantity that the model is capable of predicting.

Prediction on the calibration scenario

After propagation through the CFD solver, the mean squared error (MSE) between the predicted data and the observed data, defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\Delta_{\text{obs},i} - \Delta_{\text{pred},i}\right)^2, \tag{6.17}$$

can be computed. These MSEs are illustrated in the bar plots presented in Figures 6.15a to 6.15d. The figures demonstrate that, for all models and clusters, the calibration step significantly reduces the mean squared error *within the calibration cluster*.



(a) MSEs for cluster BL

(b) MSEs for cluster SBL

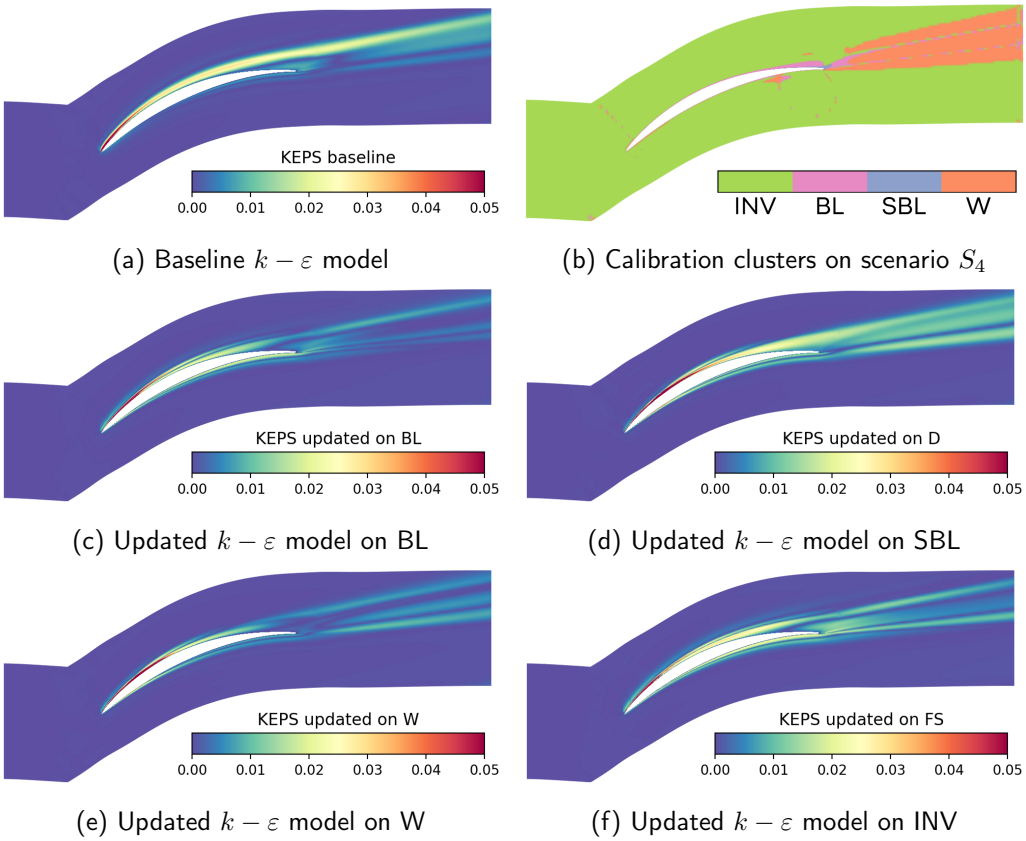
(c) MSEs for cluster WF

(d) MSEs for cluster INV

Figure 6.15 – MSE of the total pressure field for the baseline and updated models after calibration on the total pressure field and prediction on S_4.

Figures 6.16a and 6.16f show the relative error in total pressure fields for both the updated and baseline $k-\varepsilon$ models on the training scenario S_4, following calibration on different clusters. The figures indicate that error reduction varies significantly across the flow, depending on the calibration cluster. Specifically, there is a noticeable reduction in error within the calibration cluster, as shown by the MSEs, while errors tend to increase outside this cluster, particularly after calibration on cluster SBL. This suggests that the cluster-specialised model may overfit in these regions. However, such behaviour is both expected and desirable, as the aim is to apply the specialised model only within its designated regions.

Figure 6.17 confirms that, while the updated models enhance performance within the calibration cluster, some models may decrease overall model performance. This underscores the critical importance of the aggregation step, which must effectively leverage the improved performance of the expert models within their regions of expertise. Additionally, this indicates



(a) Baseline $k-\varepsilon$ model

(b) Calibration clusters on scenario S_4

(c) Updated $k-\varepsilon$ model on BL

(d) Updated $k-\varepsilon$ model on SBL

(e) Updated $k-\varepsilon$ model on W

(f) Updated $k-\varepsilon$ model on INV

Figure 6.16 – Relative error on total pressure on S_4 for the baseline and the updated $k-\varepsilon$ model on the different clusters.

that the global prediction of a model calibrated for a specific cluster—representing a particular physical dynamic—should not be relied upon for predictions outside of that dynamic, as performance outside the calibrated cluster is not guaranteed.

The use of MAP parameters for the updated model facilitates the extraction of other quantities of interest (QoIs) as outputs of the CFD solver, allowing for the evaluation of the calibration step's impact on various quantities. The following bar plots in Figures 6.18a to 6.18h present the mean squared errors for the pressure and velocity fields, comparing both the baseline and updated models within the calibration scenario S_4 in the different clusters. These figures indicate that, while calibration significantly reduces the error for the calibration quantity relative to the reference data, the errors for other quantities do not necessarily show the same level of

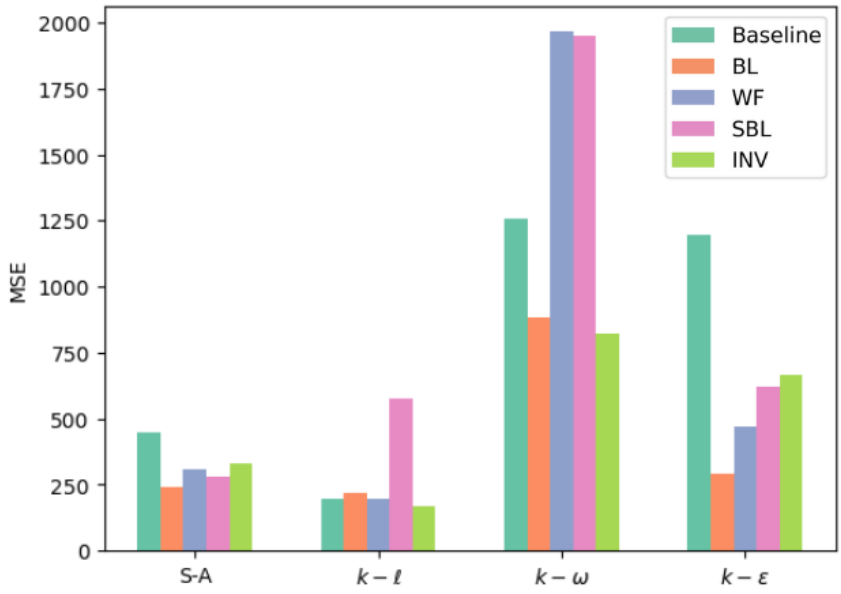


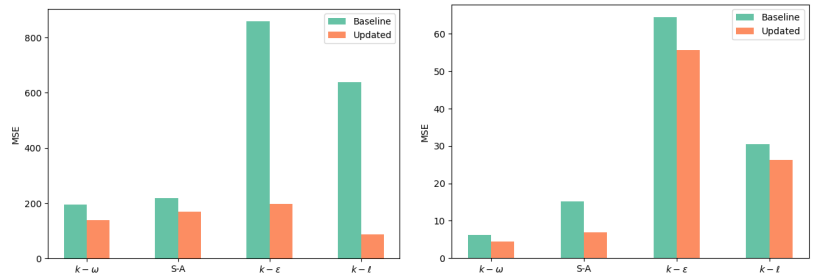
Figure 6.17 – MSE of total pressure fields for the baseline and updated models on scenario S_4 after prediction on the whole field.

improvement. The reduction in error for quantities not included in the calibration process varies considerably between models and clusters. Overall, the figures suggest that the updated models are generally as effective as, or better than, the baseline models. However, some models perform slightly worse on the velocity fields in certain clusters. This implies that while the models perform satisfactorily for unseen quantities, the results, combined with the conclusions from the previous chapter, suggest that parameters obtained from calibration for one quantity should be applied with caution, as there is no strong evidence that they will consistently improve performance for different quantities.

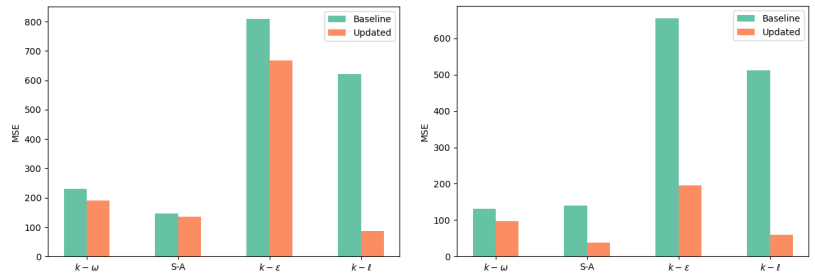
Prediction on an unseen scenario

The MAP parameters obtained during the calibration on scenario S_4 can be applied to simulate a different scenario to assess the generalisability of these updated parameters. In this case, the updated models simulate scenario S_3. The clusters identified in scenario S_3, using the Gaussian Mixture of Models trained on S_4, are depicted in Figure 6.19.

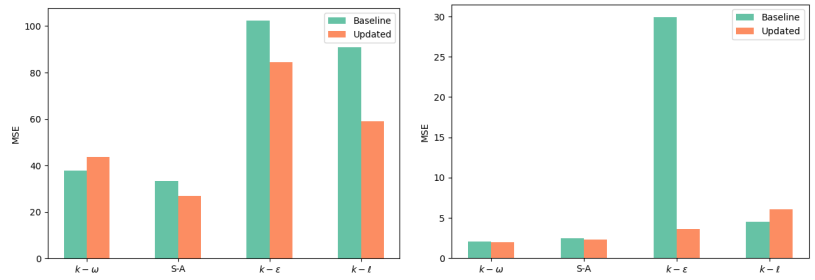
Figures 6.20a to 6.20d reveal that the calibration applied to a different scenario reduces



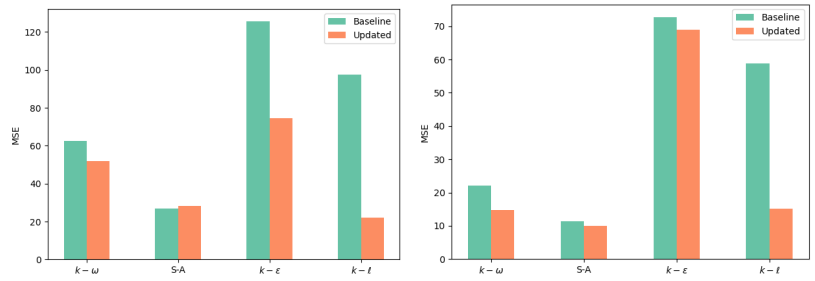
(a) MSEs on pressure for cluster BL (b) MSEs on pressure for cluster SBL



(c) MSEs on pressure for cluster WF (d) MSEs on pressure for cluster INV



(e) MSEs on velocity for cluster BL (f) MSEs on velocity for cluster SBL



(g) MSEs on velocity for cluster WF (h) MSEs for cluster INV

Figure 6.18 – MSE of the pressure and velocity fields for the baseline and updated models after calibration on the total pressure field and prediction on S_4.

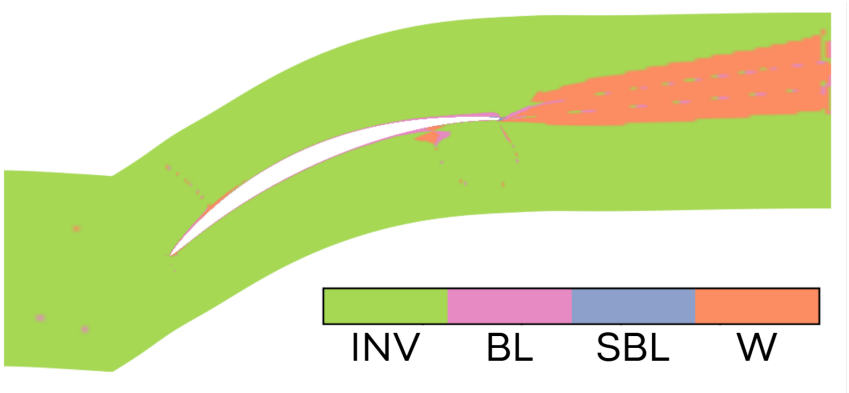


Figure 6.19 – Wake (W), Separated boundary layer (SBL), Boundary Layer (BL) and Inviscid flow (INV) clusters on scenario S_3 of the NACA65 compressor cascade.

the error of the calibration quantity for nearly all clusters, indicating that calibration within a specific cluster, and thus a specific physical dynamic, proves to be more generalisable to different scenarios compared to the global calibration method previously studied.

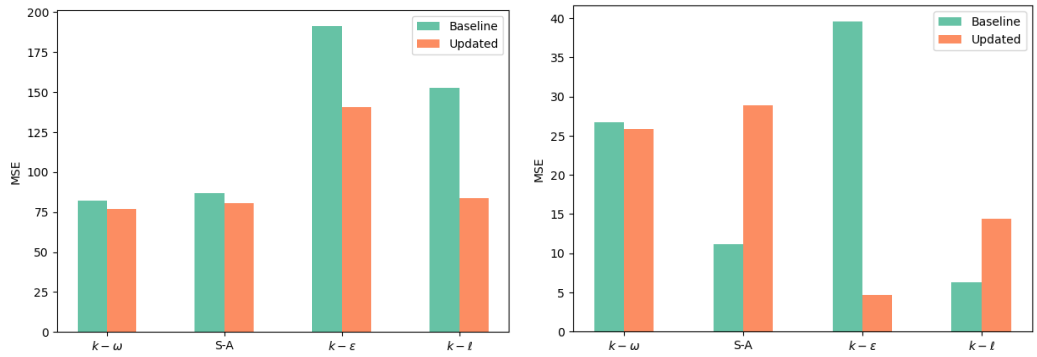
In the separated boundary layer cluster, the updated models perform worse than the baseline models. However, as indicated by the clusters identified in scenario S_3 displayed in figure 6.19, there are only a few data points in such a cluster, because there is almost no separation around the trailing edge, although the clustering step identifies some points in this region. The increased MSE in this cluster is not concerning and is unlikely to affect the overall predictions significantly.

The results for a quantity not used for the calibration quantity are more mitigated, i.e. their performances can vary depending on the models and the cluster and no guiding principle can be extracted from this result.

6.3.4 Section summary

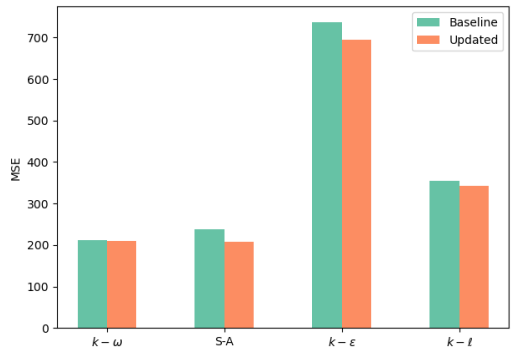
The calibration of the closure parameters for four different Reynolds-Averaged Navier-Stokes models was performed within clusters defined by similar physical dynamics. This approach aims to avoid the over-specialisation of the model that occurs with global calibration and to enhance model accuracy in specific regions of the flow.

1. Calibration results show that the posterior distributions are well-informed and not centred around baseline values, indicating that the models require adjustments to fit the observed

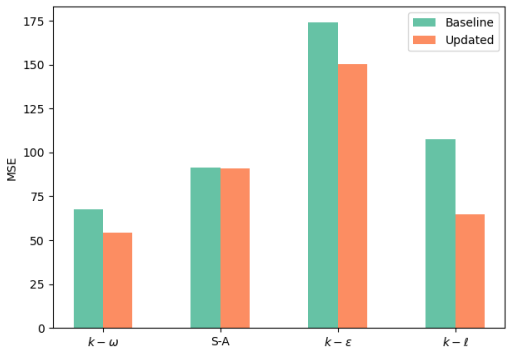


(a) MSEs for cluster BL

(b) MSEs for cluster SBL



(c) MSEs for cluster WF



(d) MSEs for cluster INV

Figure 6.20 – MSE of the total pressure field for the baseline and updated models on scenario S_3 after calibration on the total pressure field on S_4.

data. The hyper-parameter values are low across all clusters, suggesting that the models are well-calibrated to fit the observed data, with low model-form uncertainty. The updated models show significant differences in parameters across clusters, reflecting their specialisation to different physical dynamics. This specialisation helps improve performance within the calibration clusters but may lead to reduced performance outside these regions.

2. Calibration within clusters significantly reduces the mean squared error for the calibration quantity within the calibration cluster, demonstrating the effectiveness of cluster-specific calibration. However, this may lead to increased errors outside the calibration cluster. This suggests that while models perform better within their specialised regions, caution is needed when applying them to regions outside their calibration scope, highlighting the need for model aggregation and taking advantage of models' performances in their specialisation

dynamic. Updated models generally perform better than baseline models for the calibration quantity, but performance varies for other quantities, indicating that the improvements might not consistently transfer to all quantities.

3. Calibration performed on one scenario can improve model predictions for unseen scenarios, demonstrating that cluster-specific calibration enhances the generalisability of the updated models. Performance improvements for calibration quantities generally transfer to unseen scenarios, but variations in performance for other quantities suggest that the models' effectiveness may be scenario-specific.

6.4 Clustered aggregation of updated models

The philosophy behind the proposed methodology is to combine models within each cluster to generate cluster-specific predictions that are spatially restricted. These individual cluster predictions are then aggregated to reconstruct the global prediction. Essentially, the aim is to create cluster-specific predictions—akin to pieces of a puzzle—and then assemble these pieces to form the overall prediction. Since physical fields are continuous, these *puzzle pieces* must be smooth. Therefore, fuzzy clustering methods such as GMMs, which provide probabilities of cluster membership, are essential for ensuring the continuity and smoothness of the predictions.

6.4.1 Cluster dependent weighting

To construct the model mixtures within each cluster for subsequent predictions, it is necessary to compute the cluster weights, according to the proposed methodology, the weights defined in equation 6.3 and represent the average of the local model plausibilities of each cluster updated model.

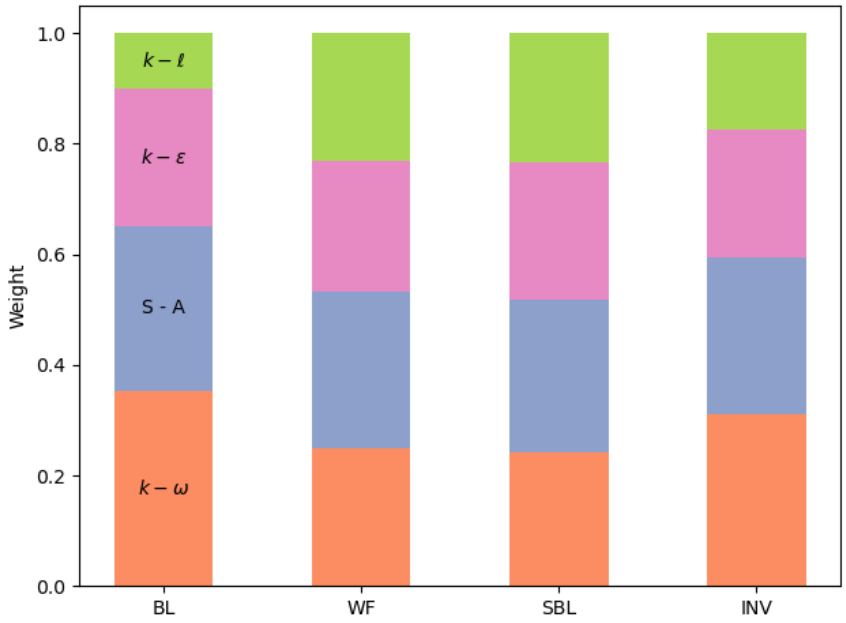


Figure 6.21 – Weights for each cluster in the flow, computed for scenario S_4.

Figure 6.21 and table 6.2 indicate significant variations in model performance across different

	S-A	$k-\ell$	$k-\omega$	$k-\varepsilon$
BL	0.35	0.30	0.25	0.10
SBL	0.25	0.28	0.24	0.23
WF	0.24	0.28	0.25	0.23
INV	0.31	0.28	0.23	0.17

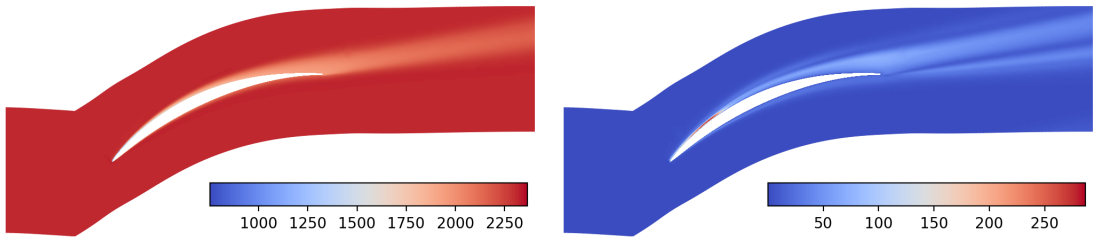
Table 6.2 – Weights for each cluster in the flow, computed for scenario S_4.

clusters. Notably, models such as Spalart-Allmaras and $k-\omega$ consistently receive higher preference compared to $k-\epsilon$, which appears to be less favoured despite showing significant error reduction within the clusters, as detailed in the calibration section of this chapter.

6.4.2 Reconstruction and prediction of a quantity of interest

Based on the methodology provided earlier, we can reconstruct the CMA predictions of a QoI Δ^* according to equation 6.4. The prediction methodology is applied to the training scenario and unseen scenario to predict the calibration quantity and unseen quantities to test the CMA framework.

On the training scenario



(a) expected value (b) Standard deviation

Figure 6.22 – Total pressure field expected value standard deviation after training and prediction on scenario S_4 within the CMA framework.

Using the equations provided in the methodology section of this chapter, the expectation and variance of the total pressure field on scenario S_4 can be easily computed. These results are presented in figures 6.22a and 6.22b, respectively. Additionally, 6.23 shows the error of the expectation relative to the observed total pressure field.

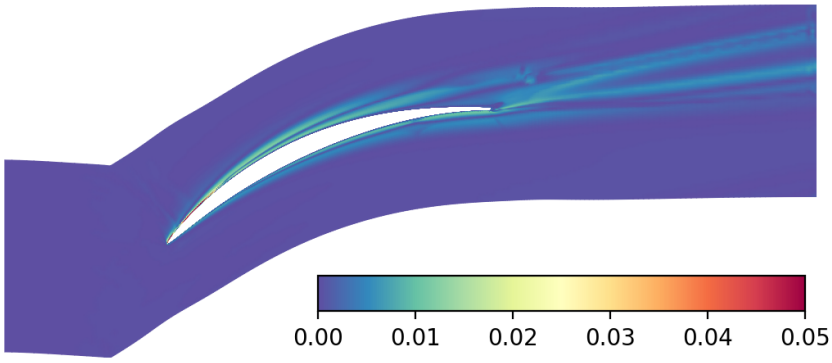


Figure 6.23 – Relative error on total pressure field after training and prediction on scenario S_4 within the CMA framework.

Firstly, the relative error of the reconstructed field appears significantly reduced compared to the baseline models (for error fields, see here). However, the combination does not consistently outperform the baseline models in all regions, although it may provide superior performance on a global scale.

Although the standard deviation of the predictions remains relatively small—about 10% of the prediction value—it is nearly five times greater than the standard deviation of the XUMA predictions discussed in Chapter 5, section 5.3. This increased standard deviation can be attributed to the formulation of the CMA weighting method. In the CMA method, within-cluster weights are constant scalars that represent the average weight of a model within the cluster, resulting in fewer discriminative weights compared to those in the previous chapter. There, the weights varied significantly, whereas, in this chapter, they ranged between 0.10 and 0.35. Consequently, every model retains significant influence within the cluster, even if it has the smallest weight, leading to a higher standard deviation compared to the XUMA method, where model weights are more discriminative.

Figure 6.24 presents the MSEs for scenario S_4 after training on the same scenario. The figure indicates that the CMA framework significantly reduces MSE compared to the baseline models, with the CMA combination outperforming all baseline models. However, compared to the XUMA predictions from the previous chapter (see here), the CMA prediction achieves a smaller reduction in error. This can be attributed to the CMA model weights being constant across the scenario, whereas in the XUMA framework, weights vary at each point in the flow. As a result, the XUMA combination can more closely match the observed data.

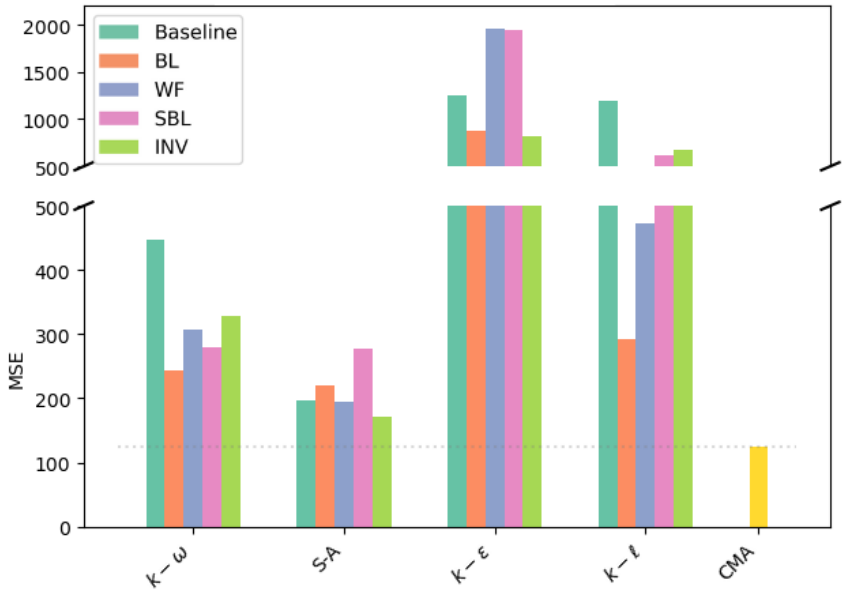
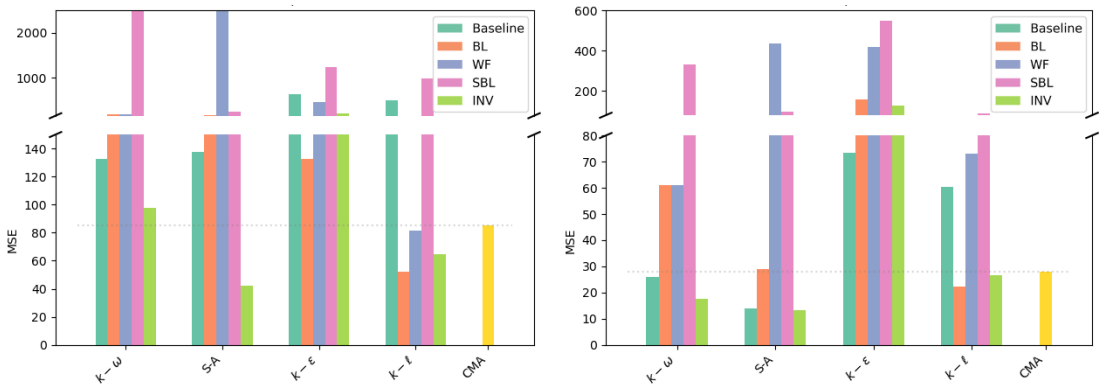


Figure 6.24 – MSE of the reconstruction of the baseline and updated models and CMA predictions after training and prediction on scenario S_4 w.r.t the reference data



(a) Pressure

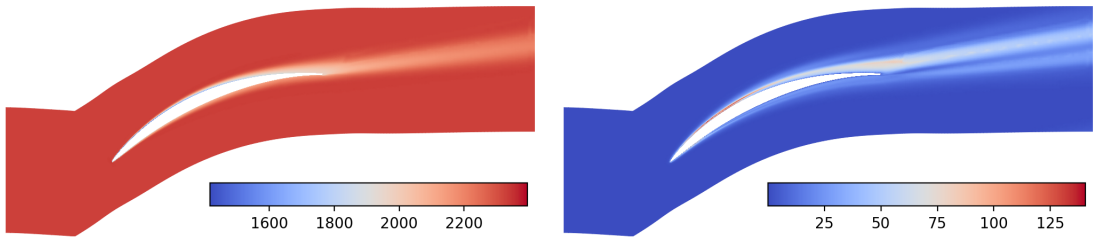
(b) Velocity

Figure 6.25 – MSE of the reconstruction of the baseline and updated models and CMA predictions after training and prediction on scenario S_4 w.r.t the reference data

Figures 6.25a and 6.25b present the MSEs for scenario S_4 after training on the same scenario for the pressure and velocity fields. These figures demonstrate that the MSEs for both quantities

are significantly reduced compared to most baseline models. However, this improvement in unseen quantities does not necessarily imply that the CMA predictions will consistently perform well on every unseen quantity. There is no substantial evidence to suggest that improved performance in one quantity leads to improvements in others. Further investigation is needed to clarify this relationship. Until then, the discussion around the prediction of unseen quantities remains interesting but should not be regarded as a definitive guideline.

On an unseen scenario



(a) expected value (b) Standard deviation

Figure 6.26 – Total pressure field expected value and standard deviation after training on scenario S_4 and prediction on scenario S_3.

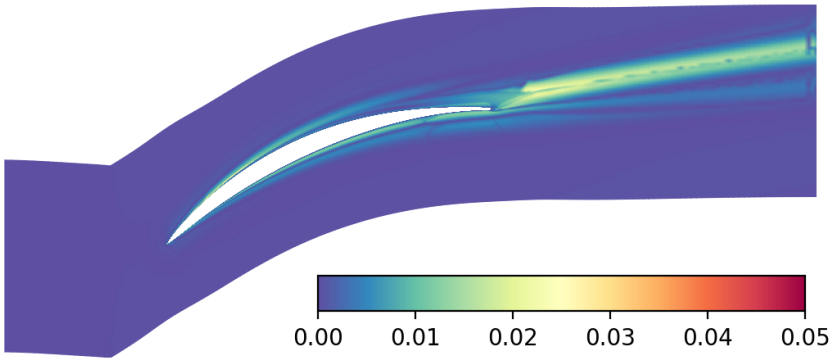


Figure 6.27 – Relative error on total pressure field after training on scenario S_4 and prediction on scenario S_3.

The CMA methodology does not require the interpolation of weights onto a feature space. Instead, model weights are assigned as scalars to each cluster. In any configuration, the weights

for a given cluster remain equal to those in the training configuration. This simplifies the implementation of the CMA method and reduces both the computational and storage demands typically associated with a Random Forest Regressor.

Figures 6.26a to 6.27 compile the prediction expected value, standard deviation and relative error for the out-of-training scenario S_3 following training on S_4. Similar observations regarding the higher standard deviation value apply here. Note that, the standard deviation shown should be interpreted as an indicator of the agreement among the different models due to the limited number of models in the mixture.

Figure 6.27 along with figure 6.28 suggest that the CMA prediction on scenario S_3 after training on scenario S_4 continues to outperform the baseline RANS models. Yet, the CMA prediction does not consistently outperform every updated RANS model. It is important to recall that each updated model is specifically calibrated to match a particular cluster, so its prediction on the global field should not be considered an accurate reflection of its overall predictive performance.

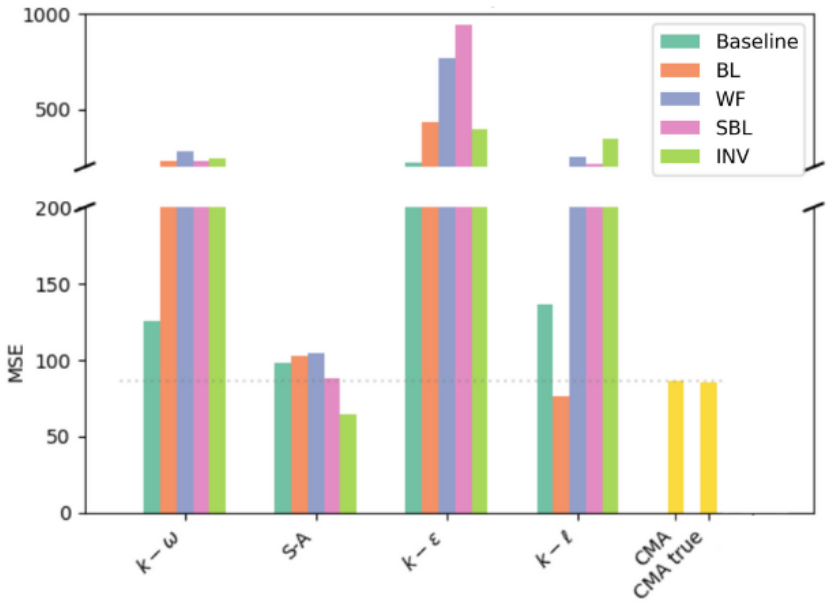


Figure 6.28 – MSE of the reconstruction of the baseline and updated models and CMA predictions after training on scenario S_4 and prediction on scenario S_3 w.r.t the reference data

Figure 6.28 shows the MSE of the CMA predictions using weights computed for scenario S_4

and the MSE of the *CMA true* which uses weights computed for scenario S_3 as data is available. The CMA *true* represents the best possible performance achievable. Remarkably, the MSEs for both CMA and *CMA true* are almost identical, indicating that the weights derived from the training scenario effectively apply to the prediction scenario. This result can be attributed to the local updates of the models and the within-cluster weighting, which primarily serves to select the best-performing model aggregation for a specific physical dynamic. Therefore, the selection for prediction depends more on the clustering process than the within-cluster weighting. This highlights the critical importance of the clustering step in the methodology.

This hypothesis is supported by comparing the weights used to construct the CMA and CMA *true* predictions. Table 6.3 shows the weights computed for the scenario S_4 clusters after calibration on the same scenario used for the CMA prediction. Table 6.4 presents the weights computed for scenario S_3 after calibration on scenario S_4 used for the CMA *true* prediction. The similarity between these weights accounts for the minimal difference in MSE between the CMA and CMA *true* predictions.

	S-A	$k-\ell$	$k-\omega$	$k-\varepsilon$
BL	0.35	0.30	0.25	0.10
SBL	0.25	0.28	0.24	0.23
WF	0.24	0.28	0.25	0.23
INV	0.31	0.28	0.23	0.17

Table 6.3 – Weights computed on S_4.

	S-A	$k-\ell$	$k-\omega$	$k-\varepsilon$
BL	0.36	0.29	0.25	0.10
SBL	0.26	0.28	0.24	0.22
WF	0.23	0.27	0.26	0.24
INV	0.31	0.28	0.24	0.17

Table 6.4 – Weights computed on S_3.

6.4.3 Comparison of two weighting methods: space-dependent and cluster-dependent method

After propagation of the Maxima A Posteriori of the model parameters, predictions of any available field in the CFD solver are possible just as if the updated models were brand new expert models. The CMA methodology relies on the clustering and within-cluster weighting of models however it is interesting to consider the mixture of each cluster-updated model as new independent updated models and combine them just as in the XUMA methodology presented in the previous chapter.

Differences in the weighting methods

CMA weights definition

Computation of weight for model M_m in cluster C_c :

$$W_{m,c} = \frac{\sum_{i=1}^{N_{D,C}} p(M_m|d_{c,i})p(d_{c,i})}{\sum_{k=1}^{N_M}\sum_{i=1}^{N_{D,C}} p(M_m|d_{c,i})p(d_{c,i})}$$

The weights are scalars associated with a cluster and a model. No regression is needed to predict a new configuration.

XUMA weights definition

Computation of weight for model M_m at observation $d_i \in \Delta_{obs}$:

$$W_{im} = p(M_m|d_i, \Delta_{obs})$$

The weights are scalars associated with a spatial point. Regression onto a feature space using an RFR is needed for future predictions.

Predictions on the training scenario

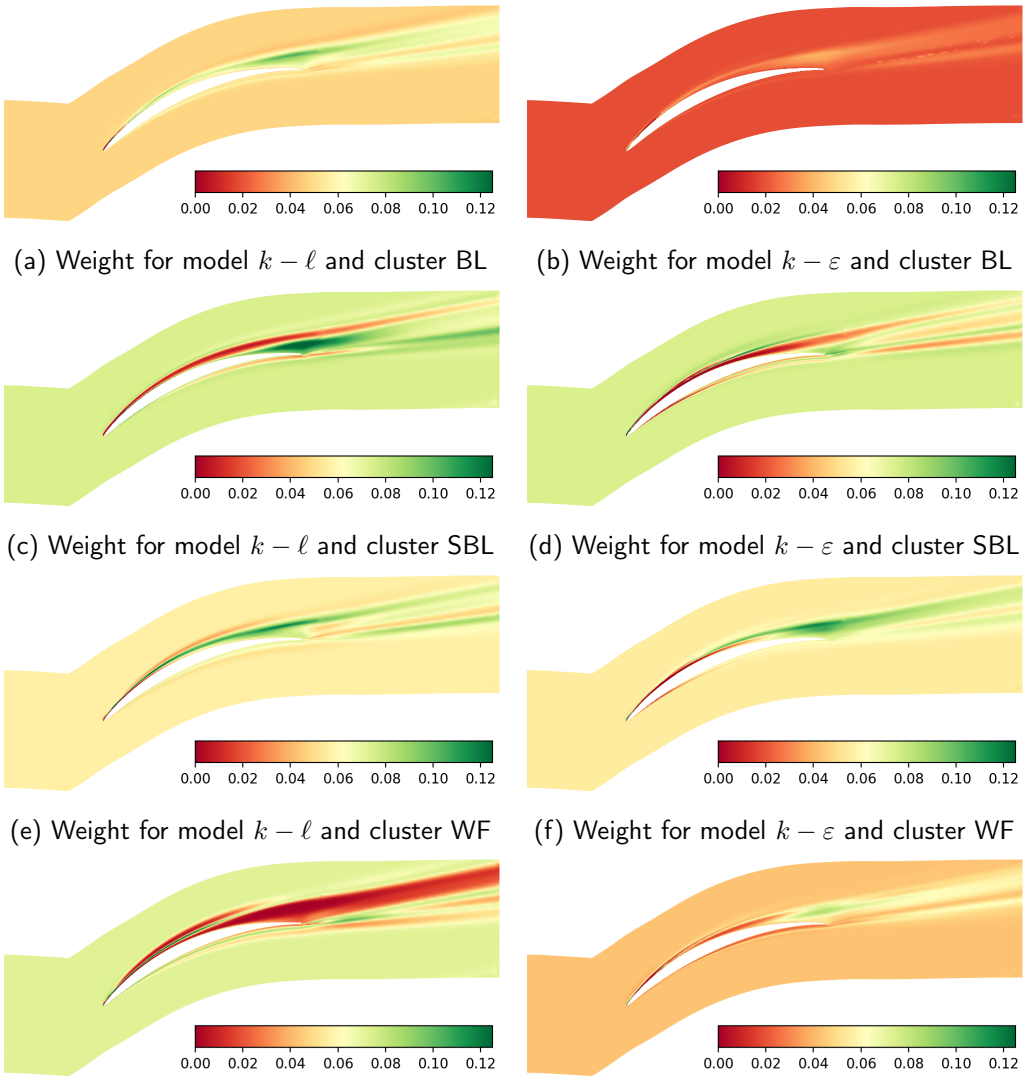
After computing the weights using the XUMA weighting method and regressing them in the feature space, as explained in the previous chapter, it is possible to predict the weights for 16 updated models (4 models in 4 clusters) in the calibration and weighting scenario S_4. The weights for the $k-\ell$ and $k-\varepsilon$ updated models are displayed in figures 6.29a to 6.29h.

There is significant variation in both global and local trends of the weights, depending on the model and the calibration cluster. For instance, the models calibrated on the wake, shown in figures 6.29e and 6.29f, are assigned higher weights in the wake area compared to other regions.

Overall, figures 6.29a to 6.29h suggest that models tend to be favoured (i.e., given a higher weight) in the physical dynamics on which they were calibrated. However, some models, like the $k-\varepsilon$, appear to be generally disfavoured depending on the calibration cluster. As mentioned in the previous chapter, this can be attributed to a higher hyper-parameter σ_η value assigned to the model during the calibration step, indicating that the model is less trustworthy according to the calibration process.

Using the expressions provided in the previous chapter, we can calculate the expected value and standard deviation of the total pressure field, as well as the relative error of the prediction when applying the XUMA weighting method. The expected value and standard deviation are shown in figures 6.30a and 6.30b, respectively.

Notably, the standard deviation in the XUMA method is lower than that obtained using the CMA weighting method, as shown in figure 6.22b. This reduction is due to XUMA's ability to select the best-performing mixture of models at each point in the flow. By assigning significantly



(a) Weight for model $k - \ell$ and cluster BL

(b) Weight for model $k - \varepsilon$ and cluster BL

(c) Weight for model $k - \ell$ and cluster SBL

(d) Weight for model $k - \varepsilon$ and cluster SBL

(e) Weight for model $k - \ell$ and cluster WF

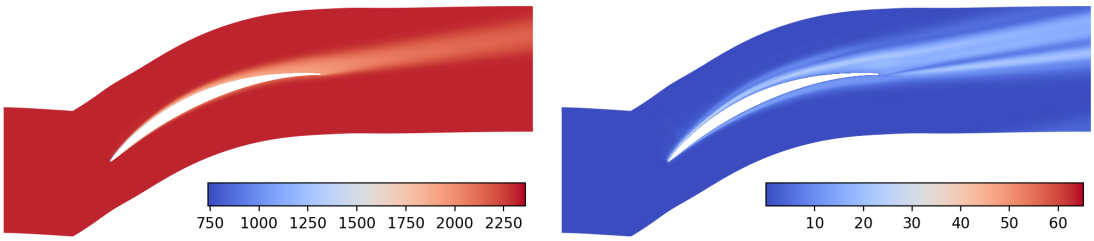
(f) Weight for model $k - \varepsilon$ and cluster WF

(g) Weight for model $k - \ell$ and cluster INV

(h) Weight for model $k - \varepsilon$ and cluster INV

Figure 6.29 – Weights of the $k - \ell$ and $k - \varepsilon$ updated models on the four clusters using the XUMA weighting method after training and prediction on scenario S_4

lower weights to less accurate models, the XUMA method minimises their impact on the overall standard deviation. In contrast, the CMA weighting method does not discriminate as effectively among models, leading to a higher influence of each model on the overall variance calculation.



(a) Expected value (b) Standard deviation

Figure 6.30 – Total pressure field expected value and standard deviation after training and prediction on scenario S_4.

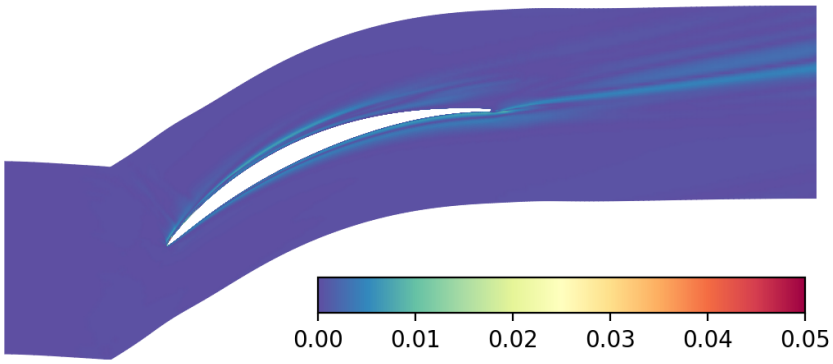


Figure 6.31 – Relative error on total pressure field after training and prediction on scenario S_4.

Figure 6.32 displays the MSEs of the baseline and updated models alongside the CMA and CMA *true*, and the XUMA and XUMA *true* predictions. First, it is important to highlight that, as expected, the CMA and XUMA predictions are as accurate as their respective *true* predictions. This indicates that both the CMA prediction step and the weight prediction step using the RFR are functioning correctly, as we are predicting the training scenario. Furthermore, the XUMA prediction significantly outperforms the CMA prediction, with the MSE of XUMA being only one-third of that of CMA. This substantial reduction in error is achieved by the XUMA weighting method's ability to select the best-performing mixture of models at each point in the flow. In contrast, the CMA weights are averages of the space-dependent weights within the prediction cluster. Even if a model's performance varies within the cluster, this variation will not be captured in the cluster-wide prediction. While this could be seen as a limitation of the CMA method, it does reduce the risk of overfitting by relying on cluster-wide predictions rather than predictions

at each point in the flow.

Finally, it is worth noting that the XUMA method using the cluster-updated models appears to perform almost twice as well as the XUMA method using globally updated models-presented in chapter 5, section 5.3. Specifically, the MSE of the former (≈ 31) is significantly lower than the latter (≈ 47). This suggests that within-cluster calibration enables strong specialisation of the models, allowing for better discrimination and, consequently, improved predictions.

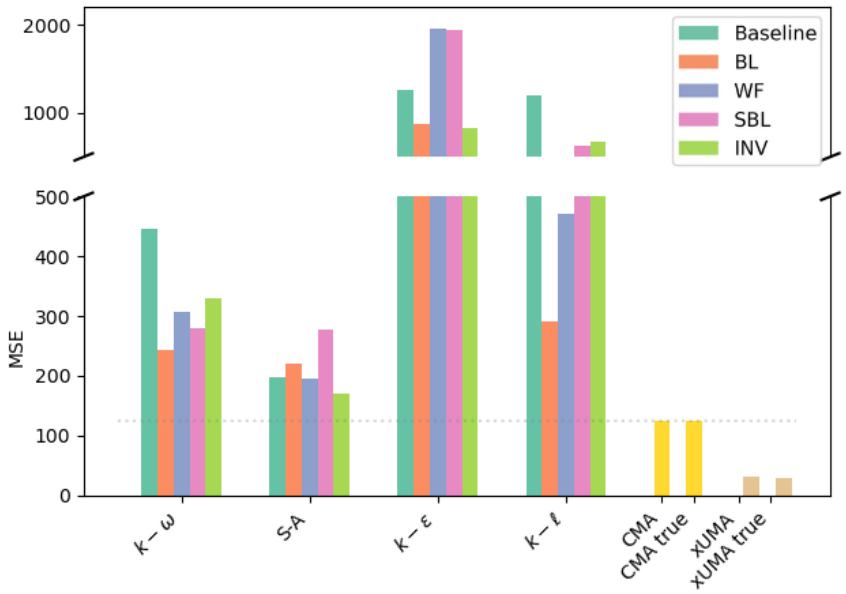
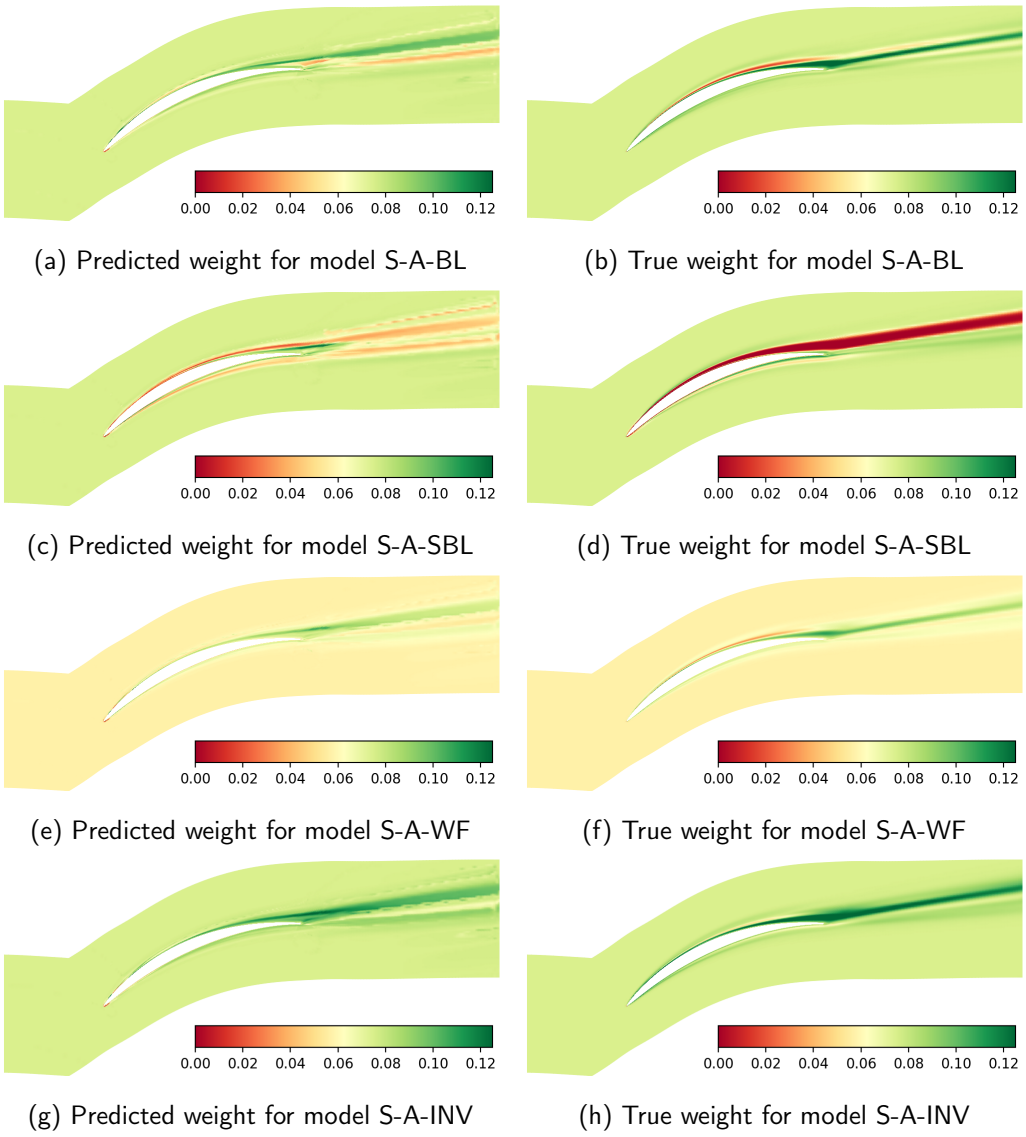


Figure 6.32 – MSE of the reconstruction of the baseline and updated models and CMA predictions after training and prediction on scenario S_4

Predictions on an unseen scenario

Using the RFR trained on the XUMA weights computed on S_4, the weights of the updated models can be predicted in scenario S_3. The comparison between the predicted weights and the *true* weights computed using the synthetic observed data on S_3 for the S-A model is displayed on figures 6.33a to 6.33h.

There is a significant difference between the true model weights and the predicted weights. This discrepancy suggests potential limitations in the predictive performance of the RFR. Several factors could contribute to this disparity. However, given that the RFR training was conducted



(a) Predicted weight for model S-A-BL

(b) True weight for model S-A-BL

(c) Predicted weight for model S-A-SBL

(d) True weight for model S-A-SBL

(e) Predicted weight for model S-A-WF

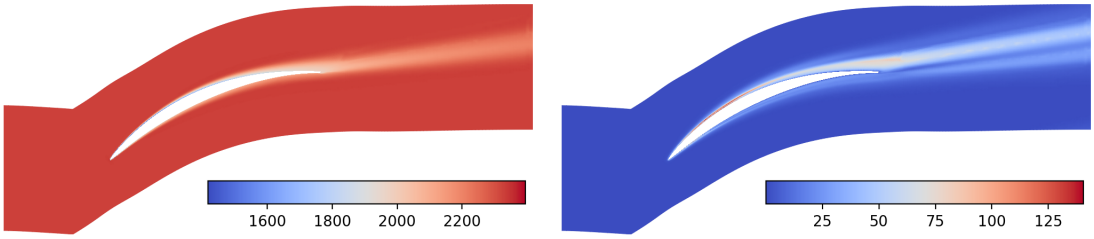
(f) True weight for model S-A-WF

(g) Predicted weight for model S-A-INV

(h) True weight for model S-A-INV

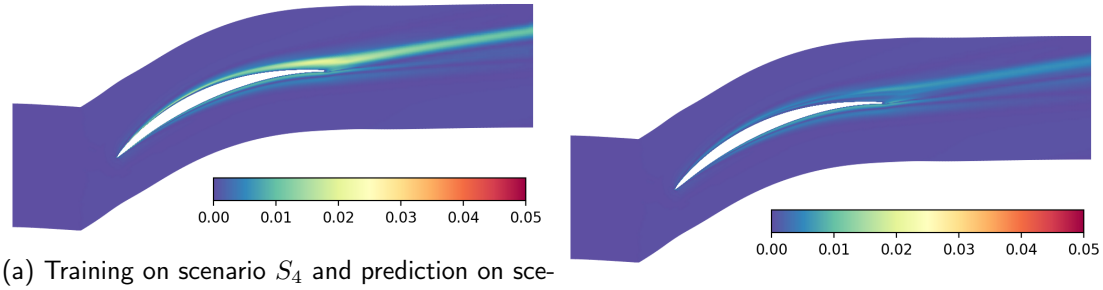
Figure 6.33 – Predicted and *true* weights of the S-A model on the 4 different clusters using the XUMA weighting method after training on scenario S_4 and prediction on scenario S_3.

with careful attention to multiple cross-validation steps, one possible explanation is that the feature set used in this study might not fully capture the variability in the weights. As a result, the XUMA MSE reduction in scenario S_3 is likely to be less important than in the training scenario.



(a) Expected value (b) Standard deviation

Figure 6.34 – Total pressure field expected value and standard deviation after training on scenario S_4 and prediction on scenario S_3 using the XUMA weighting method



(a) Training on scenario S_4 and prediction on scenario S_3

(b) Training and prediction on scenario S_3

Figure 6.35 – Relative error on total pressure field for XUMA and XUMA *true* predictions on scenario S_3

The expected value and standard deviation are computed on scenario S_3 and displayed on figures 6.34a and 6.34b along with the relative error of the XUMA and the XUMA *true* relative error predictions on figures 6.35a and 6.35b. As expected by the disparity between the predicted and *true* weights, the prediction performs less well than using the predicted weights than the true weights. Also, figure 6.36 suggests that even though the *true* XUMA prediction is still way more accurate in terms of MSE than any updated or baseline models or the CMA predictions, the XUMA prediction not only do not perform better than any models but do not even perform as good as the CMA predictions.

Additionally, the XUMA *true* using the cluster-updated model seems to perform twice as well as the XUMA *true* using globally updated models, as the MSE of the fist (≈ 29) is half as small as the MSE of the second (≈ 52). However, the same thing cannot be said for the XUMA prediction using the interpolated weights as the cluster update XUMA prediction performs significantly worse than its globally updated counterpart (MSE of 98 versus 60). The possibility

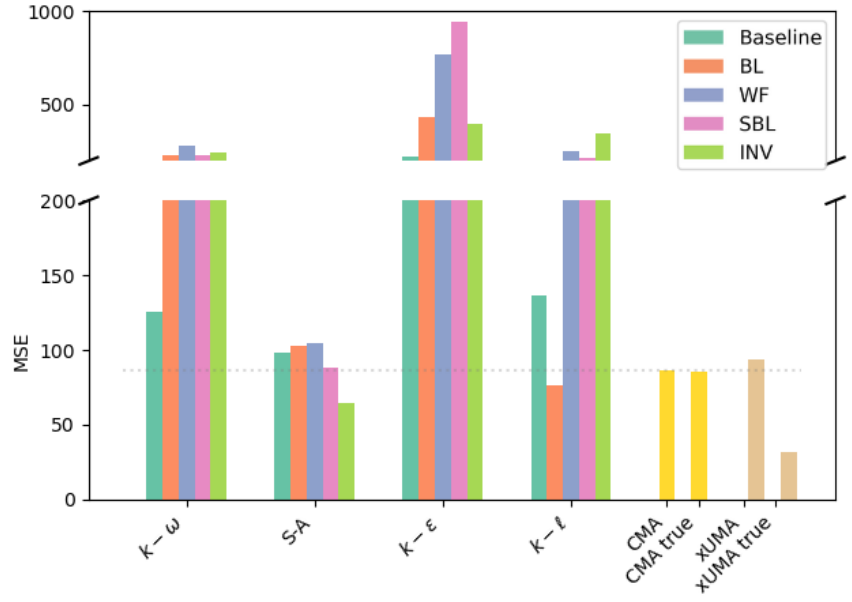


Figure 6.36 – MSE of the reconstruction of the baseline and updated models and CMA predictions after training on scenario S_4 and prediction on scenario S_3 w.r.t the reference data

that the space-dependent weighting might be overfitting the data to provide the best possible scenario regardless of its poor generalisability performances has already been discussed in the previous chapter. What should also be pointed out is that, previously, the discrepancy in MSE between the XUMA and XUMA *true* predictions was around 20% of the MSE, in this chapter the XUMA prediction is almost 3 times greater than the XUMA *true* MSE. One explanation could be that because the models are cluster-updated, they are more specialised to predict the physical dynamic on which they have been calibrated. This means that outside of their calibration context, they should not be trusted as no data from outside of the cluster was used to temper the specialisation and allow a satisfactory prediction outside of the calibration physical dynamic. As the weights are not generalisable as suggested by the figures 6.33a to 6.33h, some models will be used even though they should not, leading to poor predictive performances. This ultimately indicates that the results of XUMA should be used with caution when applied to a different scenario or configuration than the one used for training. While there is currently no evidence to suggest that the XUMA prediction would perform worse than the best baseline model, the training and testing configurations remain quite similar. Therefore, it cannot be stated that this

performance will hold in a more distinct testing configuration.

Figure 6.36 finally suggests that, although the CMA's predictive performance in terms of MSE is not as strong as the XUMA prediction in the training scenario, it is significantly less affected by changes in the scenario and demonstrate better generalisability properties compared to the XUMA method.

6.4.4 Section summary

This section presents the Clustered Mixture of Models (CMA) methodology and provides a detailed comparison with the previously introduced space-dependent method (XUMA). The CMA method combines cluster-updated models to generate smooth, specialised predictions, which are then aggregated to form a global prediction. Fuzzy clustering methods, such as GMMs, are employed to ensure continuity and smoothness across the predictions. The cluster-dependent approach has proven to be robust, particularly for predictions in unseen configurations, by organising data into clusters that capture distinct physical dynamics.

Key takeaways include:

1. **CMA Performance and Generalisation**:
   - CMA predictions show a significant reduction in error compared to baseline models, particularly within training scenarios. When applied to unseen scenarios, CMA predictions remain effective, with cluster-derived weights from training scenarios performing well in testing scenarios. This highlights the crucial role of clustering in the model, as the impact of clustering on performance is more pronounced than within-cluster weighting variations.
   - CMA's implementation is efficient in terms of computational and storage costs, as it avoids the need for training and storing complex interpolators, making it a more scalable solution for complex fluid dynamics simulations.
2. **Comparison Between CMA and XUMA**:
   - The CMA method uses cluster-wide average weights, simplifying implementation and reducing the risk of overfitting by avoiding overly precise local predictions. In contrast, XUMA applies point-specific weights, offering more precision but at a higher computational cost.
   - XUMA outperforms CMA in terms of reducing Mean Squared Error (MSE) by effectively selecting the best model mixture for each spatial point. However, XUMA's

performance drops significantly in unseen scenarios, indicating potential overfitting to training data and/or suffering from the limitation of the RFR using the chosen set of features as mentioned in the previous chapter, while CMA offers more robust and generalisable performance across different scenarios.

Finally, the section underscores the need for ongoing research to refine clustering algorithms, develop new features, and integrate advanced machine learning techniques to enhance the predictive accuracy of these methods, particularly in more complex configurations such as 3D geometries.

6.5 Chapter summary

This chapter introduced a novel model calibration and aggregation approach aimed at addressing the limitations of global calibration on full-field data, as discussed in Chapter 5. The proposed methodology uses clustering techniques to create specialised models for different physical dynamics, thereby improving model accuracy by avoiding over-specialisation in certain flow regions. By calibrating models within clusters of similar physical dynamics and aggregating these specialised predictions, the method seeks to reconstruct a comprehensive global prediction. The use of fuzzy clustering ensures smooth transitions between clusters, enhancing prediction accuracy and generalisation.

The methodology involves three main steps: offline clustering to identify physical dynamics, in-cluster model calibration, and cluster-dependent aggregation of predictions. The first step employs clustering algorithms, notably the Gaussian Mixture Model (GMM), to segment the flow domain into regions with similar physical dynamics. This technique creates clusters that capture distinct flow characteristics, such as boundary layers, wakes, and free streams. The GMM effectively differentiates between these regions, which is essential for tailoring models to the specific dynamics within each cluster. The novel set of features developed in this chapter plays a crucial role in ensuring that clustering predictions generalise across different flow configurations. Building upon Callaham's prior work [15], these new features are non-dimensional and Galilean-invariant, making them applicable to a variety of flow domains. This enhancement allows the GMM to manage diverse flow configurations within a unified framework. The results indicate that this approach is promising: the GMM accurately identifies flow regions in training scenarios and the more complex, unseen NACA65 configuration. This success highlights the potential of the new features to improve generalisability between simpler and more complex scenarios. Nevertheless, while the advancements are significant, they also point to the need for further development. Continued refinement of the features is essential to ensure that the clustering approach consistently provides robust and accurate predictions across a wider range of fluid dynamics problems, including more complex 3D configurations.

In the calibration step, models are adjusted for each cluster to better fit the specific data from that region. This focused approach enhances the models' ability to capture the unique flow dynamics of each cluster, resulting in lower errors for measurements within those areas. However, this specialisation comes with limitations. Models calibrated for specific clusters may

not perform as well outside their designated regions. While calibration improves accuracy within the clusters, it can lead to higher errors when these models are applied to different flow conditions. Despite these improvements, challenges remain. The models do show some improvement in predicting new, unseen scenarios, but their performance can vary, especially for quantities not included in the calibration. To address these issues, future work should involve calibrating the models for a broader range of quantities and evaluating their predictive performance. Additionally, model aggregation remains essential. It ensures that the specialised models are used effectively, maximising their accuracy in their intended regions and minimising errors in other areas.

The CMA methodology uses cluster-wide average weights to combine predictions from different clusters, creating a smooth and unified global prediction. This approach greatly reduces errors compared to baseline models and works well even in new or unseen scenarios. It is also more efficient, with lower computational and storage costs, compared to the earlier space-dependent method, XUMA. XUMA provides precise local predictions by using point-specific weights, but this requires detailed interpolation with a Random Forest Regressor (RFR). This process is complex and expensive, making XUMA less practical for some applications. While XUMA performs well within known scenarios, it struggles with new scenarios and is more prone to overfitting. In summary, the CMA method offers a more practical and adaptable solution for model aggregation. It balances accuracy with efficiency, making it a good choice for a wide range of fluid dynamics problems.

Overall, this chapter demonstrates that the integration of clustering with model calibration and aggregation significantly enhances CFD predictions. The new methodology improves prediction accuracy and generalisation across various scenarios compared to traditional methods. However, ongoing research is needed to further refine clustering algorithms, develop new features and/or incorporate advanced machine learning techniques to address challenges in more complex configurations, such as 3D geometries. This continued effort will be crucial for advancing the field and achieving reliable and accurate predictions across a wider range of applications.

7

Building-block model aggregation

The previous chapters suggest that aggregation of locally updated models (such as CMA) exhibit better generalisation properties than aggregates of globally calibrated models (as in XUMA) because they take into account physical processes, such as boundary layers, wakes, or separated regions, that can be encountered in any other flow, instead of inferring a set of parameters that constitutes a compromise solution among different model requirements for the flow configuration used for training. While promising, CMA requires a preliminary clustering step both for calibration and for training, which is in turn highly dependent on the set of features used to define the clustering space. The potential advantage of this is that, for flows involving complex features, regions of interest are automatically extracted from the data. The drawback is that a full field of data is needed to compute the features and train the clustering algorithm, which is only possible for flow configurations for which full fields of high-fidelity data are available. As an alternative, we imagine a strategy where sets of calibration data representative of various dominant flow dynamics are chosen by expert judgement, and used to calibrate expert models,

thus avoiding the clustering step.

Chapter contents

In the remainder of this chapter, we present a methodology that, similar to CMA, generates specialised models, or "experts", each calibrated to capture different physical dynamics. However, instead of using a clustering algorithm to generate subsets of data, it relies on a set of so-called "building-block" flows for doing the calibrations. The use of such "building blocks" is rather standard in turbulence model calibration, classical turbulence model closure coefficients being calibrated for a set of "canonical" flows (such as homogeneous and isotropic turbulence, free shear layers, and attached wall-bounded flows), see e.g. [83]. The parameters are calibrated on all those flows, representative of limit states of turbulence, simultaneously, with the idea of building universally valid models. In this case, instead, the calibration aims at specialising the models for capturing a given physical process with the best possible accuracy (within the limitations due to model structure), while model aggregation is used to flexibly combine the local experts, overcoming the structural limitations of the individual component models, and ensuring generalisation capabilities. The terminology "building-block" flows has been proposed initially by Lozano-Duran *et al.* [65], who generated data-driven wall model for large-eddy simulations (LES) by devising the flow as a combination of building blocks under the assumption that a finite set

of simple flows contains the essential physics to predict the wall shear stress in more complex scenarios. The "canonical" flows traditionally used in turbulence modelling being insufficient to inform models for such scenarios, the data set is enriched with additional flows, e.g. representative of adverse pressure gradients or cross-flow effects.

Hereafter, we build on such an idea by defining a set of building block flows, and by calibrating specialised models for different physical processes. Since no clustering probabilities are available in this case, the competing models are then aggregated using the XUMA approach.

7.1 Methodology

The proposed methodology consists of three steps :

1. Calibration of a set of competing RANS models for selected building block flows, leading to specialised models;
2. training of weighting functions for model aggregation;
3. application of the aggregated model for the prediction on a new configuration.

The three steps are described in detail in the following.

7.1.1 Calibration of RANS models on the building blocks

Consider a set $\mathcal{M} = \{M_1, \dots, M_m, \dots, M_{N_M}\}$ of N_M models, each characterised by a vector of parameters $\boldsymbol{\theta}_m$ that are to be calibrated against N_{BB} building blocks. Finally, consider the observed data $\Delta_{obs} = \cup_{b=1}^{N_{BB}} \Delta_{obs,b}$ used to calibrate the different models, corresponding to the collection of observations available for each building block, $\Delta_{obs,b}$.

The N_M baseline models are calibrated individually against each one of the N_{BB} building blocks by using the Bayesian framework of Chap. 3, leading to $N_m \times N_{BB}$ updated models, characterised by the MAP estimates $\boldsymbol{\theta}_{m,b}^{MAP}$ of the posterior distributions $p(\theta|\Delta_{obs,b}, M_m)$ of model M_m calibrated for the building block BB_b. We call them *building-block models*. As done throughout this work, the MAP estimates are used to approximate the full posterior and propagated through the flow solver to formulate an a posteriori estimate of some QoI for a given flow configuration.

For a given building block, the N_M models are used to quantify structural model uncertainty at capturing that particular physical process, while the N_{BB} sets of MAP parameters for a given model are used to locally fit the model to different flow processes across some physical feature

space. The next step consists then in aggregating all such models into a single hyper-model by using the XUMA approach.

7.1.2 Space-dependent aggregation of building-block models

We now want to use the $N_M \times N_{BB}$ building-block models to produce an aggregated prediction of a quantity of interest Δ^*. The individual model predictions will all differ because of different model structures and choice of calibration data, but we expect that, if Δ^* corresponds to a physical process included in one of the building blocks, then models calibrated for such building block should capture it more accurately than those that are not and should be assigned higher weights in the model mixture.

We then adopt the methodology introduced in chapter 5, and we use local posterior probabilities to weight the different updated models in a space-dependent fashion.

For that aim, let us consider a quantity Δ_W used for learning the model weighting functions. Such data is observed at some location $\boldsymbol{\eta}_W$ of a feature space. Building on the XUMA weighting method from 5, we define the local model posterior plausibility at the weighting location $\boldsymbol{\eta}_W$:

$$p(M_{m,b}|\Delta_W, \boldsymbol{\eta}_W) = \frac{p(\Delta_W, \boldsymbol{\eta}_W|M_{m,b})p(M_{m,b})}{\sum_{k=1}^{N_M}\sum_{b=1}^{N_{BB}} p(\Delta_W, \boldsymbol{\eta}_W|M_{k,b})p(M_{k,b})} \tag{7.1}$$

where $M_{m,b}$ is the M_m model updated on building block BB_b, $p(M_m)$ are the prior model plausibilities (generally assumed to be uniform) and where $p(\Delta_W, \boldsymbol{\eta}_W|M_{m,b})$ is a shortcut notation for $p(\Delta_W, \boldsymbol{\eta}_W|\boldsymbol{\theta}_{m,b}^{MAP}, M_m)$ is the point-wise marginal likelihood under the MAP approximation.

We then weigh each model in the mixture by its corresponding local plausibility, and note:

$$W_{m,b,W} = p(M_{m,b}|\Delta_W, \boldsymbol{\eta}_W) \tag{7.2}$$

the weight of model M_m updated on building block BB_b at the weighting location $\boldsymbol{\eta}_W$.

The estimated weights $W_{m,b,W}$ are known only at locations in the feature space where the data are observed, which in turn have been extracted at observation locations in the physical space, x_W, for one or more flow configurations used to train the weighting functions. To be able to estimate the weights at locations not included in the training set (corresponding to different

spatial locations in a given flow or a different flow configuration), the feature-weight relationship is regressed across the feature space by using a Random Forest Regressor. As discussed previously in Chap. 5.1, the Random Forest hyper-parameters are defined using a grid search. In the following numerical experiments, we adopt again the set of features described in Table 5.1. At the end of the procedure, a set of weighting functions:

$$w_{m,b} = w_{m,b}(\boldsymbol{\eta}_W) \tag{7.3}$$

is obtained, where $w_{m,b}$ denotes the weighting function associated with model $M_{m,b}$, and $\boldsymbol{\eta}_W$ corresponds to the features estimated under model $M_{m,b}$.

The regressed model weights are finally used to predict any QoI Δ^* at any location $\boldsymbol{\eta}^*$ by using the model aggregation formula. The aggregation procedure follows the XUMA approach, with each of the $N_M \times N_{BB}$ updated models being considered as an independent component model :

$$p\left(\Delta^* \mid \Delta_{obs}, \Delta_W, \boldsymbol{\eta}^*\right) = \sum_{m=1}^{N_M} \sum_{b=1}^{N_{BB}} p\left(\Delta^* \mid M_{m,b}, \Delta_{obs}, \boldsymbol{\eta}^*\right) w_{m,b}(\boldsymbol{\eta}^*) \tag{7.4}$$

The latter aggregated distribution can be used to estimate the aggregated expectancy and variance:

$$\mathbb{E}\left(\Delta^* \mid \Delta_{obs}, \Delta_W, \boldsymbol{\eta}^*\right) = \sum_{m=1}^{N_M} \sum_{b=1}^{N_{BB}} \mathbb{E}\left(\Delta^* \mid M_{m,b}, \Delta_{obs}, \boldsymbol{\eta}^*\right) w_{m,b}(\boldsymbol{\eta}^*) \tag{7.5}$$

$$\mathrm{Var}\left(\Delta^* \mid \Delta_{obs}, \Delta_W, \boldsymbol{\eta}^*\right) = \sum_{m=1}^{N_M} \sum_{b=1}^{N_{BB}} \left(\mathbb{E}\left(\Delta^* \mid M_{m,b}, \Delta_{obs}, \boldsymbol{\eta}^*\right) - \mathbb{E}\left(\Delta^* \mid \Delta_{obs}, \boldsymbol{\eta}^*\right)\right)^2 w_{m,b}(\boldsymbol{\eta}^*) \tag{7.6}$$

where the variance expresses as usual the consensus among the estimates of Δ^* provided by each competing model, and where we used the MAP approximation to remove the contribution of the within-model variances in equation 7.6.

7.1.3 Summary of the method

The method consists of three steps :

1. **Calibrate** a set of N_M models, characterised by parameters $\boldsymbol{\theta}_m$ against N_{BB} building block using data $\Delta_{obs,b}$ observed for each block and **Approximate** the posterior distributions as

Dirac distributions centred around the MAP estimate $\boldsymbol{\theta}_{m,b}^{MAP}$.

2. **Estimate** a set of space-dependent model weighting functions
   - — **Predict** the quantity of interest Δ_W with the $N_M \times N_{BB}$ updated models using the MAP estimates of the parameters $\boldsymbol{\theta}_{m,b}^{MAP}$.
   - — compute the exact model weights $W_{m,b}$ using an additional set of data Δ_W
   - — Regress the space-dependent weights across the feature space defined in Table 5.1 using a Random Forest Regressor
3. **Infer** model weights and **Aggregate** building-block models to make predictions
   - — **Predict** the quantity of interest Δ^* with the $N_M \times N_{BB}$ updated models ;
   - — **Compute** the features $\boldsymbol{\eta}_{m,b}$;
   - — **Aggregate** all $N_M \times N_{BB}$ predictions and estimate the posterior expectancy and variance with equation (5.7)-(5.8)

In the next section, we illustrate the potential of the proposed framework for the prediction of a three-dimensional turbomachinery configuration, namely, the 3D NACA65 V103 linear compressor cascade presented in Chapter 4.2.2.

7.2 Calibration of RANS models for building-block flows

7.2.1 Observed Data, Models, and Prior Distribution

In this study, four Reynolds-Averaged Navier-Stokes (RANS) models are calibrated using four distinct building blocks that represent the primary physical dynamics of a turbomachinery flow. The four models are two of the linear eddy viscosity models considered in the previous sections, namely the Spalart-Allmaras model [97] and the $k-\ell$ model [95], and their Quadratic Constitutive Relation (QCR) corrections (see 2.2.3). The QCR models were introduced in the mixture because of their better-expected performance in capturing 3D flow phenomena, such as corner vortices.

For each one of the four models, a set of parameters was selected for updating through Bayesian calibration. The prior distributions of these parameters are assumed to be uniform. The baseline values (BV) of the parameters, along with the lower bound (LB) and upper bound (UB) of the uniform distributions for the selected parameters of each model, are reported in Tables 7.1 to 7.2. For the $k-\ell$QCR and the Spalart-Allmaras QCR models, in addition to the parameters

of the underlying standard models, we also calibrate the QCR parameter. The bounds of the *a priori* distribution, presented in Table 7.3 have been chosen based on a private communication by Philippe Spalart, suggesting that greater values of the QCR correction would create non-physical solutions in the flow. To build an aggregated predictor for a three-dimensional linear and

Param.	BV	LB	UB
B1	18	1	36
E2	1.20	0.01	1.60
σ_l	1.43	0.65	3.60
σ_k	1.43	0.20	2.50
κ	0.41	0.36	0.95

Table 7.1 – Uniform prior bounds for the $k-\ell$ model.

Param.	BV	LB	UB
Cb_1	0.1355	0.07	0.15
Cb_2	0.622	0.50	1.00
Cv_1	7.1	6.00	15.0
Cw_2	0.3	0.10	0.90
Cw_3	2.0	0.80	2.40
σ	2./3.	0.60	1.50
κ	0.41	0.36	0.55

Table 7.2 – Uniform prior bounds for the S-A model.

Param.	BV	LB	UB
QCR	0.30	0.00	0.50

Table 7.3 – Uniform prior bounds for the QCR coefficient.

fixed cascade with no tip gap, we chose to calibrate the models on the following building-block configurations, representative of the dominant physical processes encountered in the target flow:

— **Two-dimensional, attached flat plate boundary layer (BL)**: this BB is chosen to represent attached boundary layers; however, effects of favourable or adverse pressure gradients, are not taken into account;

— **Two-dimensional wake flow (WF)**: this BB is chosen to represent the wake regions behind the blade;

— **Backward-facing step (SBL)**: this BB accounts for boundary layer separation; however, the separation point is fixed by the geometry in the training set, which is generally not the case in turbomachinery;

— **Square duct flow (SD)**: this BB is chosen to account for 3D effects in the blade passage, and specifically for the corner vortices arising near the hub and casing.

For each BB case, described in chapter 4, a set of experimental observations is available. For the BL case, we used the four longitudinal velocity profiles taken at locations defined by their Reynolds number based on the momentum thickness, as well as the deduced friction coefficient measurements, resulting in 66 data points. We used interpolated data from four longitudinal measured velocity profiles for the WF case, also accumulating 66 data points. For the SBL case, we extracted longitudinal velocity measurements in the recirculation region from six longitudinal velocity profiles, amounting to 38 experimental data points. Finally, we extracted data from the digitalized iso-contours of the transverse velocity for the SD case, resulting in 210 experimental data points. Only the QCR-corrected models were used for this BB, as the standard linear eddy viscosity models are unsuitable for capturing the secondary flow [108] and cannot be informed from the data.

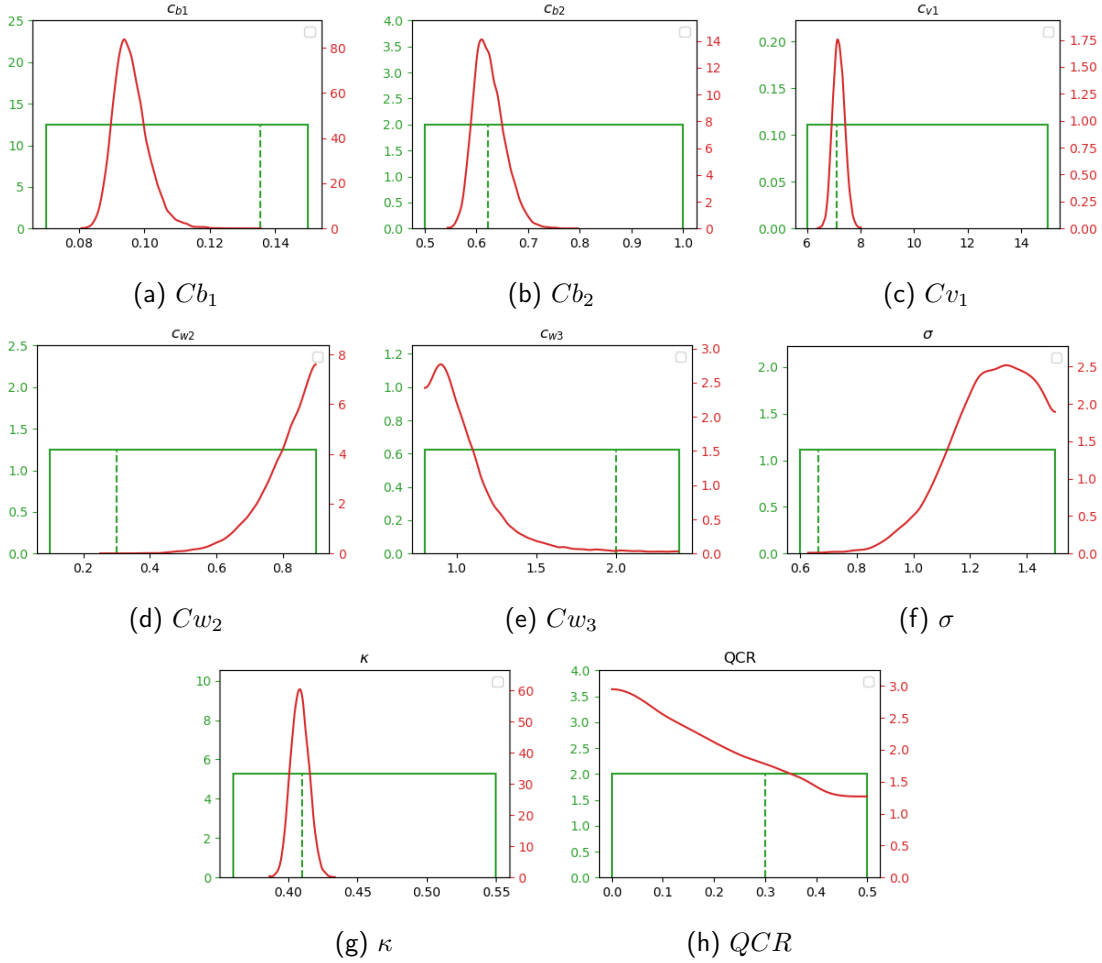
The four models are calibrated on the data available for the various BBs using the same stochastic formulation used previously, resulting in fourteen new sets of parameters—since only two models are calibrated on the square duct corner flow building block.

Calibration results are discussed hereafter.

7.2.2 Posterior distributions

This section displays typical results obtained for the posterior parameter distributions. For brevity, we focus on the SA-QCR model. The MAP estimates for all building block parameters are available in the Appendix A.

The posterior distributions obtained for the boundary layer building block are presented in Figures 7.1a to 7.1h. Many parameters exhibit posterior distributions centred around their baseline values, most notably the Von Kármán κ parameter, which governs the boundary layer. This observation is reassuring as it indicates that the calibration process does not significantly deviate from the baseline values, particularly for critical parameters like κ, given that the turbulent boundary layer is a typical calibration scenario for such models. Additionally, it is important to observe that the posterior distribution of the QCR parameter, shown in Figure 7.1h, is not



(a) Cb_1 (b) Cb_2 (c) Cv_1

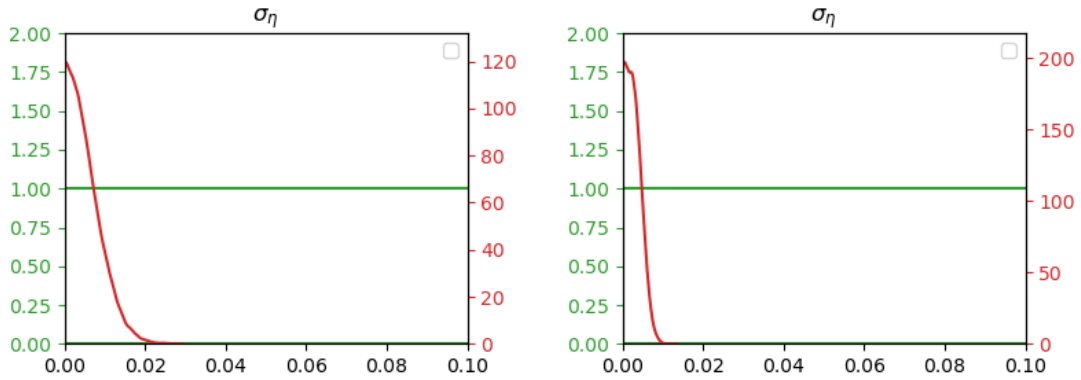
(d) Cw_2 (e) Cw_3 (f) σ

(g) κ (h) QCR

Figure 7.1 – Prior and posterior distributions for the S-A QCR parameters on the BL building block. The dashed line is the baseline value.

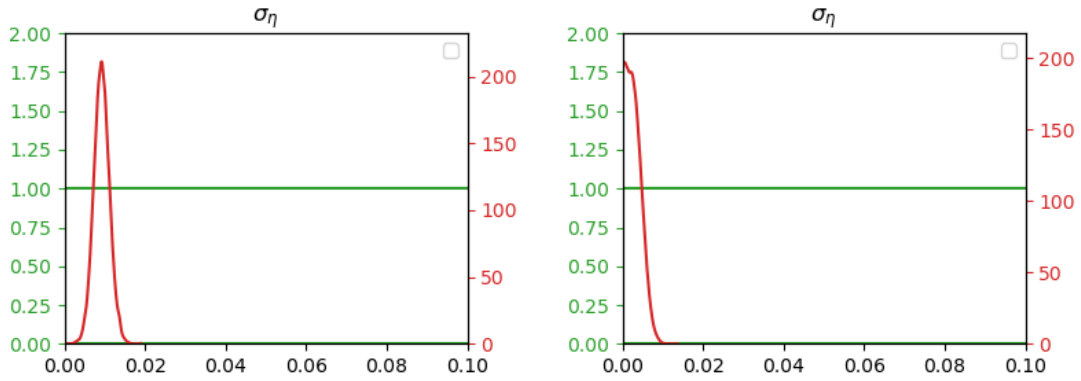
strongly informed, with the mode of the posterior distribution being around zero. This suggests that the QCR parameter is not relevant in this flow (which is expected, the solution being dominated by the shear stress).

Figures 7.2a to 7.2d illustrate the hyper-parameter values for the four different models calibrated on the BL building block. Their relatively small values suggest that the model form errors are low and that the variations in the parameters enable the models to match the experimental data accurately. In other terms, all models capture accurately the BL flow, which often belongs to the "canonical" flows used for the baseline model calibrations. The



(a) $k - \ell$ QCR

(b) S-A QCR

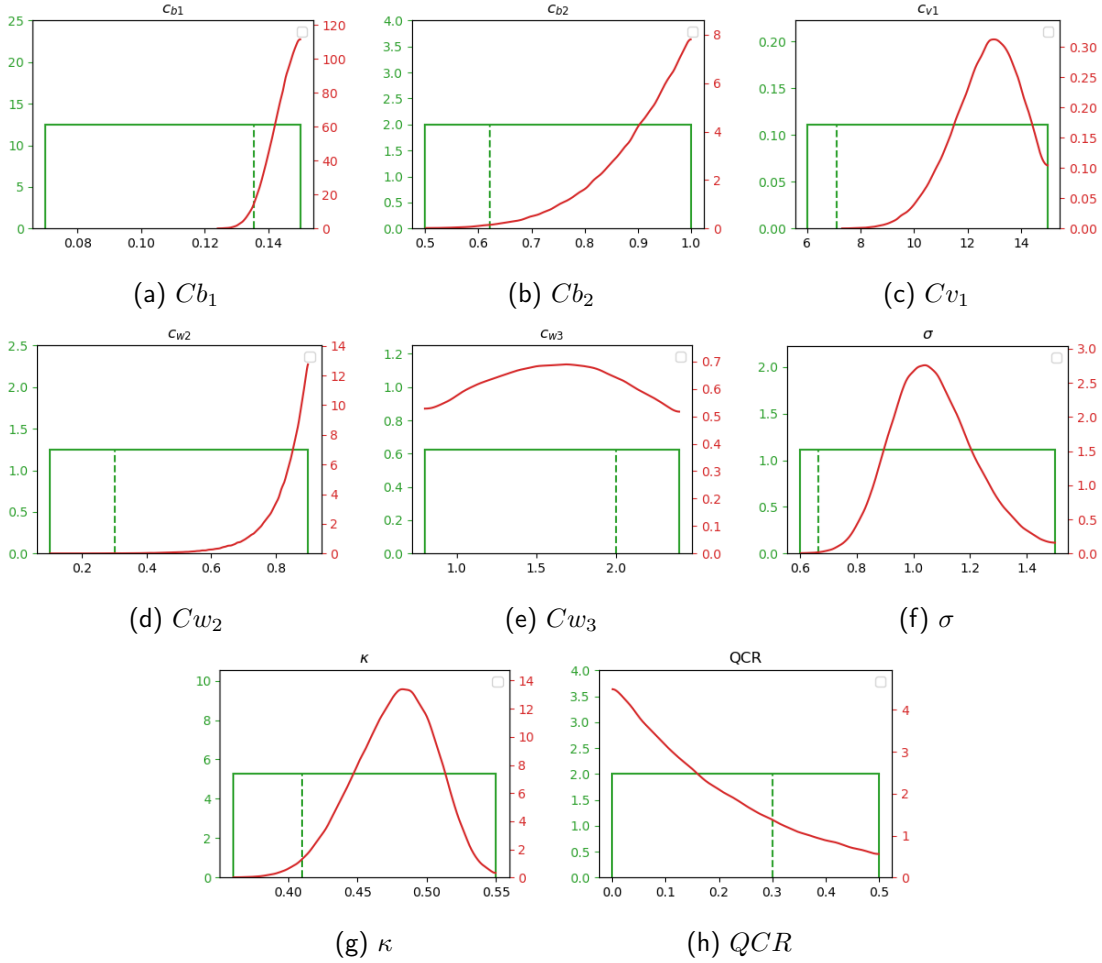


(c) $k - \ell$

(d) S-A

Figure 7.2 – Prior and posterior distributions for the hyper-parameters σ_η on the BL building block. The dashed line is the baseline value.

posterior distributions of the S-A QCR model calibrated on the WF building block are presented in Figures 7.3a to 7.3h. A few parameters exhibit posterior distributions not centred around their baseline values, suggesting that the original values were not suitable to fit accurately this kind of flow. Some parameters, such as the Cw_3 parameter, are not strongly informed, which can be explained by their lack of influence in this configuration. In the Spalart-Allmaras model, this coefficient appears actually in the expression of the damping functions applied near the walls in wall-bounded flows, and the latter is not relevant for free shear flows such as wakes. Additionally, it is important to observe that the posterior distribution of the QCR parameter, shown in Figure 7.3h, is once again not strongly informed, with the mode of the posterior



(a) Cb_1 (b) Cb_2 (c) Cv_1

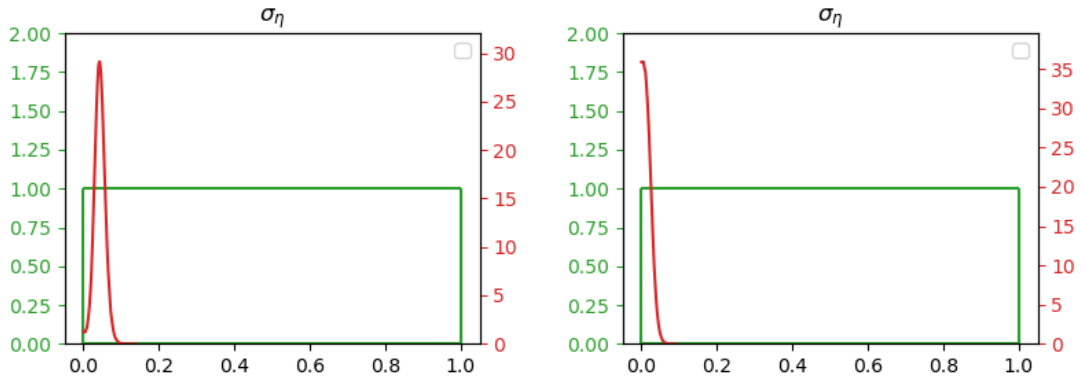
(d) Cw_2 (e) Cw_3 (f) σ

(g) κ (h) QCR

Figure 7.3 – Prior and posterior distributions for the S-A QCR parameters on the WF building block. The dashed line is the baseline value.

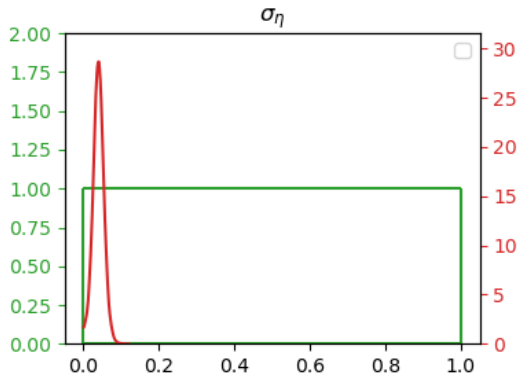
distribution being around zero. This is due once again to the fact that the WF is dominated by the shear stress, and the latter is not affected by the QCR correction.

Figures 7.4a to 7.4d illustrate the hyper-parameter values for the four different models calibrated on the WF building block. Once again, this kind of flow belongs to the class of flows for which most RANS models are designed and tuned, and capturing this flow does not require large adjustments. The small differences compared to the nominal values found for the BL and the WF are attributed to the different data used for the calibrations, and to the fact that the models are calibrated individually against such flows using full Navier–Stokes simulations instead

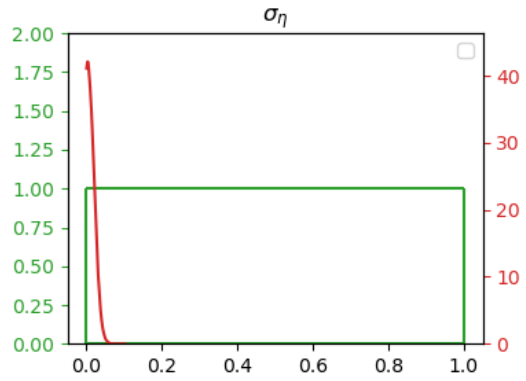


(a) $k - \ell$ QCR

(b) S-A QCR



(c) $k - \ell$

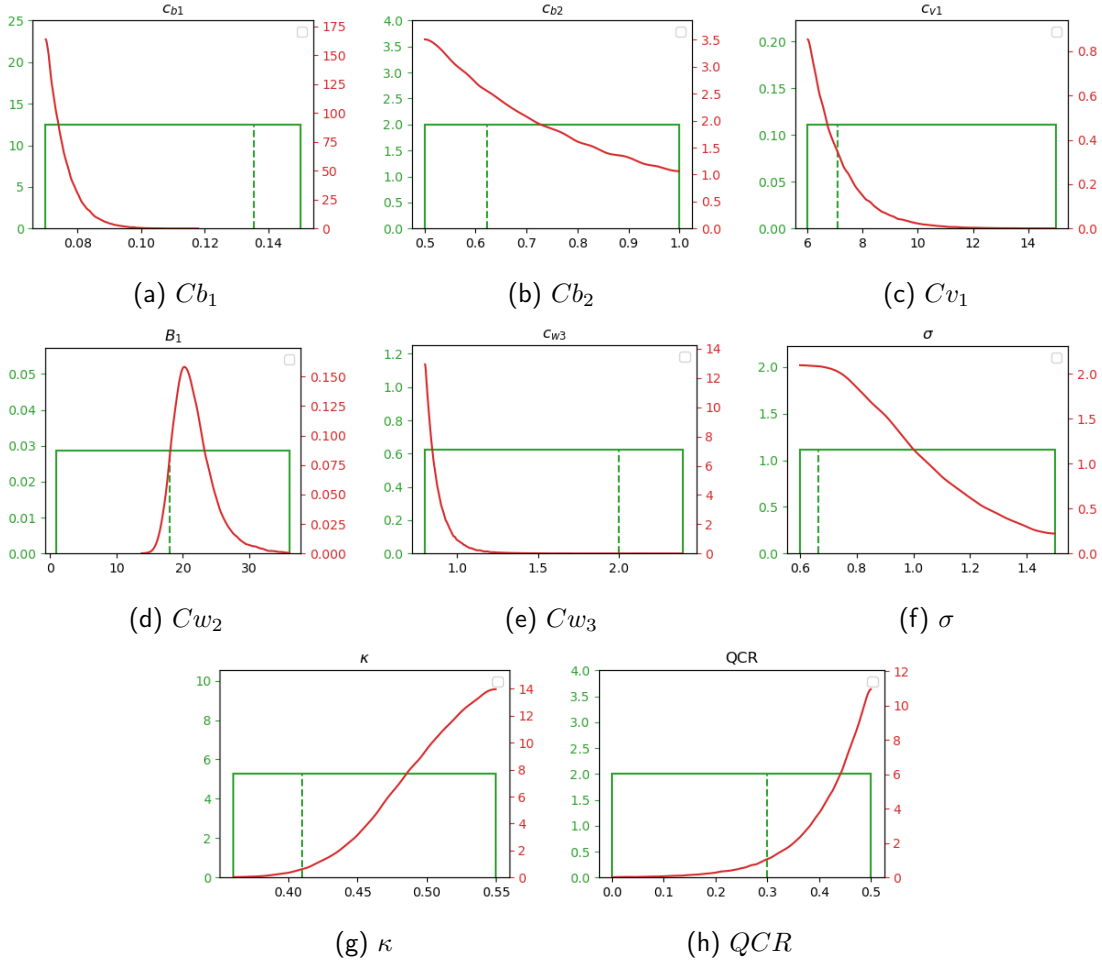


(d) S-A

Figure 7.4 – Prior and posterior distributions for the hyper-parameter σ_η on the WF building block. The dashed line is the baseline value.

of simplified similarity solutions, such as often done for the tuning of the baseline models.

The posterior distributions of the S-A QCR model calibrated on the SBL building block are presented in Figures 7.5a to 7.5h. Some parameters exhibit posterior distributions pushed to the bounds of the prior distribution. This indicates that the bounds are too restrictive or that the model cannot sufficiently adjust the parameters to match the experimental data. The bounds of the prior distribution were chosen based on expert opinion to respect physical constraints and sensibilities. These posterior distributions suggest that the models strive to capture the observed data with the recirculation bubble behind the step. This interpretation is reinforced by the values of the σ_η hyper-parameter posterior distribution for each model, depicted in Figures 7.6a



(a) Cb_1 (b) Cb_2 (c) Cv_1

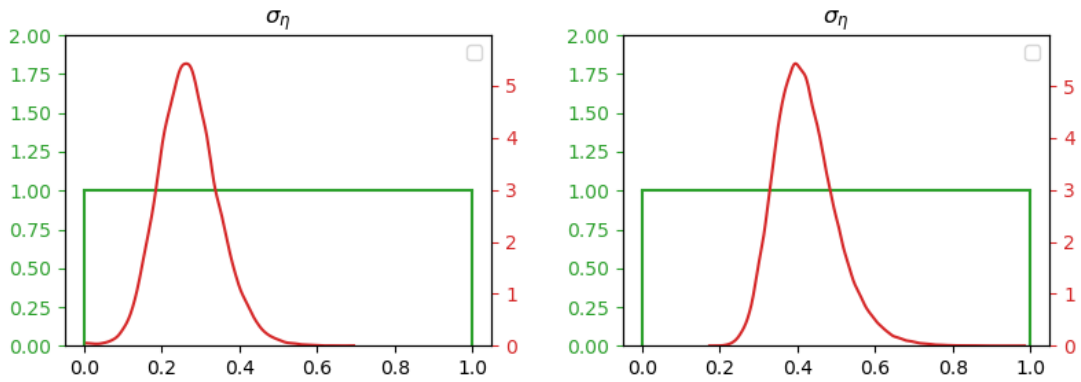
(d) Cw_2 (e) Cw_3 (f) σ

(g) κ (h) QCR

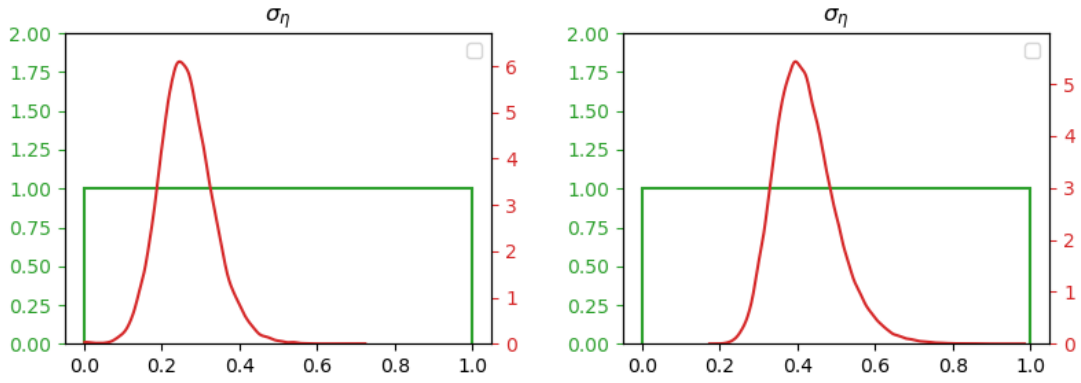
Figure 7.5 – Prior and posterior distributions for the S-A QCR parameters on the SBL building block. The dashed line is the baseline value.

to 7.6d. The distributions are centred around higher values than for the previous building blocks, roughly around 0.2 or 0.4 depending on the model, indicating that the model-form uncertainty is significant. This underscores that the model (similar results are obtained for all the other models) cannot match the experimental data solely by adjusting the parameter values but relies on additional model-form uncertainty to increase the likelihood of capturing the data. Consequently, the models are not to be trusted blindly in this case due to the high model-form uncertainty.

Additionally, it is important to observe that the posterior distribution of the QCR parameter, shown in Figure 7.5h, is informed, with the mode of the posterior distribution being pushed to



(a) $k - \ell$ QCR (b) S-A QCR

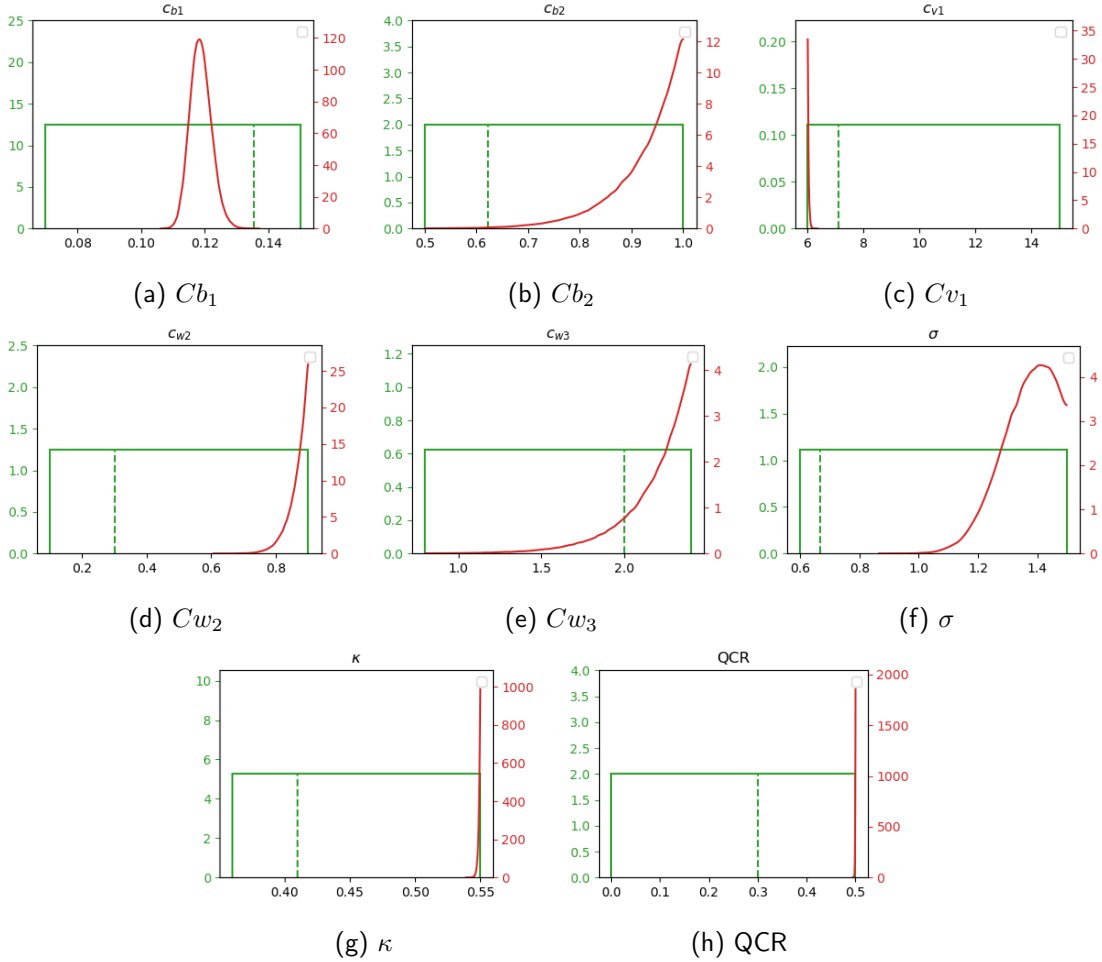


(c) $k - \ell$ (d) S-A

Figure 7.6 – Prior and posterior distributions for the hyper-parameters σ_η on the SBL building block. The dashed line is the baseline value.

the upper bound of the prior distribution. This suggests that the QCR parameter is significant in the calibration process to capture the experimental data, as it adds a quadratic term in the Reynolds stress tensor to better capture turbulence anisotropy.

The posterior distributions of the S-A QCR model calibrated on the square duct corner flow building block are presented in Figures 7.7a to 7.7h. As in the case of the SBL building block, several parameters exhibit posterior distributions pushed to the bounds of the prior distribution. This indicates that the model cannot sufficiently adjust the parameter values to match the experimental data, i.e. the model-form inadequacy is high. Similar to the SBL case, the posterior distributions of the σ_η hyper-parameter posterior distribution (Figures 7.8a to 7.8b) are



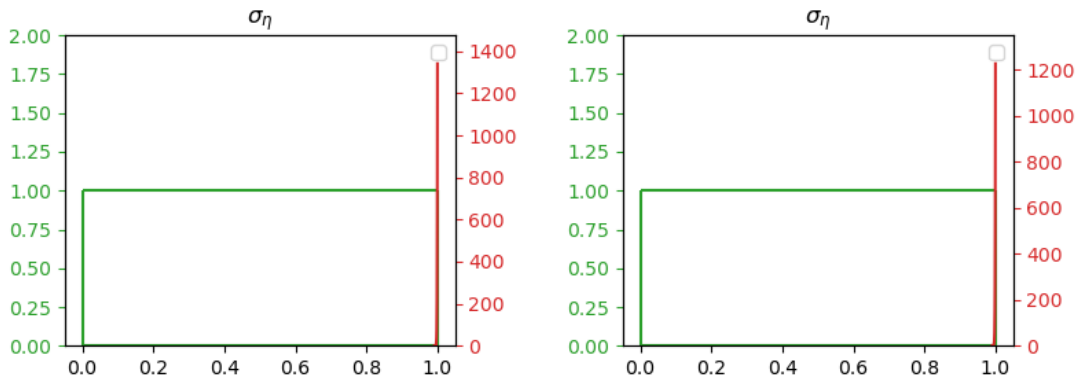
(a) Cb_1 (b) Cb_2 (c) Cv_1

(d) Cw_2 (e) Cw_3 (f) σ

(g) κ (h) QCR

Figure 7.7 – Prior and posterior distributions for the S-A QCR parameters on the SD building block. The dashed line is the baseline value.

completely squashed on the higher bound of the prior distribution, indicating that the model-form uncertainty is very significant. This underscores that the model cannot, even with a QCR correction, capture the experimental data solely by adjusting the parameter values and requires additional model-form uncertainty. Consequently, these models are less trustworthy for this BB than for the others, the model-form uncertainty hyper-parameter being significant. Of note, the posterior distribution of the QCR parameter, shown in Figure 7.7h, is well informed, with the mode of the posterior distribution being pushed to the upper bound of the prior distribution. This suggests that the QCR parameter is paramount in the calibration process to capture the



(a) $k - \ell$ QCR (b) S-A QCR

Figure 7.8 – Prior and posterior distributions for the σ_η hyper-parameter on the SD building block. The dashed line is the baseline value.

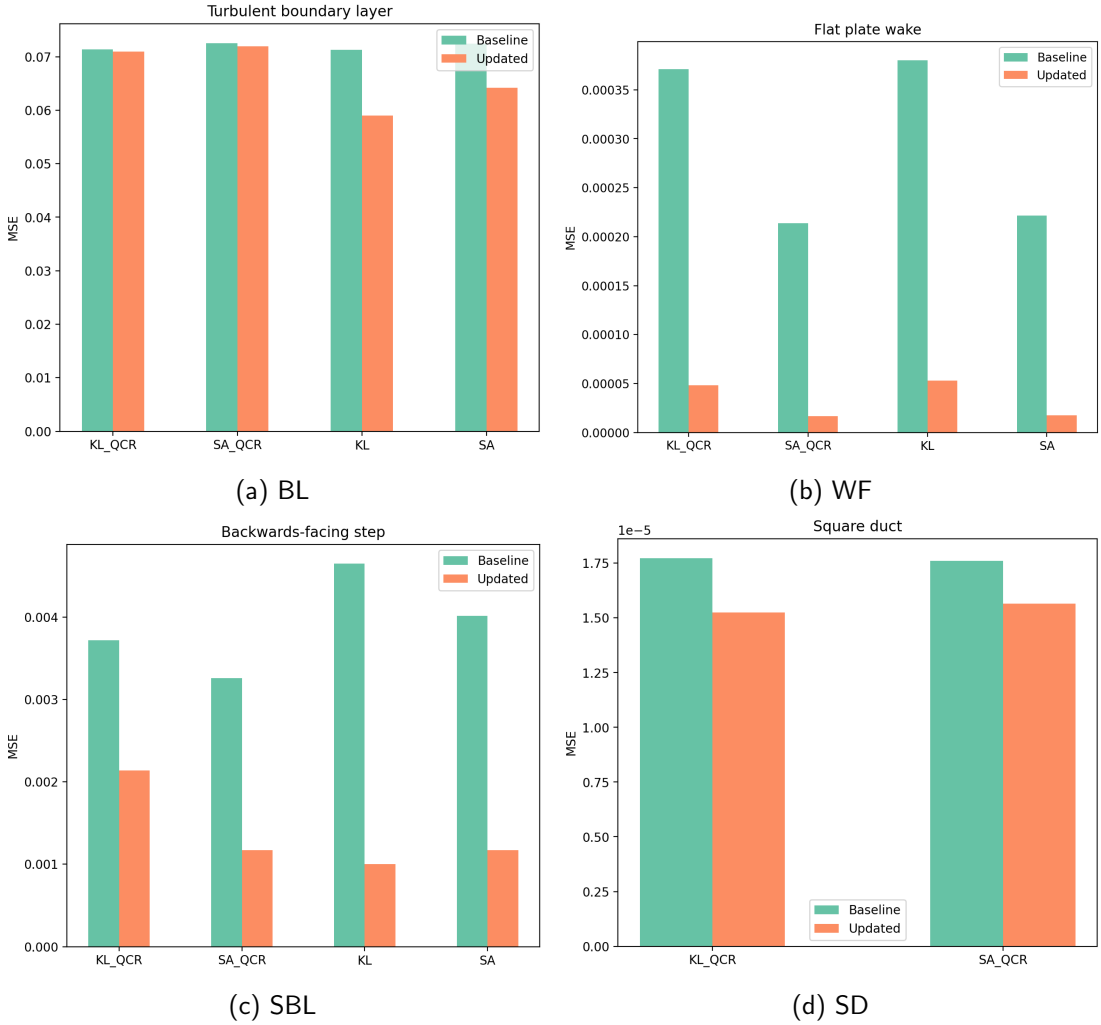
experimental data, as it introduces nonlinear Reynolds stress corrections needed to capture the corner vortices and the associated secondary motions. However, the high values for both the QCR and hyper-parameter suggest that even with a strong QCR correction, the linear eddy viscosity models do not capture accurately the corner flow. A more complex model should be used for better results, which would be interesting to explore in future research.

7.2.3 Predictions using the updated models

Predictions of the building block flows

Based on the *Maxima A Posteriori* of the model parameters, a posteriori predictions of the observed quantity or other QoIs can be made by propagating the MAP values through the CFD solver.

The calibrated models are first used to generate an a posteriori prediction for the calibration flow scenario, to assess accuracy improvements due to calibration. Once the Maximum A Posteriori (MAP) values of the $N_M \times N_{BB}$ models are propagated through the CFD solver, the mean squared error (MSE) between the predicted and observed data fields can be computed. The MSE of the baseline and updated models are gathered in Figures 7.9a to 7.9d. As expected from the discussion on the posterior parameters, the model performances on the BL building block, displayed in Figure 7.9a, do not show significant improvement with calibration, the calibrated



(a) BL (b) WF

(c) SBL (d) SD

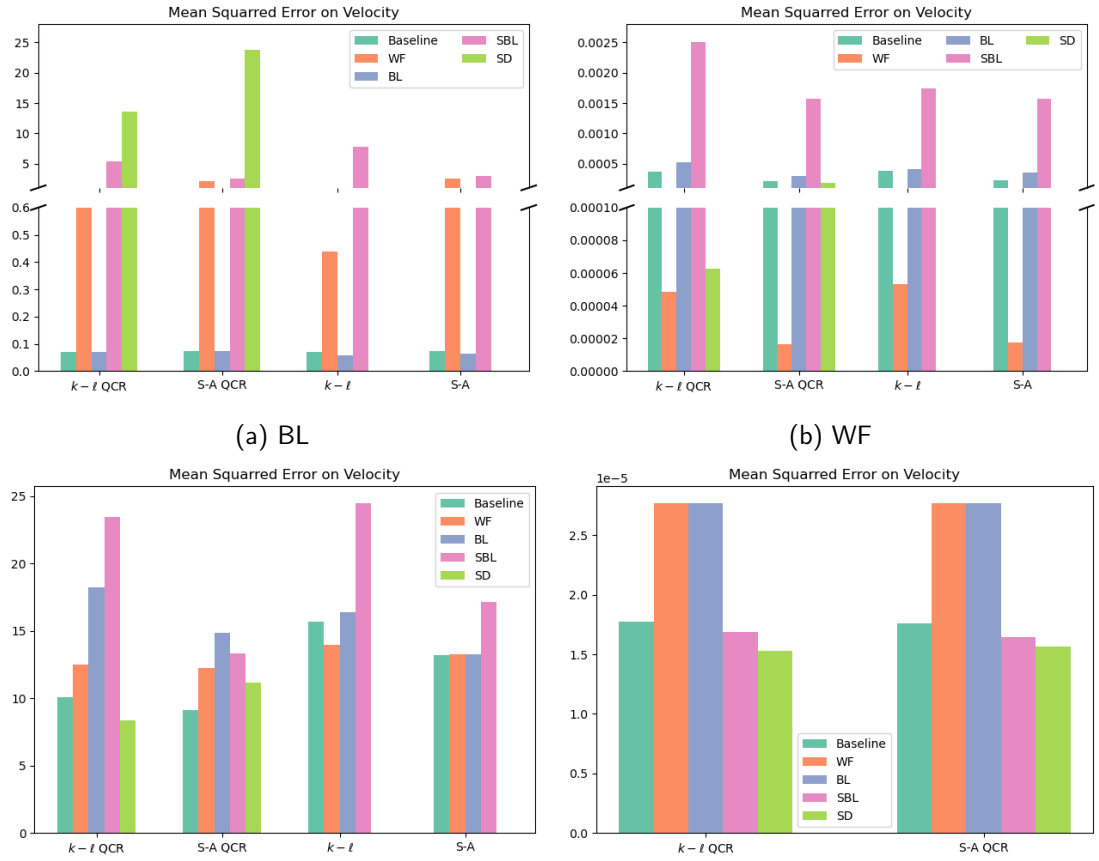
Figure 7.9 – MSEs of baseline and updated models on the different building blocks.

parameters being not markedly different from the baseline parameters. This is reassuring since the models are calibrated on BL configurations. Notably, all baseline models exhibit roughly the same performance.

In contrast, the performances of the models on the WF and the SBL significantly improve, suggesting that the updated models become specialised in predicting the WF configuration. This specialisation could potentially lead to over-specialisation on a single configuration. However, this is not problematic and might even be beneficial, as the proposed framework would aggregate

the model so that the expert models are applied preferentially in their region of expertise. For example, an over-specialised model on the WF would only be applied to regions with wake-like dynamics, making its poor performance in a backwards-facing-step-like dynamic inconsequential. Finally, the performances on the SD building block are only slightly improved, because the models cannot adjust the parameters enough to match the experimental data within the prescribed prior ranges.

Cross-prediction of the building block flows



(a) BL

(b) WF

(c) SBL

(d) SD

Figure 7.10 – MSEs of baseline and updated models on the different building blocks crossed predictions.

As a second test, we apply expert models calibrated for one building block, to the other

building blocks. This allows us to investigate the universality of the calibrated models. The solution accuracy is measured as the MSE to the target data, reported in figures 7.10a to 7.10d. For each building block, the corresponding expert models have smaller errors, confirming that calibration improves the model's predictive capability for the calibration scenario. The error reduction can be dramatic, as observed, e.g. for the WF case. The drawback is that the expert models for one building block can perform very badly in another case. For example, all expert models except the ones calibrated for the BL case show errors 10 to 100 times higher than the baseline models. Of note, the QCR corrections are not influential for this building block. In addition, the performance of the calibrated models can vary greatly across the building blocks, and in a sometimes (apparently) unpredictable manner.

7.2.4 Section summary

This section has examined the calibration of four RANS models, the Spalart-Allmaras and $k - \ell$ models with their QCR corrections, using experimental data from four building blocks to better understand their performance across various physical dynamics.

The key findings are:

1. The calibration reveals that model parameters are highly sensitive to the calibration data, and no universal set of coefficients is effective for all flow types.
2. Calibrating RANS models for specific building blocks enhances their predictive accuracy for those particular flows, demonstrating improved performance on calibration scenarios. Models tend to perform poorly when applied to building blocks different from those used for calibration.
3. The values of the σ_η hyper-parameters, which represent model-form uncertainty, vary significantly across different building blocks. For flows like the backwards-facing step (SBL) and square duct (SD), high values of σ_η indicate considerable model-form uncertainty, suggesting that parameter adjustments alone are insufficient for accurate predictions and highlighting the need for further refinement of the models.

In conclusion, while no single RANS model can universally predict all flow types with high accuracy, integrating multiple expert models into a hyper-model can offer a promising solution. The next section will explore this approach through the BBMA aggregation procedure, aiming to harness the advantages of specialised models while mitigating their limitations.

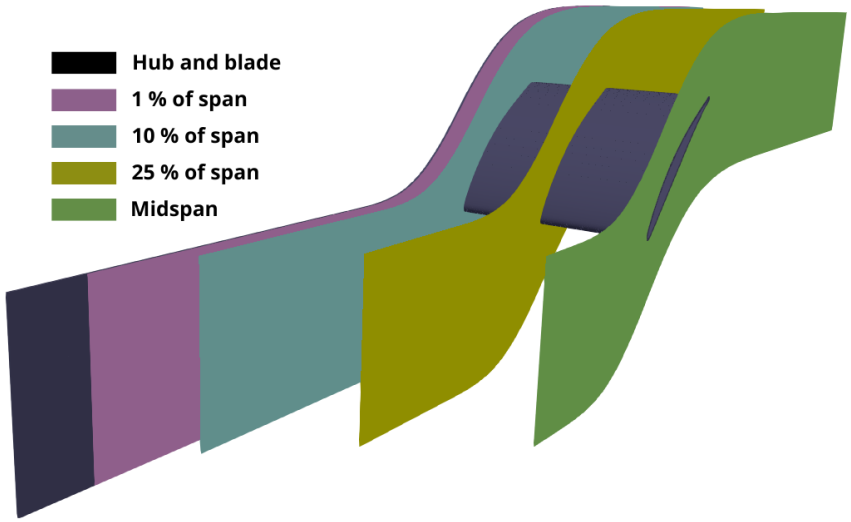


Figure 7.11 – Indicative plot of the slices used for plotting results

7.3 Aggregation of Building-Block Expert Models: Application to the 3D NACA65 compressor cascade

In this Section, we train a space-dependent aggregated model (following the XUMA idea) and we show that models calibrated on a few, well-chosen building block flows can be combined to capture a complex 3D flow configuration based on a set of local flow features that distinguish which of the component models are the best suited to capture some local physical process.

The demonstrator case is the 3D NACA65 V103 linear cascade configuration presented in Section 4.2.2. The Maximum A Posteriori (MAP) values of the parameters are used to predict the 3D configuration for two flow scenarios, corresponding to inlet angles of 4 and 7 degrees (noted 4DEG and 7DEG scenarios). Space-dependent weighting functions are trained for the 14 expert models by using selected data extracted from the 4DEG. Such functions are then used to reconstruct all other flow quantities for the 4DEG scenario and the unseen 7DEG scenario.

7.3.1 Application of expert models to the 4DEG scenario

Before training the aggregated model, we investigate the accuracy of the individual building block model solutions for the configuration at stake. For that purpose, we compute the 4DEG configuration with each of the 14 update models, and we plot the MSE to reference velocity

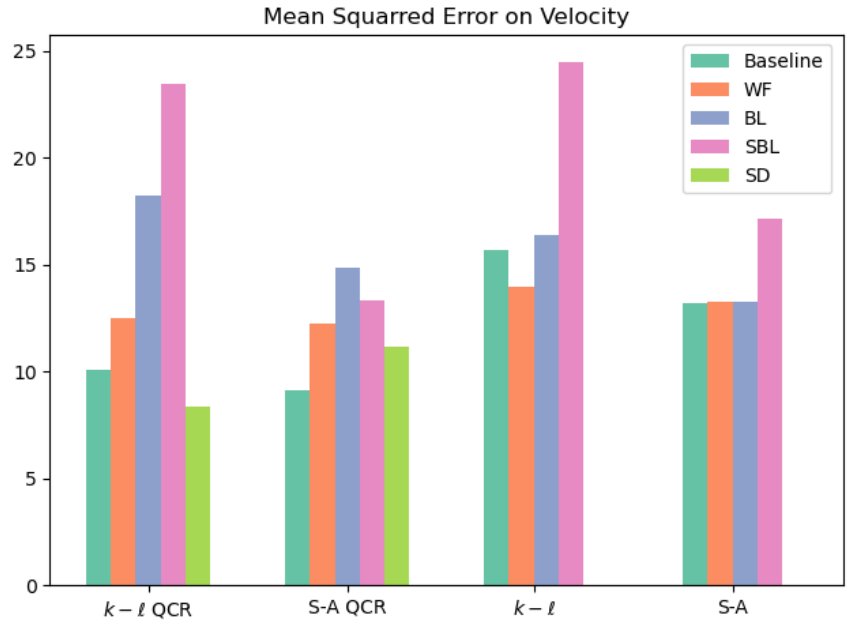


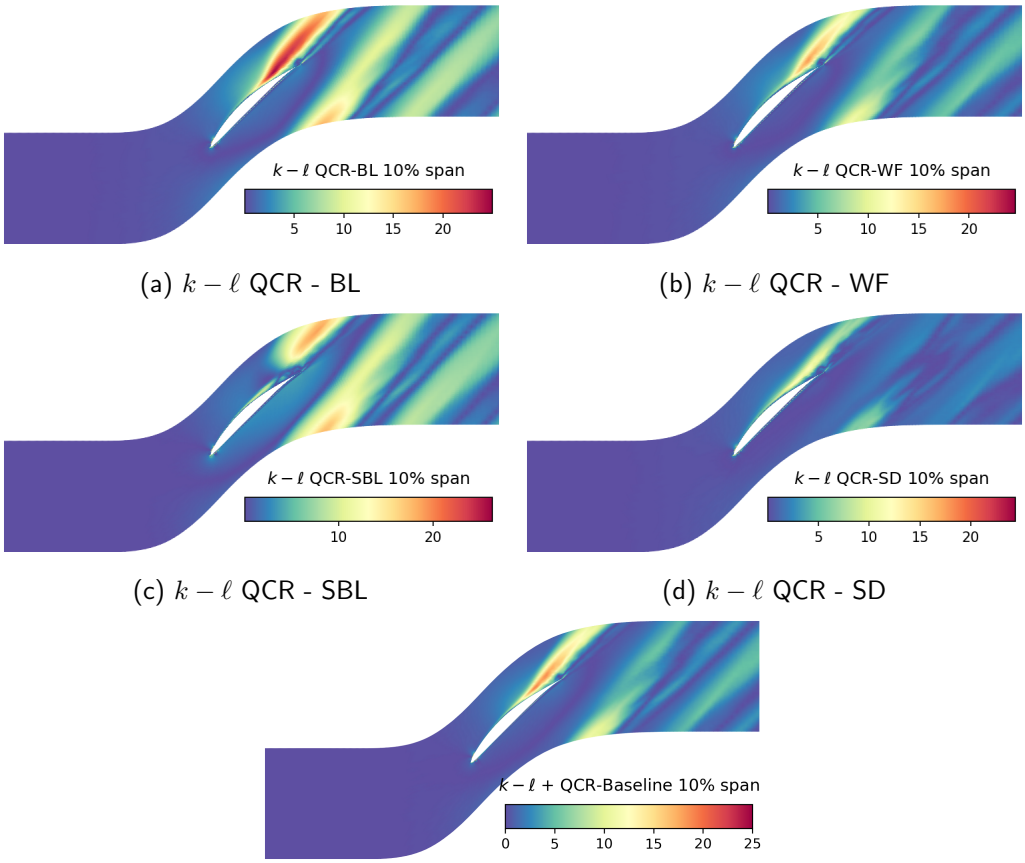
Figure 7.12 – MSEs of baseline and updated models on the different building blocks after prediction on the 3D NACA 65 on the 4DEG scenario.

data in Figure 7.12, where the baseline model MSEs are also reported for reference. It can be observed that most models have an increased MSE compared to the baseline models, i.e. are performing worst that the baseline models. This is due to the over-specialisation of the models on each building block, causing the models to perform better only in flow regions that resemble closely to the calibration scenarios.

To get further insight into the error sources, we plot distributions of the local error along slices extracted at various spans, as shown in Figure 7.11.

Figures 7.13ato 7.14e show the absolute error between the observed data for the updated and baseline models at 10% and midspan of the span for the $k-\ell$ QCR model, calibrated against various building blocks. The performance of the updated models varies significantly depending on the calibration building block. For instance, the $k-\ell$ models calibrated on the SD or the SBL blocks outperform the baseline model near the blade and at the hub, because they better capture the corner vortices. The model calibrated on the wake outperforms the baseline one in the wake, while other models deteriorate performance in this region.

Being impossible to determine a priori the application region of each model, it seems



(a) $k-\ell$ QCR - BL

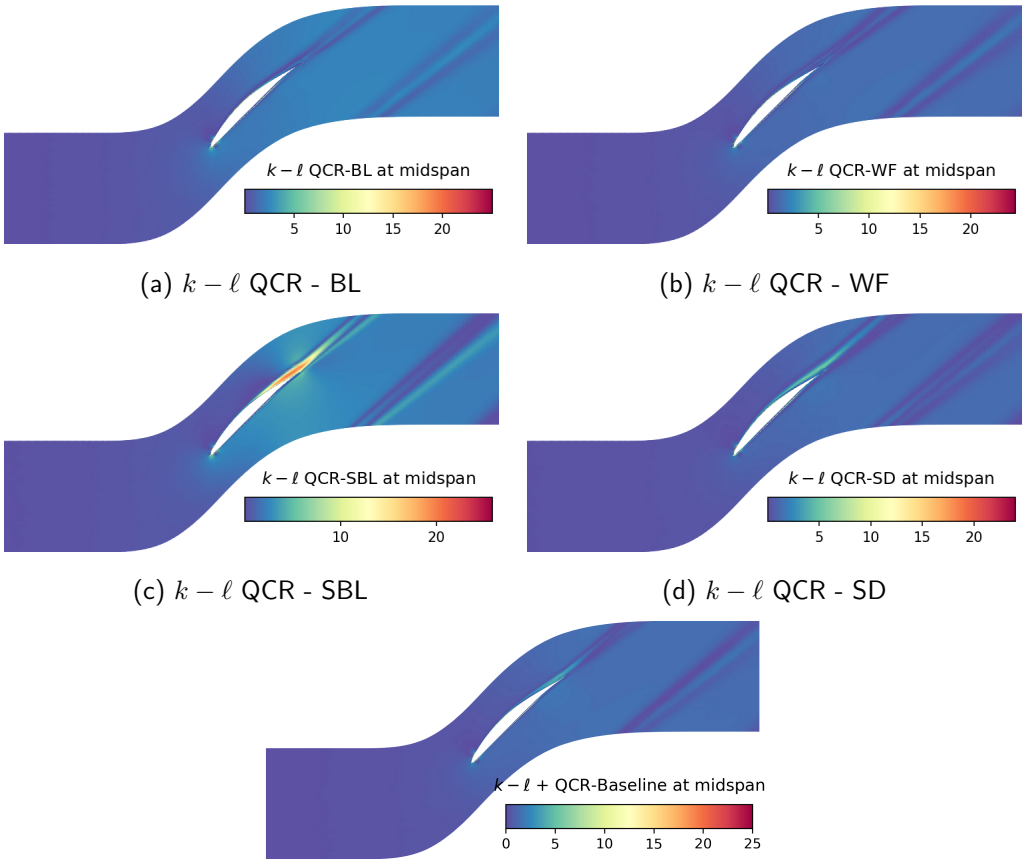
(b) $k-\ell$ QCR - WF

(c) $k-\ell$ QCR - SBL

(d) $k-\ell$ QCR - SD

(e) $k-\ell$ QCR - Baseline

Figure 7.13 – Velocity absolute error plots for the $k-\ell$ model using the baseline or building-block updated parameters. 4DEG scenario, 10% of the span.



(a) $k-\ell$ QCR - BL

(b) $k-\ell$ QCR - WF

(c) $k-\ell$ QCR - SBL

(d) $k-\ell$ QCR - SD

(e) $k-\ell$ QCR - Baseline

Figure 7.14 – Velocity absolute error plots for the $k-\ell$ model using the baseline or building-block updated parameters. 4DEG scenario, 10% of the span.

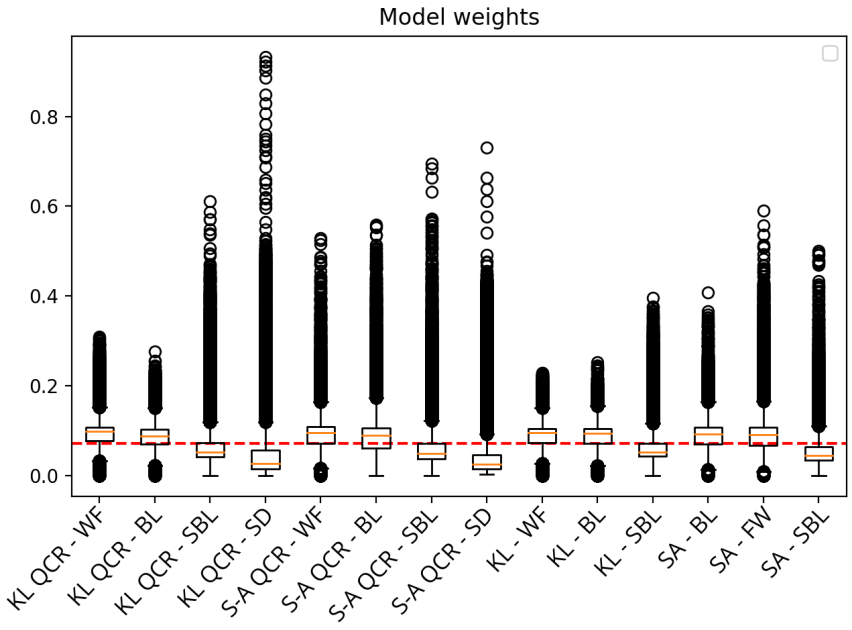


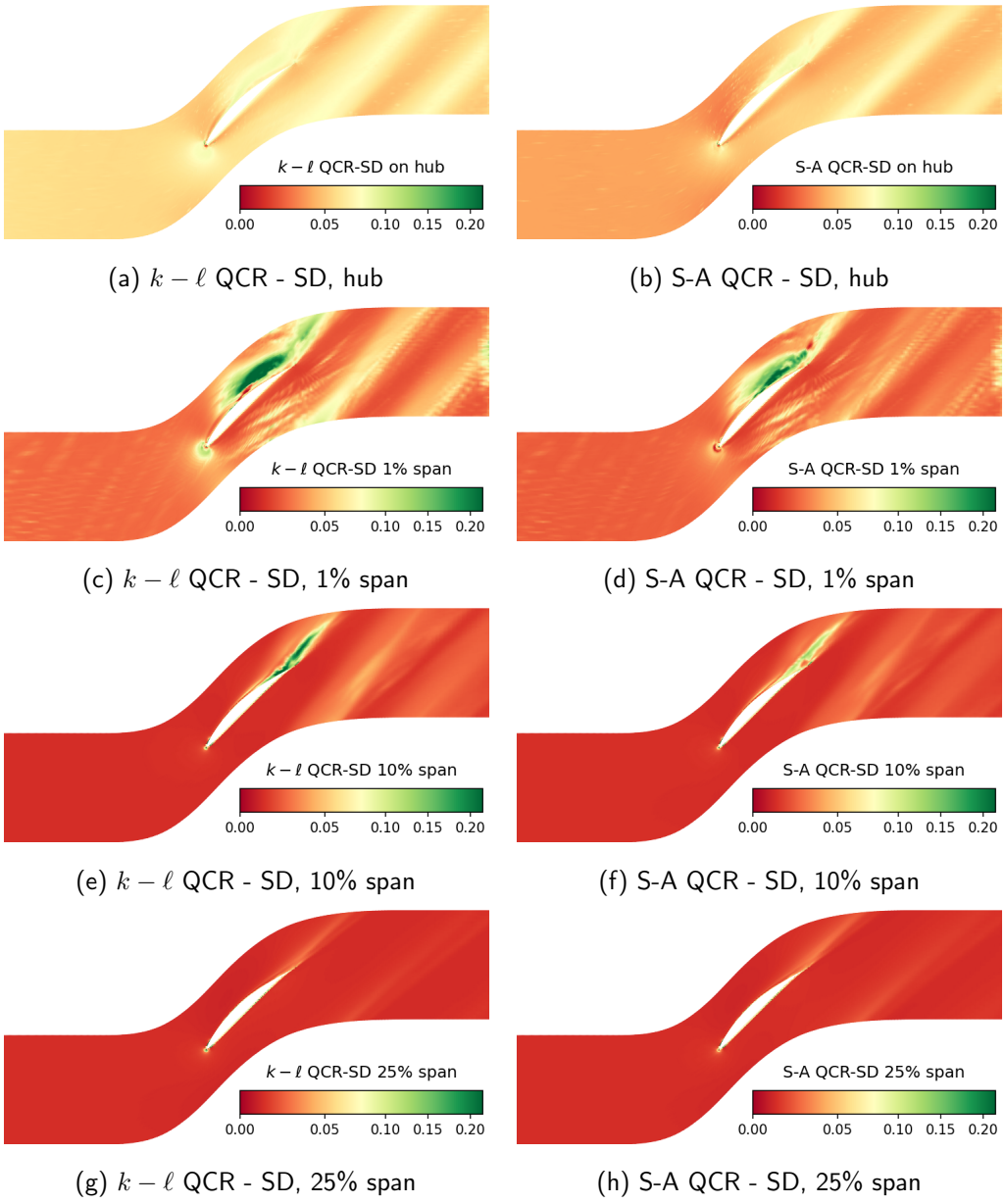
Figure 7.15 – Box plot of weights computed on the 4DEG incidence scenario. The red dashed line indicates the prior plausibility and equiprobability.

meaningful to use the XUMA procedure to aggregate the model using data-driven weighting functions.

7.3.2 Space-Dependent Weights

The solutions of the 14 expert models for the 4DEG 3D cascade scenario are used to compute their posterior local plausibilities, given by equation 7.1. Given the difficulty of determining *a priori* which model performs better, we assume that the models are *a priori* equally likely, and we set the prior plausibilities as $p(M_{m,b}) = \frac{1}{14}$. The posterior plausibility (model weights) being available only at points where observations are available, we then proceed to train a Random Forest Regressor to interpolate the weights in a space of features, following the procedure described in Chap. 5.

Figure 7.15 displays the box plots of the model weights computed on the 4DEG scenario. The box plot is the graphical representation of the weight distribution and displays the median, quartiles, and potential outliers. Each box spans the interquartile range (IQR) and whiskers extend to the minimum and maximum data points within 1.5 times the IQR, the outliers are



(a) $k-\ell$ QCR - SD, hub

(b) S-A QCR - SD, hub

(c) $k-\ell$ QCR - SD, 1% span

(d) S-A QCR - SD, 1% span

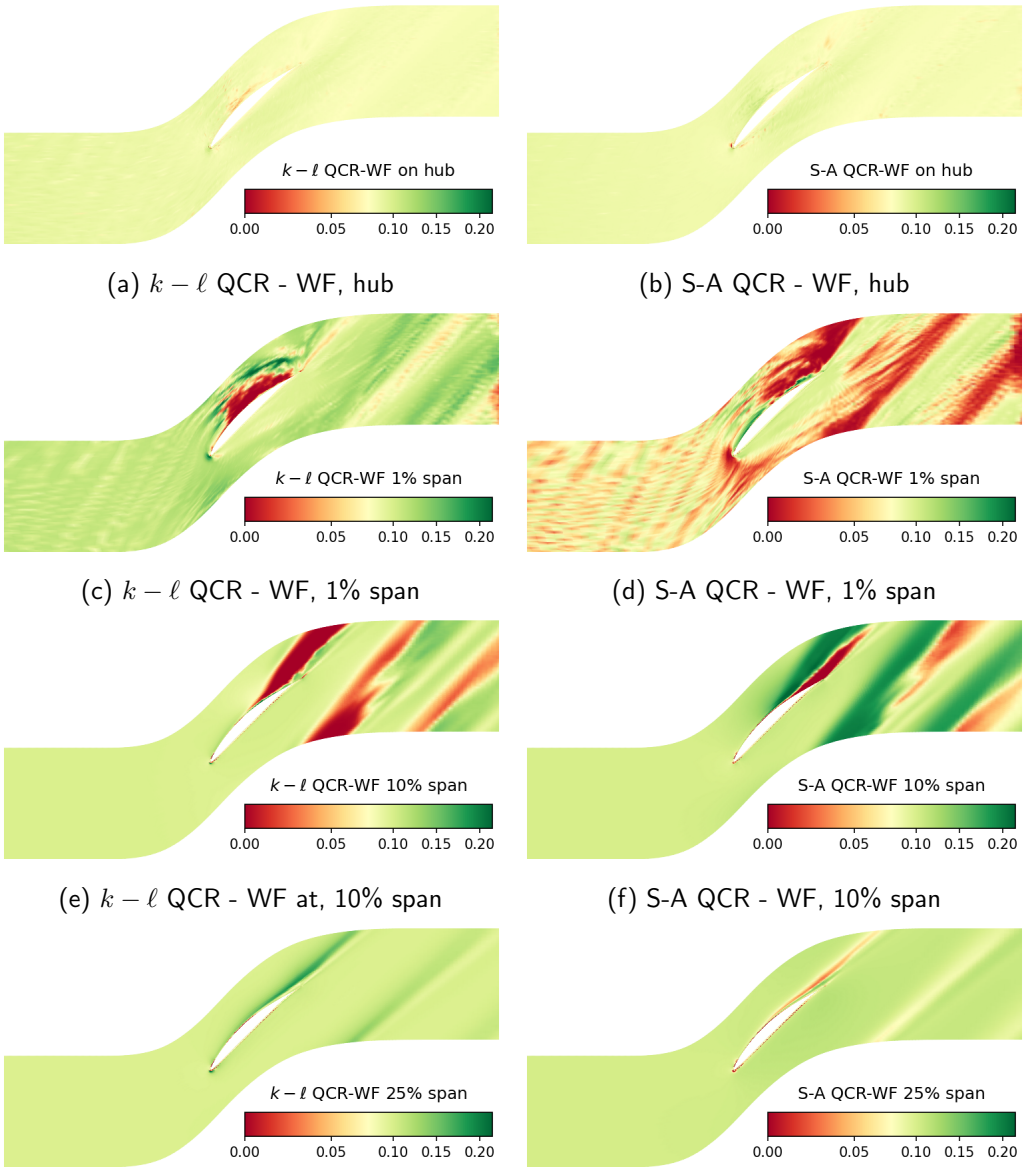
(e) $k-\ell$ QCR - SD, 10% span

(f) S-A QCR - SD, 10% span

(g) $k-\ell$ QCR - SD, 25% span

(h) S-A QCR - SD, 25% span

Figure 7.16 – Isocontours of the observed weights (local posterior plausibilities) for the $k-\ell$ and S-A models along selected spanwise planes.



(a) $k-\ell$ QCR - WF, hub

(b) S-A QCR - WF, hub

(c) $k-\ell$ QCR - WF, 1% span

(d) S-A QCR - WF, 1% span

(e) $k-\ell$ QCR - WF at, 10% span

(f) S-A QCR - WF, 10% span

(g) $k-\ell$ QCR - WF, 25% span

(h) S-A QCR - WF, 25% span

Figure 7.17 – Weight for QCR corrected models trained on the Flat Plate Wake building block at different slices of blade span.

then shown as individual points. It suggests that the model's local weights vary significantly across the flow. Some models are generally more plausible than others on average: this is the case for instance of the models trained on the BL and WF building blocks. Such results are consistent with the hyper-parameter MAP values obtained after calibration. Some models, such as those calibrated on the SBL or SD blocks, seem less probable on average but have also many probes where the weights are higher than all the other models, indicating that these models are favoured in specific regions. This is confirmed by inspection of Figures 7.16a to 7.16h, which display the weights of the QCR models calibrated on the SD block. Such models are disfavoured in most of the flow but exhibit high local values of weights within the corner flow between the hub and the blade. On the contrary, the QCR models trained on the BL block Figures 7.17a to 7.17h exhibit overall high weights, but low ones within the corner vortex region.

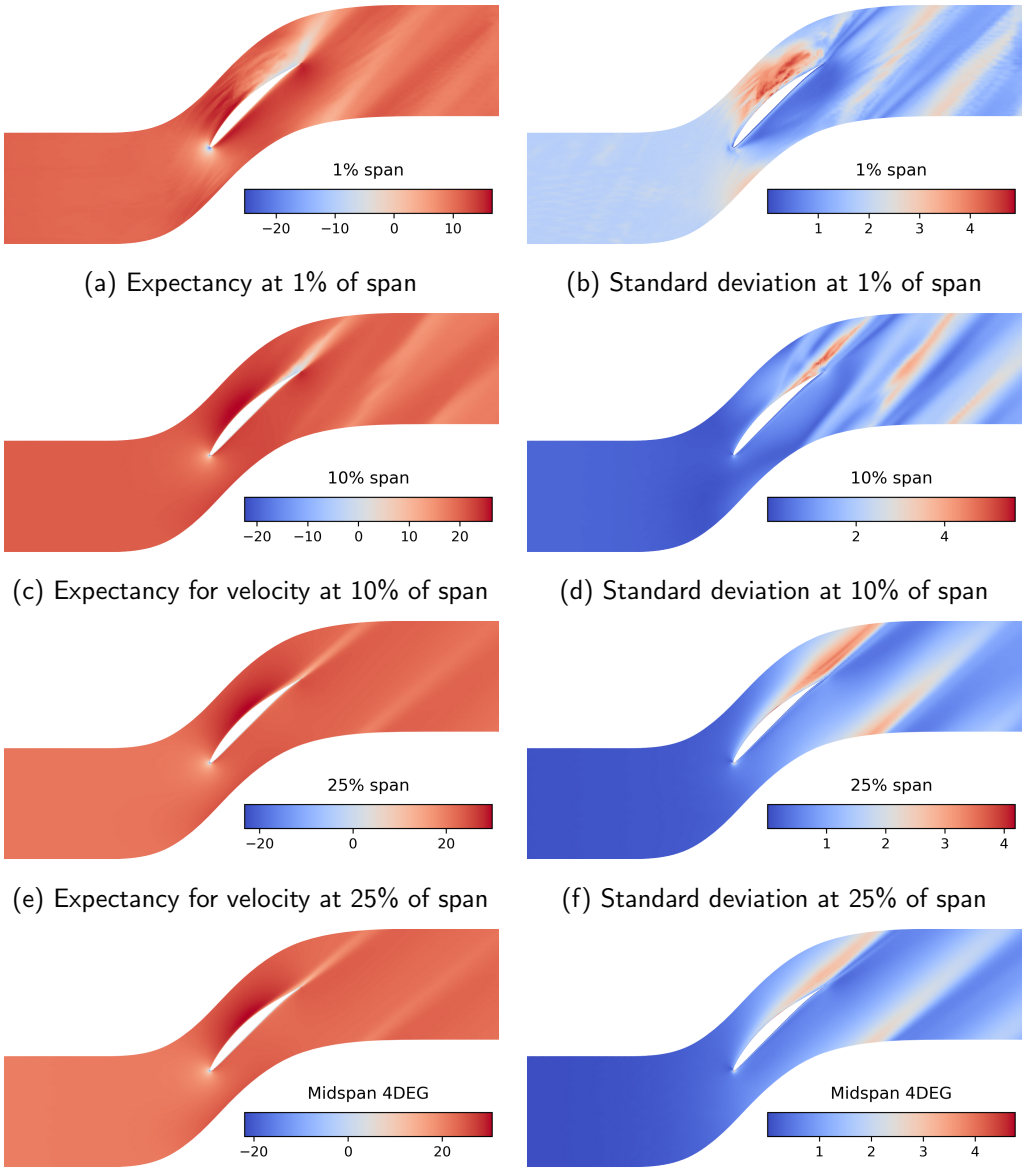
These results are very important as it was what the calibration step aimed to do: to generate specialised models that capture specific physical processes with high accuracy.

7.3.3 Model aggregation and prediction of a quantity of interest

The trained weighting functions can be injected in Eqs 7.4 to 7.5, to obtain the expectancy and variance of any quantity of interest, for both the training scenario and a completely unseen scenario. In the following, we first assess the accuracy of the aggregated *a posteriori* prediction for the 4DEG scenario used to train the weights (we recall that only velocity data at selected observation points are used for training), and afterwards for the 7DEG scenario.

On the training scenario (4DEG)

First, let us analyse the prediction of the BBMA method on the training scenario, i.e. the scenario on which the weights were computed using the available observed data and regressed onto the feature space. Figures 7.18a to 7.18h display the expectancy and the standard deviation of the longitudinal velocity, i.e., the weighting quantity. The standard deviation is null in the potential flow region (which is essentially insensitive to the turbulence models) but appears more significant in the wake and corner flow. In the BBMA framework, the standard deviation needs to be interpreted to indicate how well the different models agree with one another. A relatively high standard deviation indicates that the models are not in agreement, a point previously highlighted. Indeed, the different models are calibrated to be expert models on specific physical dynamics and do not perform well outside of these calibration dynamics.



(a) Expectancy at 1% of span

(b) Standard deviation at 1% of span

(c) Expectancy for velocity at 10% of span

(d) Standard deviation at 10% of span

(e) Expectancy for velocity at 25% of span

(f) Standard deviation at 25% of span

(g) Expectancy at midspan

(h) Standard deviation at midspan

Figure 7.18 – Expectancy and standard deviation of the posterior longitudinal velocity BBMA prediction for the training scenario 4DEG at different spanwise positions.

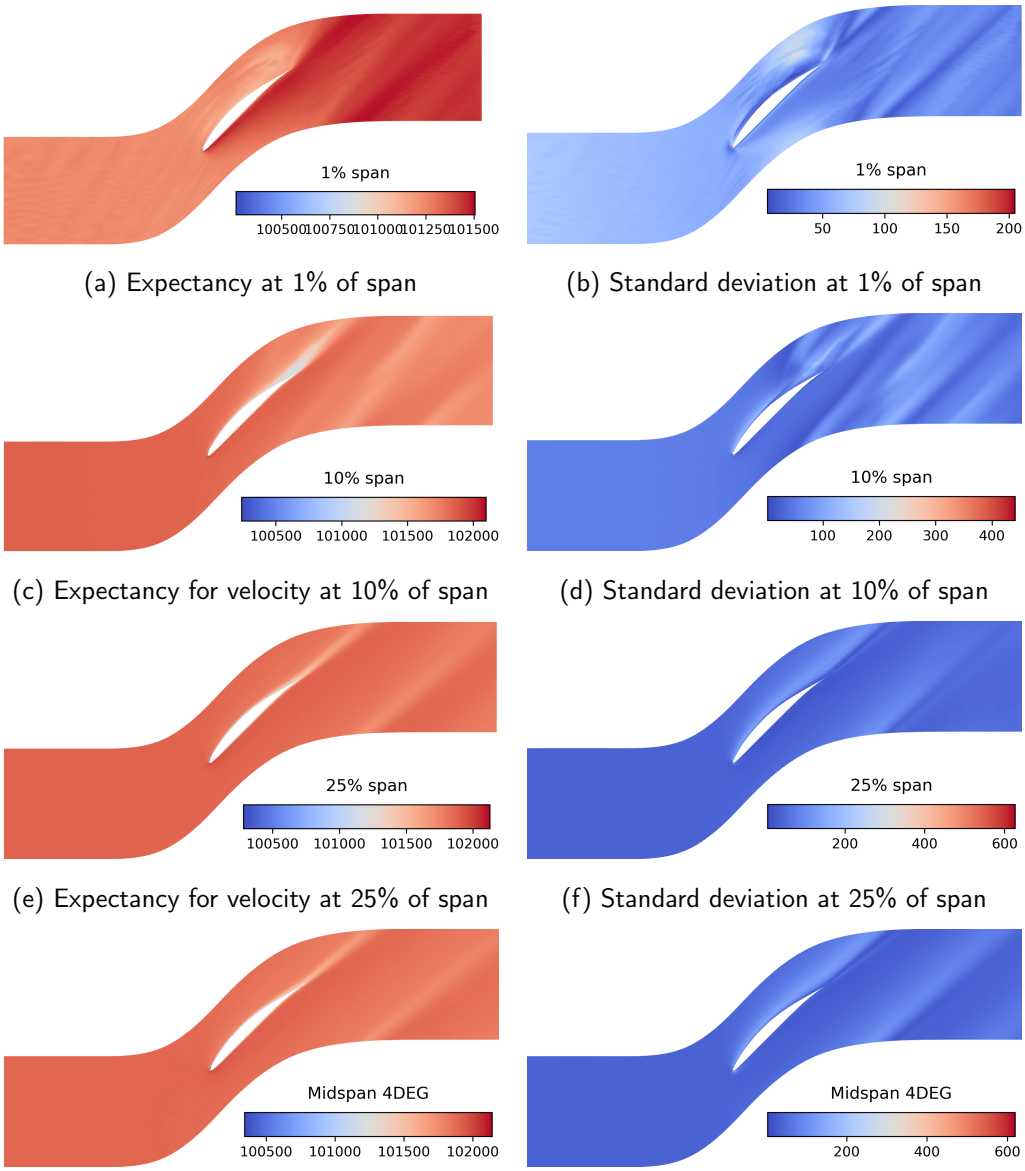
Figure 7.19 – Mean squared error on longitudinal velocity for baseline and updated models and the BBMA trained on scenario 4DEG

Figure 7.19 displays the mean squared error (MSE) on longitudinal velocity for the baseline and updated RANS models and the BBMA. We also report results obtained by simply performing an arithmetic mean of the component models with uniform constant weights at all prediction locations (denoted "mean" in the picture). The results suggest that i) the updated model performing better for the 4DEG flow is $k - \ell$ calibrated on the SD data (which informs well the QCR correction); ii) the BBMA reconstruction performs better than any of the updated or baseline models. While the updated models do not always perform better than the baseline models, the space-dependent aggregation, i.e., which promotes locally the best-performing models, allows for a significant reduction of the MSE throughout the domain. Additionally, while the algebraic mean of all models performs better than the worst models, it does not perform as well as the best-updated model or the BBMA prediction, demonstrating the interest in using space-dependent model weighting.

Next, we analyse the prediction of the BBMA method on different quantities than the weighting quantity. Figures 7.20a to 7.21h display the expectancy and standard deviation of the total pressure and pressure fields, i.e., unseen quantities. The standard deviation remains null in the freestream but becomes more significant in the wake and corner flow for the total pressure, similar to the longitudinal velocity, and remains small for the pressure field. The similarity in behaviour between the total pressure field and the longitudinal velocity field is expected, as these quantities are closely linked.

Figures 7.22a and 7.22b display the mean squared error (MSE) on longitudinal velocity for the baseline and updated RANS models, the BBMA, and the mean reconstruction after calibration and weighting on scenario 4DEG, using the velocity as the weighting quantity and predicting the total pressure or the static pressure fields. The MSEs suggest that the BBMA reconstruction performs better than any updated or baseline models for the total pressure field (which depends on the flow velocity seen during the training of BBMA weights) but not for the pressure field. Notably, the BBMA applied to the pressure field performs worse than the QCR-corrected baseline models, indicating that the weights computed for the velocity are not necessarily optimal for a different QoI. Although BBMA is not more accurate than the most accurate component model or than some of the baseline models at predicting the pressure field, it is still more accurate than most of the models, showing that model aggregation allows to mitigate risks associated with the choice of a single model. Moreover, BBMA provides a lower error than the arithmetic mean of component model predictions, demonstrating the interest of space-dependent weighting.

To further assess the sensitivity of BBMA results to the quantity used to train the weighting



(a) Expectancy at 1% of span

(b) Standard deviation at 1% of span

(c) Expectancy for velocity at 10% of span

(d) Standard deviation at 10% of span

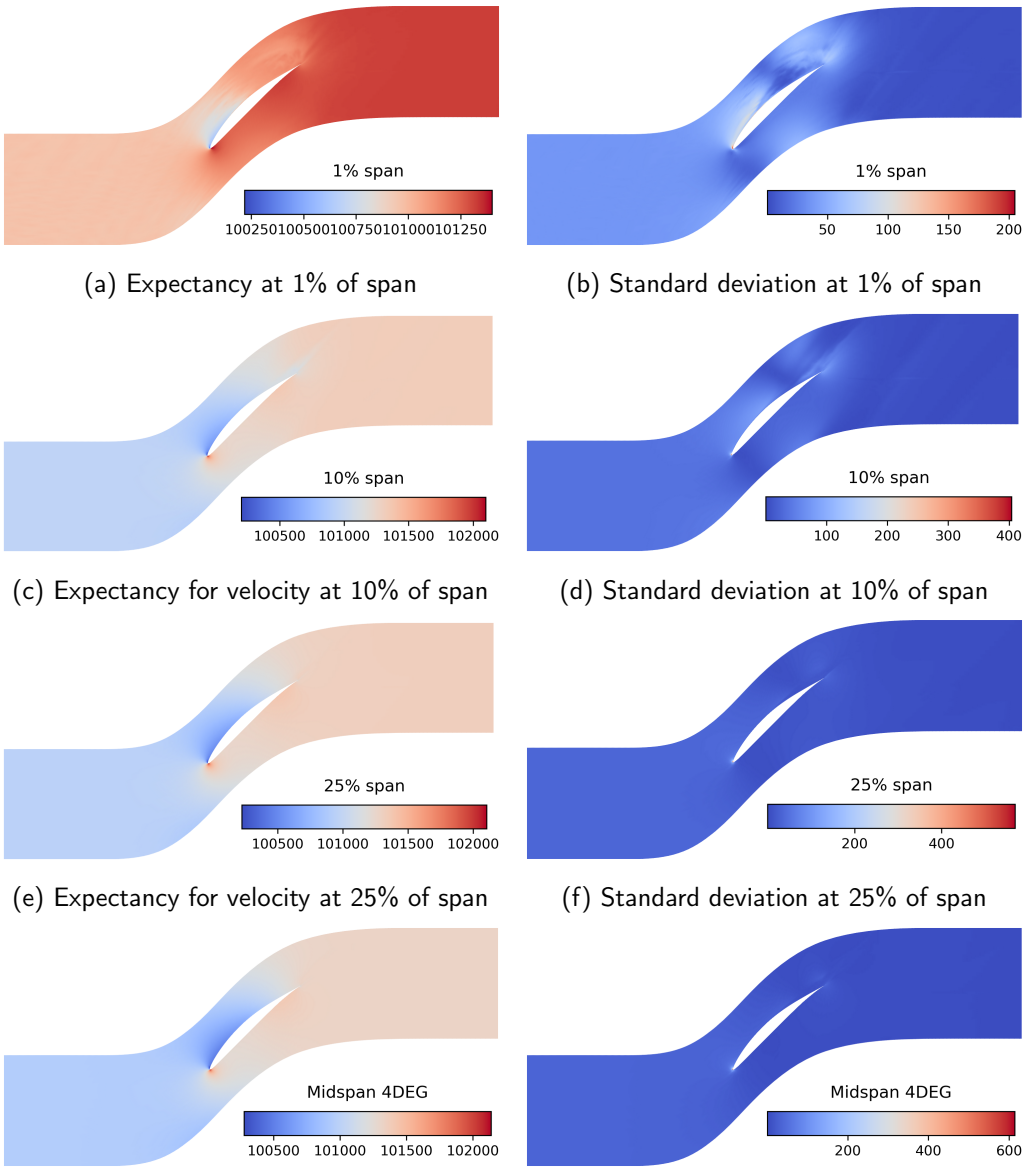
(e) Expectancy for velocity at 25% of span

(f) Standard deviation at 25% of span

(g) Expectancy at midspan

(h) Standard deviation at midspan

Figure 7.20 – Expectancy and standard deviation of total pressure BBMA posterior prediction for the training scenario 4DEG at different spanwise positions.



(a) Expectancy at 1% of span

(b) Standard deviation at 1% of span

(c) Expectancy for velocity at 10% of span

(d) Standard deviation at 10% of span

(e) Expectancy for velocity at 25% of span

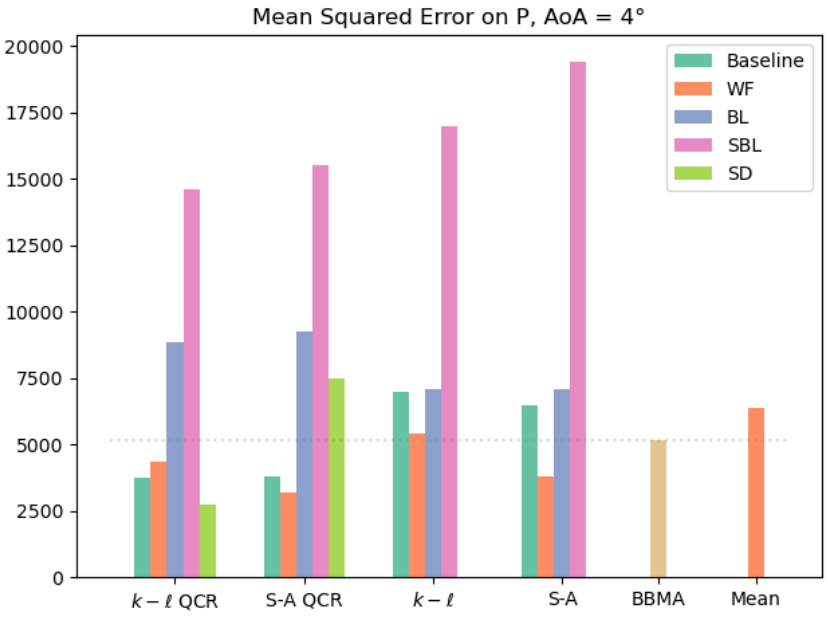
(f) Standard deviation at 25% of span

(g) Expectancy at midspan

(h) Standard deviation at midspan

Figure 7.21 – Expectancy and standard deviation of static pressure BBMA posterior prediction for the training scenario 4DEG at different spanwise positions.

(a) Mean squared error on total pressure



(b) Mean squared error on pressure

Figure 7.22 – Mean squared errors on unobserved QoIs for baseline and updated models and the BBMA trained on scenario 4DEG

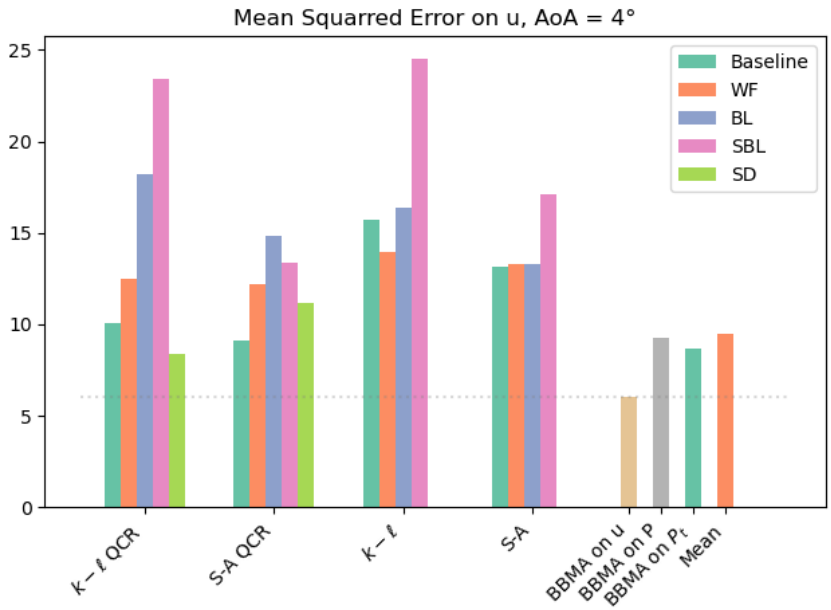
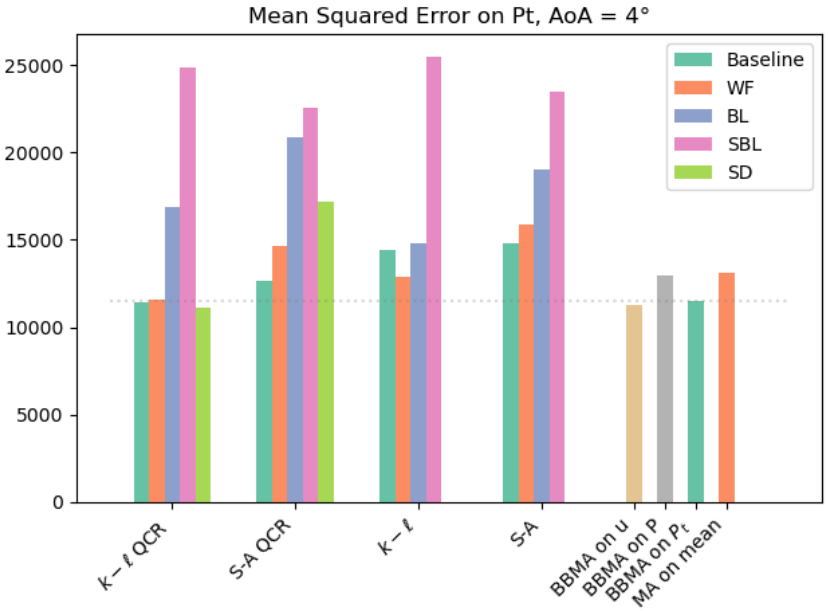
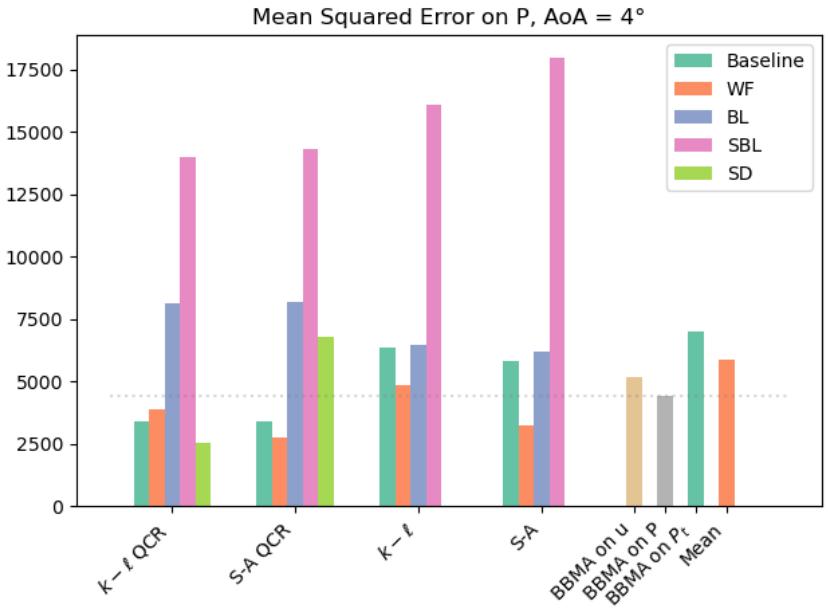


Figure 7.23 – Mean squared error on longitudinal velocity for baseline and updated models and the BBMA on various quantities and mean reconstructions after calibration and weighting on scenario 4DEG



(a) Mean squared error on total pressure



(b) Mean squared error on pressure

Figure 7.24 – Mean squared error baseline and updated models and the BBMA on various quantities and mean reconstructions after calibration and weighting on scenario 4DEG

functions we present in Figures 7.24b to 7.24b the MSEs of various models, as well as BBMA with weights trained on different observations, and namely the velocity, pressure, and total pressure, for scenario 4DEG. Predictably, BBMA results for a given QoI improve when the weighting functions are trained with data for the same QoI, although the BBMA results are never better than some baseline or component models when it comes to predicting the pressure.

Prediction of an unseen scenario

To predict an unseen scenario, the updated models are first used to produce alternative solutions. The local flow features are then extracted from the individual predictions, and used to interrogate the RFR providing the model weights. Finally, the weights are used to aggregate the models into a BBMA prediction. Figure 7.25 displays side by side the predicted weights using the RFR trained on scenario 4DEG and the true weights after training on scenario 7DEG for the S-A QCR - SD updated model in different spanwise positions. The figures suggest that the predicted and true weights can differ significantly, especially in the corner flow region and near the trailing edge. The discrepancy appears because the 7DEG scenario is characterised by boundary layer separation near the trailing edge in scenario 7DEG, which is absent in the training scenario 4DEG. The RFR has never "seen" this separation during training and therefore it is not able to assign the models the required weights. We expect that errors in predicting the weights for an unseen flow scenario might affect the accuracy of the prediction.

Figure 7.28b displays the mean squared error (MSE) on longitudinal velocity for the baseline and updated RANS models, the BBMA, applied to the unseen scenario 7DEG. The BBMA prediction in this case does not outperform all the baseline or updated models but is still more accurate than most of them. Interestingly, if the models are aggregated using the exact weights based on data observed for scenario 7DEG, the BBMA prediction becomes much more accurate than all the other. This confirms that BBMA errors are due to the misprediction of the weighting functions in an unseen scenario that differs significantly from the one used to train the weights and could be improved by training the weights on a more diverse set of data. Inspection of the error distributions along spanwise slices, shown in Figures 7.27a to 7.27f, shows indeed that most of the errors arise in the prediction of the separated flow region. Furthermore, if the BBMA is trained on scenario 7DEG, the solution accuracy improves dramatically.

Similarly to what was observed for the 4DEG scenario, the accuracy of BBMA decreases when predicting QoIs other than those used for training the weights, and specifically the static

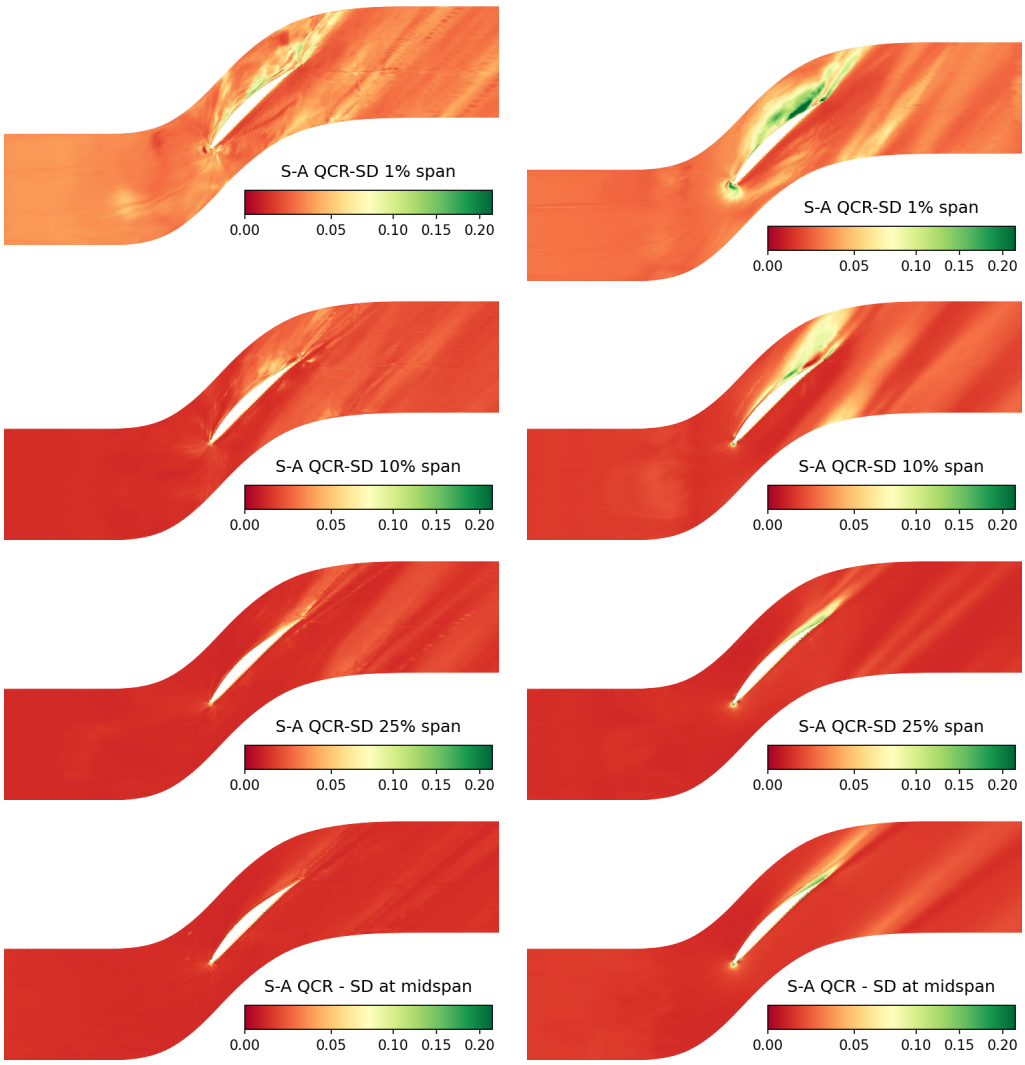


Figure 7.25 – Weighting functions assigned to S-A QCR model (calibrated on SD) at different spanwise positions for the prediction scenario 7DEG. Left: predicted weights (output of the RFR); right: exact weights (based on data observed for 7DEG).

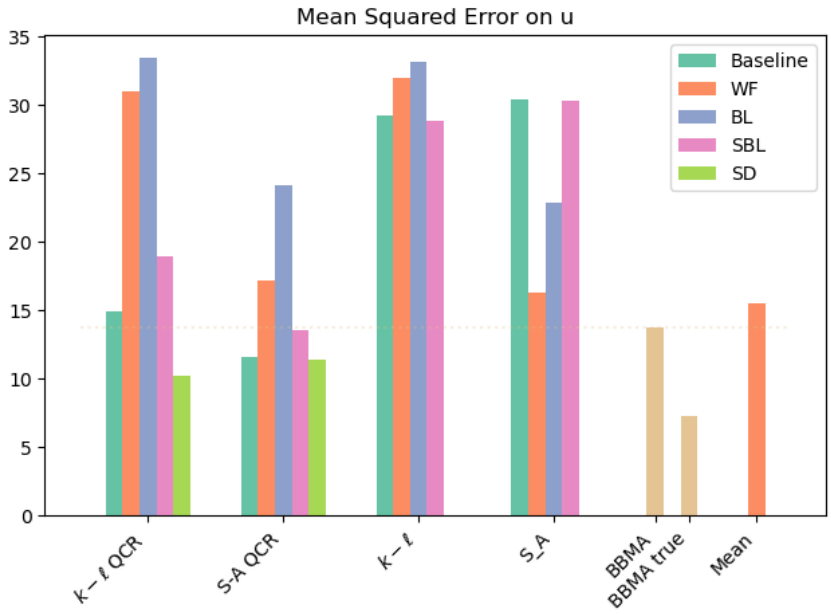
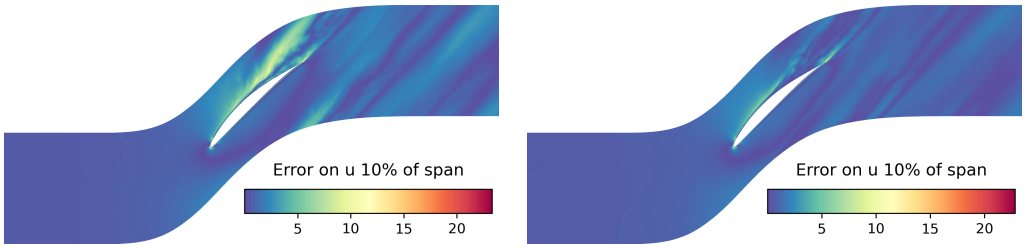
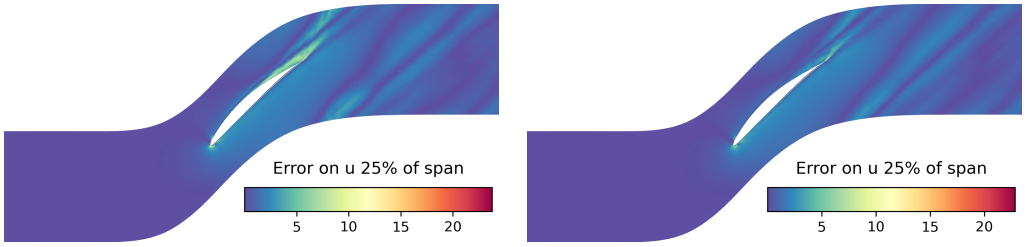


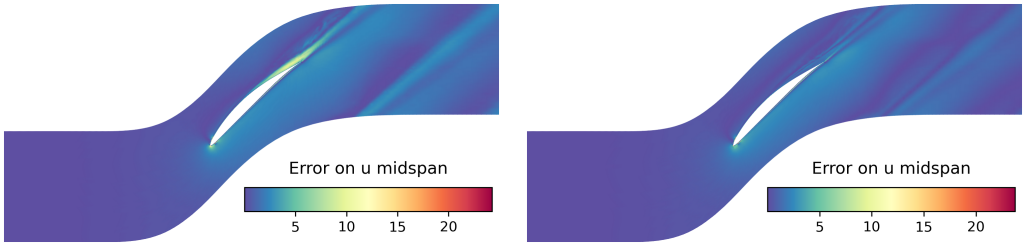
Figure 7.26 – Mean squared errors on the longitudinal velocity for baseline and updated models and the BBMA, applied to the unseen scenario 7DEG



(a) Error after training on 4DEG at 10% of span (b) Error after training on 7DEG at 10% of span



(c) Error after training on 4DEG at 25% of span (d) Error after training on 7DEG at 25% of span



(e) Error after training on 4DEG at midspan (f) Error after training on 7DEG at midspan

Figure 7.27 – Absolute error of the longitudinal velocity reconstructions using predicted weights after training on 4DEG or 7DEG and prediction on scenario 7DEG at different spanwise positions.

(a) Mean squared error on total pressure

(b) Mean squared error on pressure

Figure 7.28 – Mean squared error for baseline and updated models and the BBMA and mean reconstructions after calibration and weighting on scenario 4DEG and prediction on scenario 7DEG.

pressure, although it still performs much better than the baseline linear LEVM, as shown by expectation of the Mean squared errors reported in Figures 7.22a and 7.22b. Overall, the results show the great sensitivity of the solution to the model choice (LEVM or QCR) and the calibration of model parameters.

7.3.4 Section summary

This chapter has explored the aggregation of various expert models, each calibrated on fundamental "building block" flows, to predict the complex 3D configuration of the NACA65 V103 compressor cascade across two distinct scenarios. By leveraging the BBMA approach, which integrates multiple expert models, significant improvements in predictive accuracy were achieved compared to most individual models.

— **Enhanced Predictive Accuracy:** The BBMA approach markedly improves prediction accuracy for scenarios used in training and for the quantities of interest (QoI) used to train the weights, effectively mitigating the limitations of individual models.

— **Predictive Variance Estimation:** BBMA provides a measure of predictive variance, capturing uncertainties from model choices and calibration parameters, which helps in understanding the reliability of the predictions.

— **Performance in Unseen Scenarios:** The approach shows reduced accuracy when predicting scenarios significantly different from those used in training. The difference in predicted and actual model weights impacts performance, suggesting that incorporating a broader range of data could enhance predictions.

— **Impact of Training Quantity:** BBMA's performance varies with the type of training quantity. Predictions based on velocity fields yield better results compared to those based on pressure or total pressure fields, likely due to the calibration focus on velocity.

In conclusion, while BBMA demonstrates superior performance compared to individual models, its effectiveness is influenced by the diversity of training data and the type of quantities used for model weighting. Future work could focus on expanding the training dataset and exploring the calibration of models on different quantities to further enhance predictive accuracy.

7.4 Chapter summary

This chapter introduced the Building-Block Model Aggregation (BBMA) approach for improving the accuracy of RANS model predictions. Unlike previous methods like CMA, which rely on clustering for calibration, BBMA uses pre-defined "building-block" flows to create specialised models for different physical processes, thus eliminating the need for clustering.

The methodology involves three main steps: calibrating models using building-block flows, estimating space-dependent weights with a Random Forest Regressor (RFR), and aggregating the model predictions. The calibration process revealed that model parameters are highly sensitive to the specific flow types used. This indicates that a single set of coefficients is not effective across all flow types. Models performed better when calibrated for their specific building blocks but struggled with other flow types, highlighting limited generalisation. The variability in model-form uncertainty, as indicated by the hyper-parameter σ_η, suggests that current RANS models may not be complex enough to handle diverse turbulence conditions. High values of σ_η in certain building blocks indicate that these models might not be suitable for all turbulence scenarios, pointing to the need for more advanced RANS models.

After analysing the model weights, it was found that models trained on simpler and more common building blocks generally demonstrated greater reliability. In contrast, models calibrated on more specialised building blocks, like the Separated Boundary Layer (SBL) or Square Duct (SD), showed higher weights in specific regions where their specialisation was most effective. This finding aligns to create models that excel at capturing specific physical processes. The BBMA approach significantly improved predictive accuracy for the 3D NACA65 V103 compressor cascade by reducing both bias and variance in the training scenario. However, performance declined for unseen scenarios, highlighting, once again, the need for a more robust RFR interpolator. Predictions based on weights derived directly from observed data performed very well, underscoring the importance of accurate data in the prediction process. Additionally, the quality of predictions varied with the type of training quantity. Predictions based on velocity fields were more accurate than those based on pressure fields, suggesting that focusing on specific flow measurements can enhance the calibration and aggregation process.

In summary, the BBMA approach significantly improves prediction accuracy and provides valuable estimates of predictive variance. Future efforts should aim to strengthen the weight

interpolation step and expand the training dataset. Exploring different quantities for calibration will also be essential for further enhancing the performance of aggregated models. These improvements are crucial for broadening the methodology's applicability to a wider range of flow configurations and ensuring reliable predictions across various scenarios.

8

Conclusions

This thesis developed a comprehensive framework for addressing and reducing parametric and model-form uncertainties in Computational Fluid Dynamics (CFD), particularly for turbomachinery flows. Using a Bayesian approach, it tackled the uncertainties linked to turbulence closures in Reynolds-Averaged Navier–Stokes (RANS) equations, which are essential in aerodynamic design for turbomachinery. The focus was on reducing model-form uncertainties, which arise when selecting the most appropriate turbulence models for specific flow conditions.

Building on the work of De Zordo-Banliat et al. [21] on space-dependent model aggregation (XMA), this research introduced new methods for improving the efficiency of calibrating model parameters for costly numerical flow models. Furthermore, it investigated the feasibility of building more generalisable model aggregates by combining component models specialised at capturing specific flow dynamics. This approach was applied for the first time to a 3D turbomachinery setting, specifically a linear compressor blade cascade, representing a significant step forward in advancing turbulence models for complex engineering flows.

To enhance the predictive capabilities of the XMA method, which integrates various RANS models, we introduced the XUMA (space-dependent Updated Model Aggregation) method. XUMA builds on Bayesian Model Averaging (BMA) by calibrating model parameters within a Bayesian framework and incorporating space-dependent weights that adjust based on local model performance, similar to XMA (which relied on on-the-shelf, uncalibrated models). Unlike traditional BMA, which uses constant weights, XUMA adapts model contributions locally, leading to better predictions in areas where specific turbulence models excel, such as regions of high flow separation or near-wall phenomena. XUMA demonstrated improved predictive accuracy over baseline models by optimising individual models' strengths. However, challenges like over-specialisation and predicting weights for new scenarios remain, limiting its wider applicability in unseen cases. One of the key working hypotheses in the XUMA methodology assumed that the posterior distribution of parameters would be identical whether calibrated on individual probes or over the entire field. This assumption was later proven inaccurate in the chapter on in-cluster calibration, where a different set of parameters emerged after calibration. More refined methodologies were proposed to address these limitations, but a key advantage of this work is that the calibration remained non-intrusive, conducted outside the CFD solver. While more advanced techniques, such as calibrating space-dependent fields or employing machine learning approaches like Field Inversion or Sparse Bayesian Learning, could improve model accuracy, they would require more complex development within the CFD solver than adjusting model parameters alone.

Recognising the issue of over-specialisation, the second methodology employed clustering techniques to calibrate models based on similar physical dynamics. The Clustered Model Aggregation (CMA) method enhanced the process by using unsupervised machine learning to cluster training data before calibration. This approach involves offline clustering with Gaussian Mixture Models (GMM), in-cluster model calibration, and aggregation of predictions. For that purpose, we introduced a set of Galilean-invariant features, based on normalised terms extracted from the governing equations, that distinguish regions with specific dominant dynamics, such as irrotational regions, boundary layers, and back-flow. A fuzzy clustering algorithm, namely, the Gaussian Model Mixture (GMM) was used to learn smooth probability distributions of a flow state to belong to one of the clusters. During calibration, models are adjusted to fit data from each cluster, improving accuracy within those clusters but potentially increasing errors outside them. The model predictions are then aggregated by using cluster probabilities, which assign higher weights to models trained at capturing the cluster dynamics. While CMA provides

a smoother and more efficient global prediction compared to XUMA's complex point-specific weights, challenges remain in accurately predicting quantities of interest not included in the calibration process.

Since CMA relies on clustering and high-fidelity data, a new strategy was developed, which relies on user-defined "building-block" flows for calibration, which avoids the need for clustering. The Building-Block Model Aggregation (BBMA) method uses a set of simple flows representative of relevant flow processes, like homogeneous turbulence and free shear layers, to generate a set of expert models. Applying BBMA to the 3D NACA65 V103 compressor cascade demonstrated that models trained on simple building block flows can be effectively combined to predict a complex flow. However, while BBMA improved accuracy in training scenarios, performance dropped for unseen scenarios due to inaccurate predicted weights. This highlights the challenges of predicting weights for new configurations, as the models struggled with unseen physical dynamics, and the need to use a larger and more diverse training set for the weights, to cover a wider range of flow processes.

It is important to stress that not only do the proposed aggregation techniques smartly combine competing models to achieve improved predictions but they are also equipped with an estimator for the predictive variance, i.e. a measure of turbulence modelling uncertainty. Such information is of value for mitigating risks and can be used as a criterion for model retraining or for moving to higher-fidelity models. However, predictions are not necessarily solutions to the RANS equations. To address this limitation, future approaches could enhance the weighting function used in aggregation by incorporating a penalisation term that forces the aggregated solution to respect the governing equations, particularly mass conservation. Alternatively, the mixture of models could be introduced directly within the CFD solver to ensure that the results are a solution to the RANS equations. However, implementing this would be a challenging endeavour, particularly within industrial CFD solvers.

A limitation common to all model aggregation methods is their decreased accuracy for quantities not directly used in calibration. In industrial applications, engineers often need to predict multiple quantities simultaneously, such as various aspects of compressor design. While it is feasible to calibrate on multiple quantities, this introduces the risk of sub-optimal parameter estimation as the model tries to balance competing priorities. Additionally, surrogate models may fail to capture important correlations between calibration quantities, further complicating the process. This is particularly relevant when applying BBMA, as results suggest that predictions for calibration quantities are more accurate than for others. Combining multiple calibration

quantities to compute space-dependent or cluster weights is feasible but requires careful implementation to avoid favouring one quantity over others and to ensure balanced performance across multiple metrics.

BBMA results also indicated that current turbulence models, particularly those based on linear eddy viscosity, fail to capture complex dynamics despite enhancement, as the QCR correction. Incorporating more advanced models could improve accuracy but would significantly increase the computational cost within the BBMA and CMA frameworks. More complex RANS models, Reynolds stress models for instance, would be required to capture intricate dynamics like corner flows and tip-leakage flows commonly seen in turbomachinery. These developments would be necessary to apply these methodologies to more complex configurations, such as the 3D NACA65 compressor cascade.

Looking forward, a promising avenue for future research lies in integrating BBMA's calibration framework with CMA's weighting strategy. This hybrid approach could involve developing expert hyper-models tailored to distinct physical dynamics, which could then be aggregated based on flow characteristics detected in the prediction scenario. Such an improved framework would require a richer set of features than those used in CMA, potentially incorporating features from RANS equations or additional constitutive relations, such as the vorticity equation. This would help streamline the number of models needed, with simpler models for basic dynamics and more complex models for intricate 3D scenarios.

In conclusion, while CMA and BBMA offer promising tools for predicting turbulent flow fields and associated uncertainties, they require further refinement before they can be widely adopted in industrial applications. Future work should focus on improving weight prediction methods, expanding the set of calibration quantities, and incorporating more sophisticated turbulence models. These advancements will enhance the robustness and applicability of these methodologies across a broader range of flow configurations, potentially transforming their role in real-world aerodynamic and turbomachinery design.

Appendix

A

MAP estimates

In this appendix, all the MAP estimates of all the parameters after calibration are gathered. The calibration methodology is detailed in chapter 3 and the various data against which the models are calibrated are described in chapters 5 to 7.

A.1 XUMA MAP estimates

The MAP estimates presented here are obtained after calibration on scenario S_4 against the total pressure field as described in chapter 5.

Param.	XUMA
β	0.18
σ	0.42
σ_s	0.3
σ_{do}	0.2
κ	0.46

Table A.1 – MAPs for $k-\omega$ model

Param.	XUMA
$c_{\epsilon 2}$	2.5
σ_k	0.13
κ	0.55
σ_η	1.4e-2

Table A.2 – MAPs for $k-\epsilon$ model.

Param.	XUMA
B1	33.3
E2	1.03
σ_l	2.23
σ_k	2.27
κ	0.43
σ_η	5e-3

Table A.3 – MAPs for $k-l$ model

Param.	XUMA
$cb1$	0.09
$cb2$	0.95
$cv1$	8.69
$cw2$	0.90
$cw3$	2.40
σ	0.60
κ	0.505
σ_η	1e-5

Table A.4 – MAPs for S-A model

A.2 CMA MAP estimates

The MAP estimates presented here are obtained after calibration on scenario S_4 in the distinct clusters described in chapter 6.

Param.	CMA - BL	CMA - WF	CMA - SBL	CMA - INV
β	0.10	0.11	0.18	0.18
σ	0.42	0.35	0.50	0.38
σ_s	0.30	0.30	0.30	1.00
σ_{do}	0.20	0.20	0.20	0.20
κ	0.55	0.40	0.44	0.48

Table A.5 – Nominal parameters and prior bounds for $k-\omega$ model.

Param.	CMA - BL	CMA - WF	CMA - SBL	CMA - INV
$c_{\epsilon 2}$	2.50	1.98	2.14	2.50
σ_k	0.08	0.08	0.08	0.13
κ	0.55	0.55	0.55	0.55

Table A.6 – Nominal parameters and prior bounds for $k-\epsilon$ model.

Param.	CMA - BL	CMA - WF	CMA - SBL	CMA - INV
B1	32.40	33.22	28.22	31.85
E2	1.00	1.06	0.88	1.03
σ_l	2.50	2.37	1.24	0.28
σ_k	2.39	2.15	0.65	1.48
κ	0.45	0.55	0.37	0.42

Table A.7 – Nominal parameters and prior bounds for $k-l$ model.

Param.	CMA - BL	CMA - WF	CMA - SBL	CMA - INV
$cb1$	0.08	0.07	0.09	0.10
$cb2$	0.50	0.51	0.50	1.00
$cv1$	8.69	11.67	8.85	8.52
$cw2$	0.90	0.10	0.90	0.77
$cw3$	2.40	0.80	2.40	2.40
σ	0.66	1.31	0.60	0.60
κ	0.46	0.36	0.47	0.50

Table A.8 – Nominal parameters and prior bounds for SA model.

A.3 BBMA MAP estimates

Param.	BBMA - BL	BBMA - SBL	BBMA - WF	BBMA - SD
B1	12.41	20.21	27.38	NA
E2	1.45	1.55	0.46	NA
σ_l	3.27	0.65	2.73	NA
σ_k	0.20	2.50	0.20	NA
κ	0.43	0.55	0.42	NA

Table A.9 – MAPs for $k-l$ model

Param.	BBMA - BL	BBMA - SBL	BBMA - WF	BBMA - SD
$cb1$	0.10	0.07	0.15	NA
$cb2$	0.62	0.50	1.00	NA
$cv1$	7.39	6.01	13.30	NA
$cw2$	0.82	0.10	0.90	NA
$cw3$	1.36	0.80	1.55	NA
σ	1.50	0.81	1.12	NA
κ	0.40	0.55	0.48	NA

Table A.10 – MAPs for S-A model

Param.	BBMA - BL	BBMA - SBL	BBMA - WF	BBMA - SD
B1	35.97	25.47	19.31	18.48
E2	0.98	1.60	0.68	0.60
σ_l	3.24	0.65	2.99	1.82
σ_k	0.20	2.50	0.20	0.20
κ	0.42	0.55	0.40	0.55
QCR	0.00	0.50	0.00	0.50

Table A.11 – MAPs for $k-l$ QCR model.

Param.	BBMA - BL	BBMA - SBL	BBMA - WF	BBMA - SD
$cb1$	0.09	0.07	0.15	0.12
$cb2$	1.00	0.50	1.00	1.00
$cv1$	7.14	6.01	12.99	6.00
$cw2$	0.90	0.10	0.90	0.90
$cw3$	0.90	0.80	1.69	2.40
σ	1.33	0.60	1.04	1.41
κ	0.41	0.55	0.48	0.55
QCR	0.00	0.50	0.00	0.50

Table A.12 – MAPs for S-A QCR model

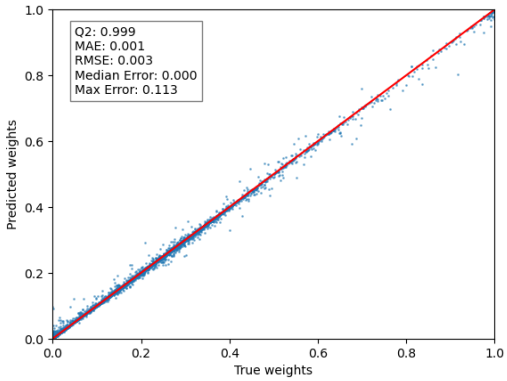
B

Performance Metrics for Random Forest Regressor Models

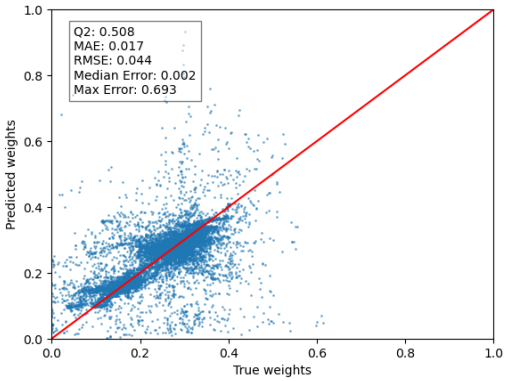
In chapter 5 and chapter 7, Random Forest Regressor is used to map the weights onto the features space composed of the proposed set by Ling & Templeton [64]. The Following figures display the predictive results of the RFRs.

B.1 XUMA Random Forest Regressor

The two scatter plots represent the predicted versus true weights using a Random Forest Regressor (RFR) model, with figure B.1a showing results for training and predicting on the same scenario (S_4) and figure B.1b showing results for training on ${}_S4$ and predicting on a different scenario (S_3). In figure B.1b, the model performs exceptionally well, as indicated by a high Q^2 of 0.999, a low RMSE of 0.003, and tightly clustered points around the red line. This reflects the model's ability to capture the underlying relationships when training and prediction occur on



(a) Prediction on S_4 after training on S_4.



(b) Prediction on S_3 after training on S_4.

Figure B.1 – Comparison of predicted vs. true weights using Random Forest Regressor (RFR) models. In (a), the model is trained and tested on the same scenario (S_4), while in (b), it is trained on S_4 and tested on S_3. The red line represents the ideal prediction scenario where predicted and true values match perfectly.

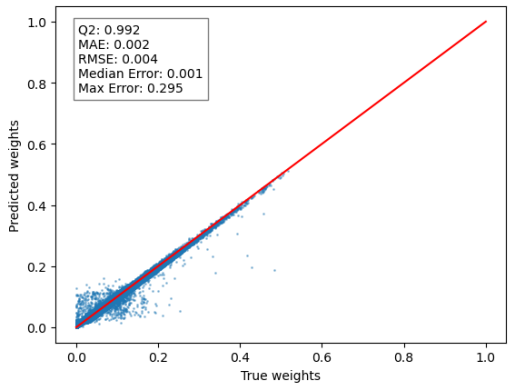
the same dataset, resulting in highly accurate predictions.

In contrast,figure B.1b reveals a significant drop in predictive performance when the model is applied to a new scenario (S3). The Q^2 value decreases to 0.508, and both the RMSE and Max Error increase considerably (to 0.044 and 0.693, respectively). The scatter points are more spread out, especially for higher true weights, which suggests that the model struggles to generalise effectively to new data not seen during training. This demonstrates a potential overfitting issue or a lack of robustness when transferring knowledge between different scenarios.

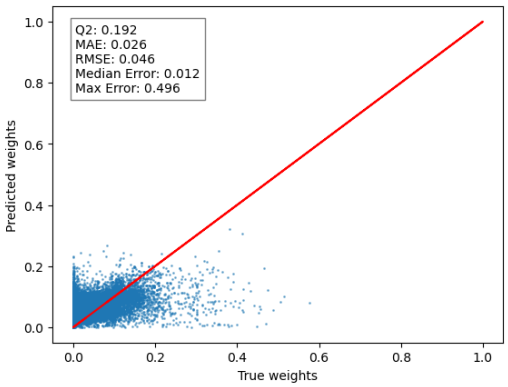
B.2 BBMA Random Forest Regressor

The two figures present the predicted vs. true weights after training the Random Forest Regressor (RFR) model under different scenarios. In Figure B.2a, the model is both trained and tested on the same scenario (4DEG). The high Q^2 score of 0.992, along with low MAE (0.002) and RMSE (0.004), demonstrate excellent model performance, with most of the predictions closely following the ideal red line. This suggests that the model captures the underlying patterns in the data well when it is applied to the same scenario for both training and testing.

Figure B.2b shows the results when the model is trained on a different scenario (4DEG) and tested on 7DEG. Here, the Q^2 drops significantly to 0.192, with a noticeable increase in



(a) Prediction on 4DEG after training on 4DEG.



(b) Prediction on 7DEG after training on 4DEG.

Figure B.2 – Comparison of predicted vs. true weights using Random Forest Regressor (RFR) models. In (a), the model is trained and tested on the same scenario (4DEG), while in (b), it is trained on 4DEG and tested on 7DEG. The red line represents the ideal prediction scenario where predicted and true values match perfectly.

RMSE (0.046) and Max Error (0.496). The scatter points are more concentrated near the lower end of the true weight range, with many predictions failing to align with the ideal line. This indicates that the model struggles to generalise from the training scenario (4DEG) to a different test scenario (7DEG), resulting in a considerable loss of predictive accuracy. This comparison highlights the challenge of model generalisation across different data scenarios.

The performance discrepancy between post-diction (B.1a and B.2a) and prediction (B.1b and B.2b) figures strongly suggests that the current feature set used by the Random Forest Regressor (RFR) model is not able to generalise well across different scenarios. While the model performs exceptionally well when trained and tested on the same scenario (as in 4DEG), the drastic reduction in accuracy when applied to a new, unseen scenario (7DEG) highlights its limitations. This indicates that the features learned during training are scenario-dependent and may lack the necessary generalisation capacity to predict weights reliably across different configurations. Future work should focus on refining the feature set or exploring alternative modelling techniques to improve cross-scenario generalisation.

APPENDIX B. PERFORMANCE METRICS FOR RANDOM FOREST REGRESSOR MODELS

C

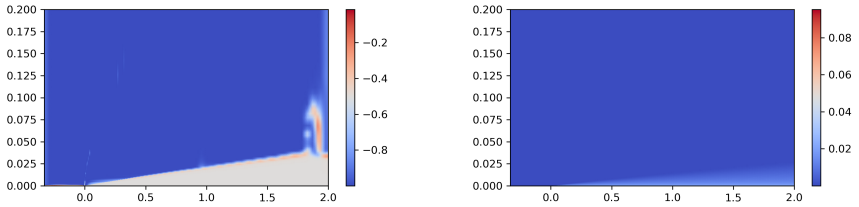
Clustering the feature space

C.1 Clustering using Ling's features

As suggested in Chapter 6, the covariate space can also be represented using Ling's features [64] to be fed to the GMM clustering algorithm to discover the different physical dynamics present in the flow.

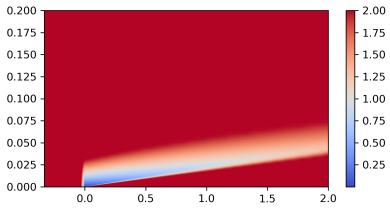
C.1.1 ZPG Turbulent Boundary Layer (BL)

A quick inspection of the features displayed in figure C.1 suggests that the features seem to be able to discriminate the boundary layer from the free stream. Figure C.2 further validates this, as using GMMs with 2 clusters, using either tied or full covariance matrices, separates the boundary layer. Adding more clusters will only capture transitions.

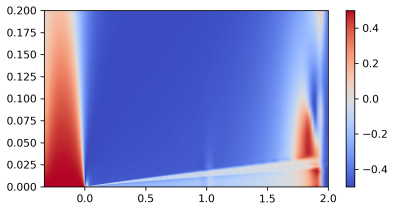


(a) Feature 1

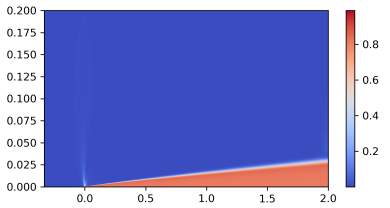
(b) Feature 2



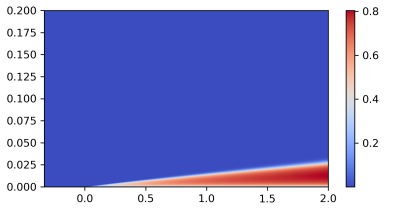
(c) Feature 3



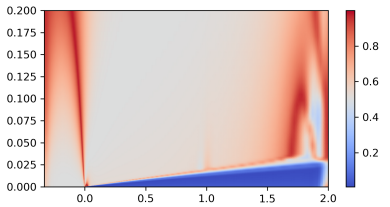
(d) Feature 4



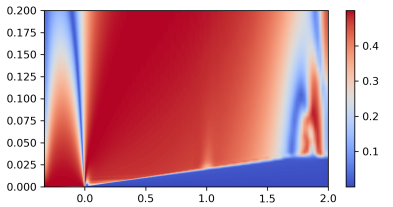
(e) Feature 5



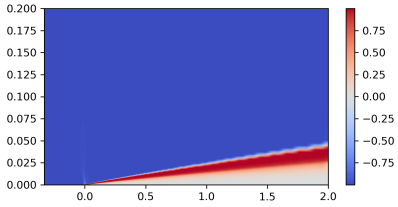
(f) Feature 6



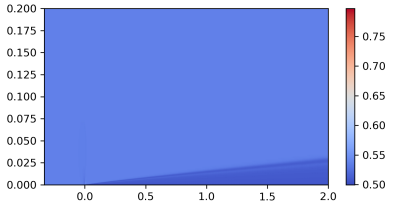
(g) Feature 7



(h) Feature 8

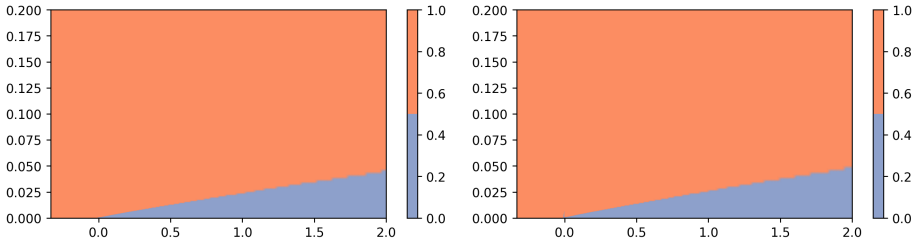


(i) Feature 9



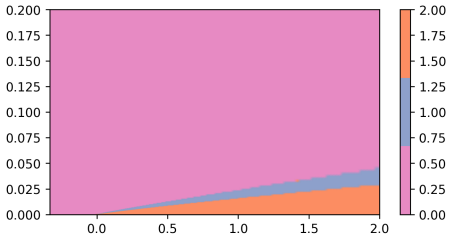
(j) Feature 10

Figure C.1 – Features on BL

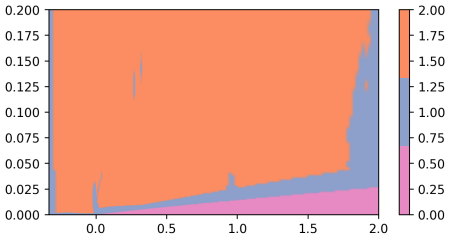


(a) $n_c = 2$, tied covariance

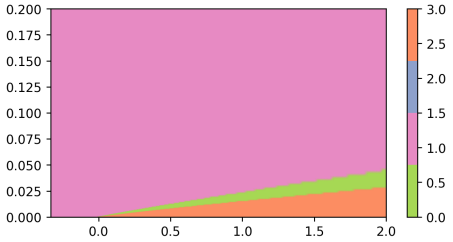
(b) $n_c = 2$, full covariance



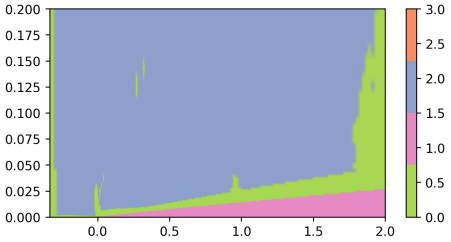
(c) $n_c = 3$, tied covariance



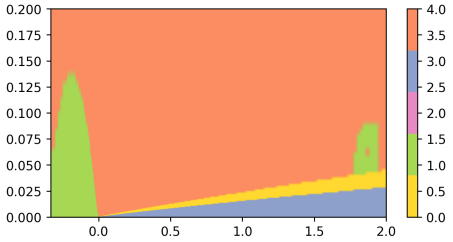
(d) $n_c = 3$, full covariance



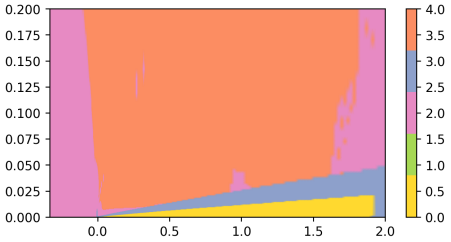
(e) $n_c = 4$, tied covariance



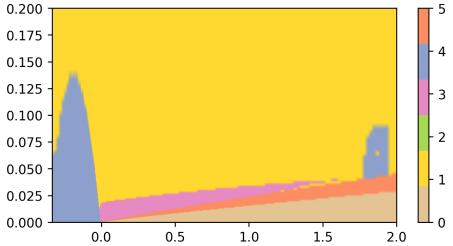
(f) $n_c = 4$, full covariance



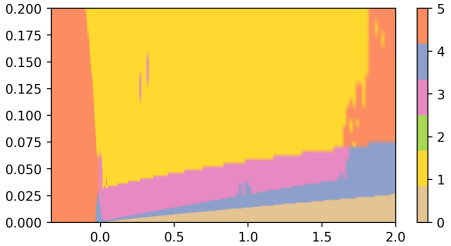
(g) $n_c = 5$, tied covariance



(h) $n_c = 5$, full covariance



(i) $n_c = 6$, tied covariance



(j) $n_c = 6$, full covariance

Figure C.2 – Clustering results on BL using various numbers of clusters and covariance type.

C.1.2 Wake Flow

Figure C.3 displays the features of the Wake Flow configuration and suggests that the wake and boundary layers could be separated from the free stream. Furthermore, figure C.4 indicates that both types of covariances in the GMM struggle to separate the viscous-dominated regions using only 2 clusters. Using 3 will separate the viscous regions and detect some parts of the wake. Only adding a 4th cluster will properly separate the wake.

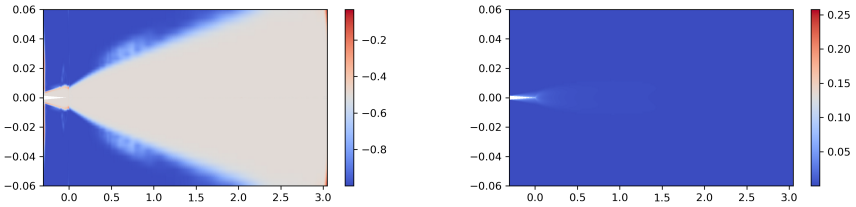
C.1.3 Backwards-Facing Step

Figure C.5 displays the features and figure C.6 displays the clustering results on the SBL configuration. Using 2 clusters and full covariances adequately separates the boundary layers from the core flow. However, one need to use many more cluster to adequately separate the separated boundary layer. Using this many clusters (at least 6) implies that there are a lot of clusters representing transitions or sub-clusters.

C.1.4 NACA65

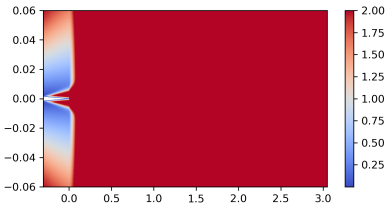
Similarly, as before, figure C.7 displays the features and figure C.8 displays the clustering results on the 2D NACA65 configuration. Clustering results using tied covariances seems more accurate than the ones using full ones. Using 5 clusters separate the potential flow, the wake, the boundary layer close to the blade and a cluster including the separation.

In conclusion, using Ling's features seems to separate the potential inviscid flow for the viscous-dominated regions. However, it does not separate the different dynamics as well as the new set of features even though it comprises more features.

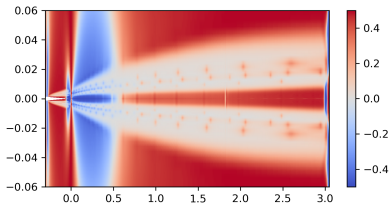


(a) Feature 1

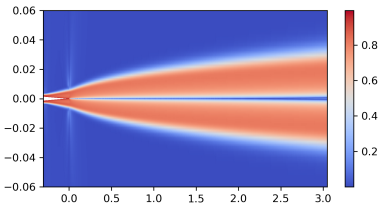
(b) Feature 2



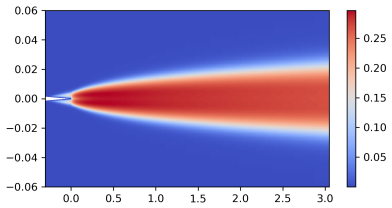
(c) Feature 3



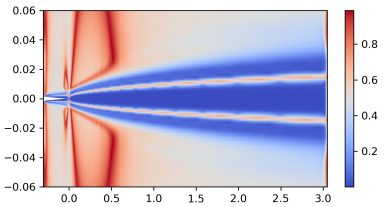
(d) Feature 4



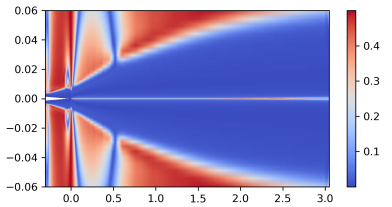
(e) Feature 5



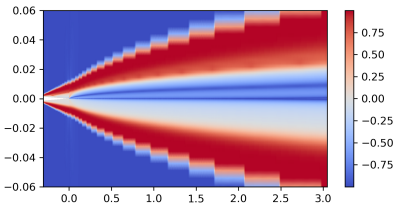
(f) Feature 6



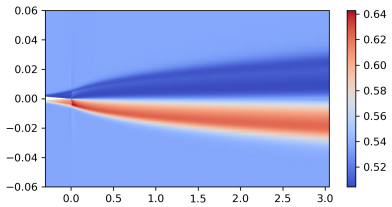
(g) Feature 7



(h) Feature 8

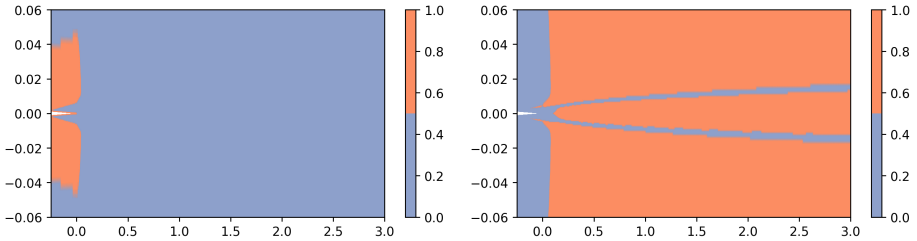


(i) Feature 9



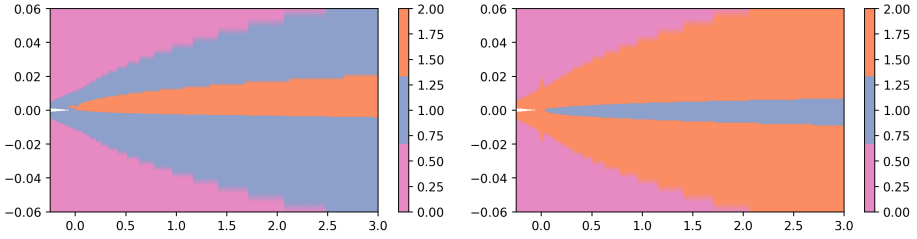
(j) Feature 10

Figure C.3 – Features on WF



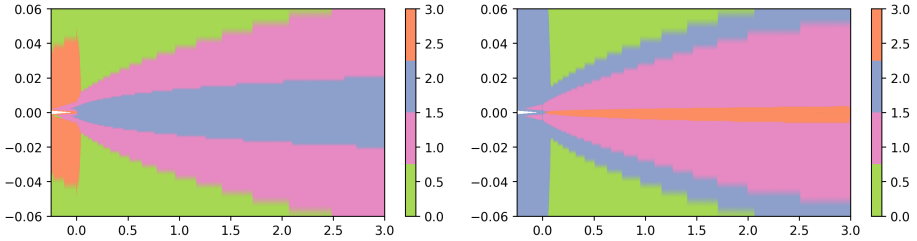
(a) $n_c = 2$, tied covariance

(b) $n_c = 2$, full covariance



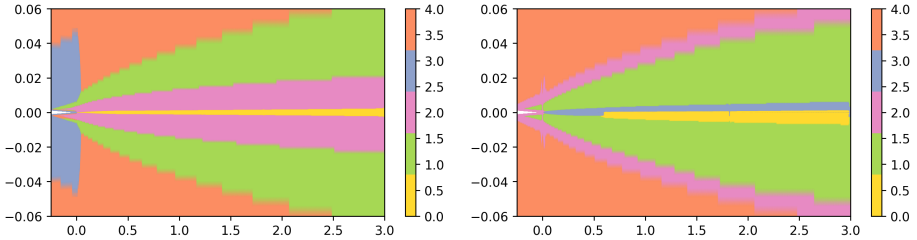
(c) $n_c = 3$, tied covariance

(d) $n_c = 3$, full covariance



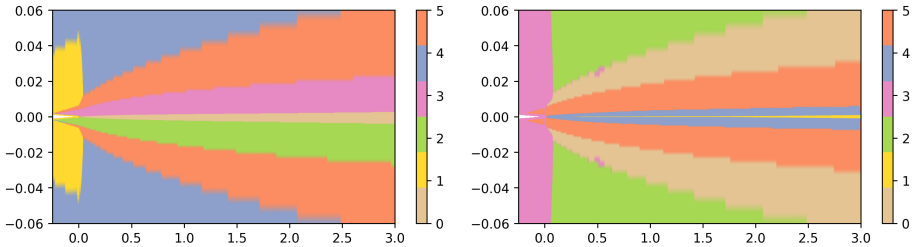
(e) $n_c = 4$, tied covariance

(f) $n_c = 4$, full covariance



(g) $n_c = 5$, tied covariance

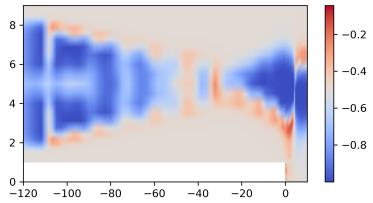
(h) $n_c = 5$, full covariance



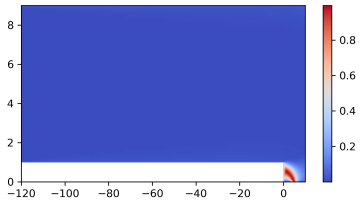
(i) $n_c = 6$, tied covariance

(j) $n_c = 6$, full covariance

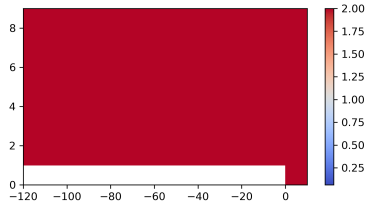
Figure C.4 – Clustering results on WF using various numbers of clusters and covariance type



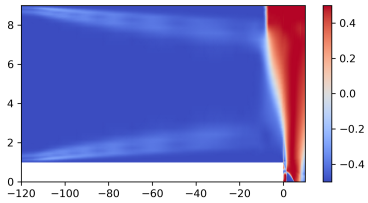
(a) Feature 1



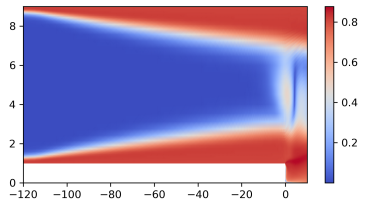
(b) Feature 2



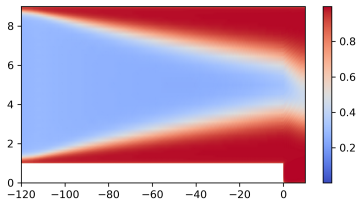
(c) Feature 3



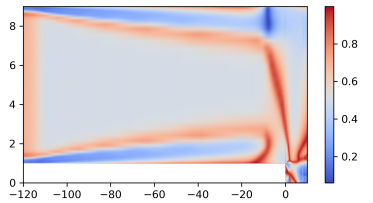
(d) Feature 4



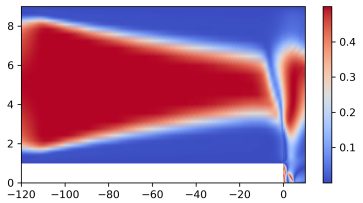
(e) Feature 5



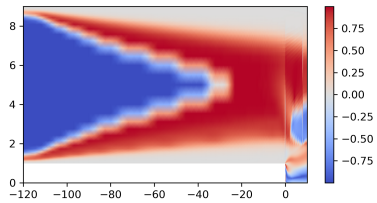
(f) Feature 6



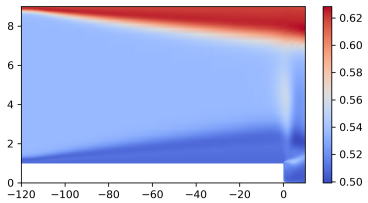
(g) Feature 7



(h) Feature 8

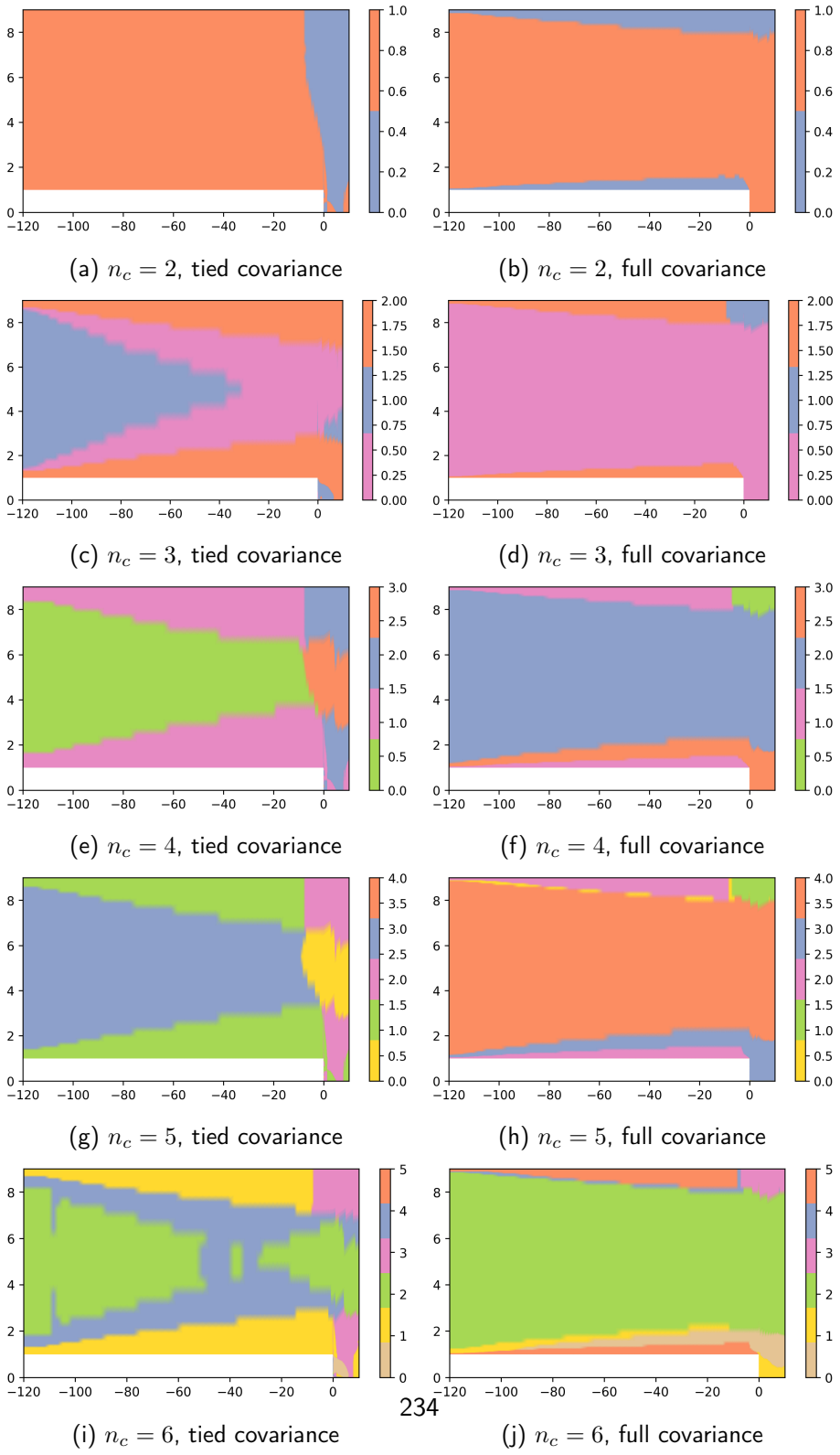


(i) Feature 9



(j) Feature 10

Figure C.5 – Features on SBL



(a) $n_c = 2$, tied covariance

(b) $n_c = 2$, full covariance

(c) $n_c = 3$, tied covariance

(d) $n_c = 3$, full covariance

(e) $n_c = 4$, tied covariance

(f) $n_c = 4$, full covariance

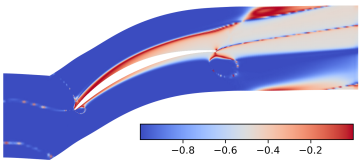
(g) $n_c = 5$, tied covariance

(h) $n_c = 5$, full covariance

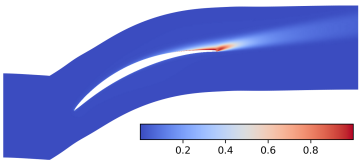
(i) $n_c = 6$, tied covariance

(j) $n_c = 6$, full covariance

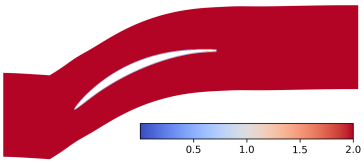
Figure C.6 – Clustering results on SBL using various numbers of clusters and covariance type



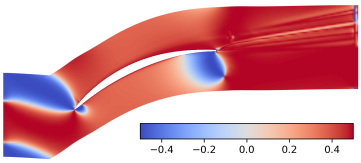
(a) Feature 1



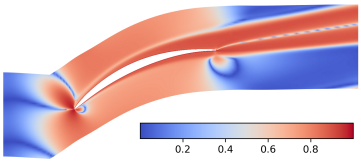
(b) Feature 2



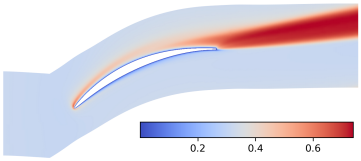
(c) Feature 3



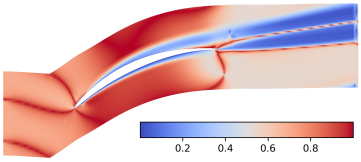
(d) Feature 4

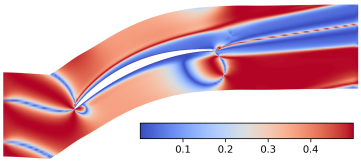


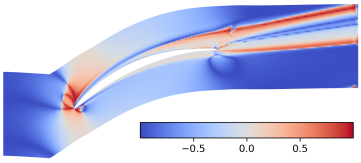
(e) Feature 5

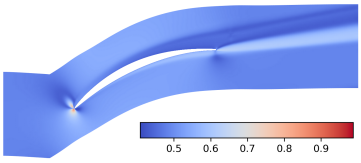


(f) Feature 6





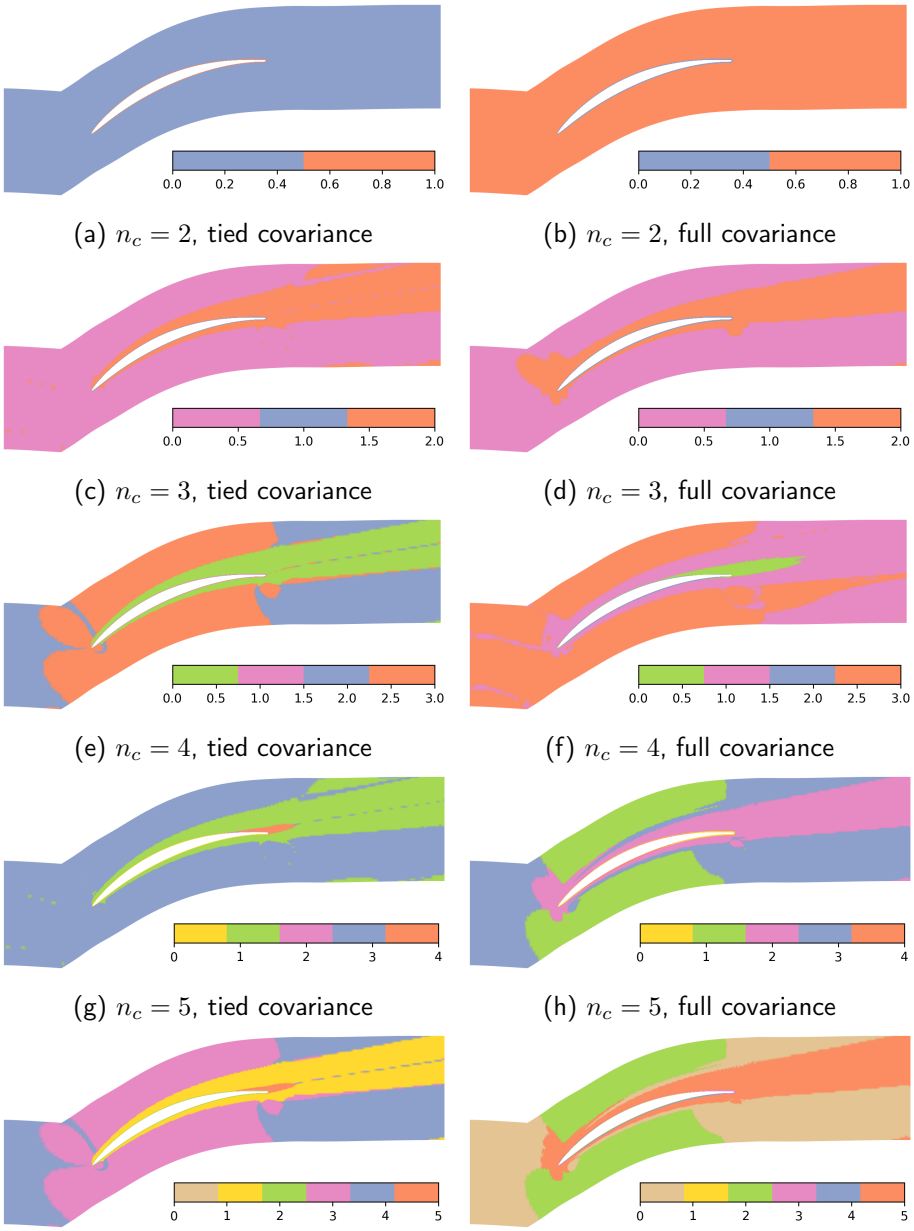




(i) Feature 9

(j) Feature 10

Figure C.7 – Features on NACA65



(a) $n_c = 2$, tied covariance (b) $n_c = 2$, full covariance

(c) $n_c = 3$, tied covariance (d) $n_c = 3$, full covariance

(e) $n_c = 4$, tied covariance (f) $n_c = 4$, full covariance

(g) $n_c = 5$, tied covariance (h) $n_c = 5$, full covariance

(i) $n_c = 6$, tied covariance (j) $n_c = 6$, full covariance

Figure C.8 – Clustering results on NACA65 using various numbers of clusters and covariance type

C.2 Clustering using the novel set of features and full covariance in the GMMs

C.2.1 WF

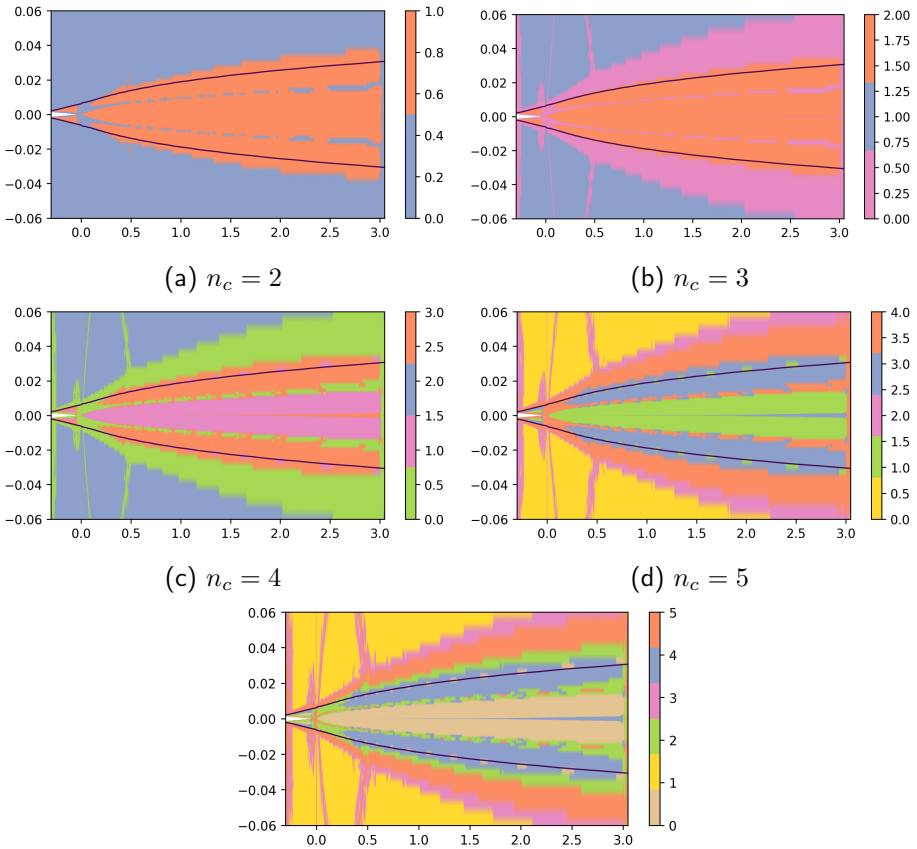
Clustering results using a full covariance type are presented in figure C.9. Using only two clusters seems to separate the free stream from the stream impacted by the flat plate. The black line indicates the contour where $u = 0.99u_{\text{inf}}$ and suggests that with only two clusters the GMM will be cautious in including all the points impacted by viscous forces in the turbulent cluster. Using three clusters, no difference is observed between the wake and the boundary layer; instead, the transition region between the wake and the freestream is identified. Using four clusters eventually separates the wake from boundary layers. Adding more clusters does not discover any more physical dynamics and only separates noise.

C.2.2 SBL

Clustering results using a full covariance type are presented in figure C.10. Results using only two clusters would only discriminate the boundary layers. A third cluster would separate the recirculation bubble. The clustering step will not however encapsulate the whole recirculation bubble. Adding more cluster does not seem to separate any more physical dynamics

C.2.3 NACA65

Clustering results using a tied covariance type are presented in figure C.11. In this case, using two clusters allows only to discriminate the reverse flow region from the rest of the flow. Using three clusters, a cluster corresponding to the boundary layer and wake is distinguished from the separated region and the free stream. However, this cluster seems to encapsulate more points than using a tied covariance. Distinct clusters for the boundary layer and wake are discovered by using four clusters but the separation is no longer captured. Adding more clusters will not discover new dynamics or separate further the previously found ones, but rather cluster transition from one cluster to the others.

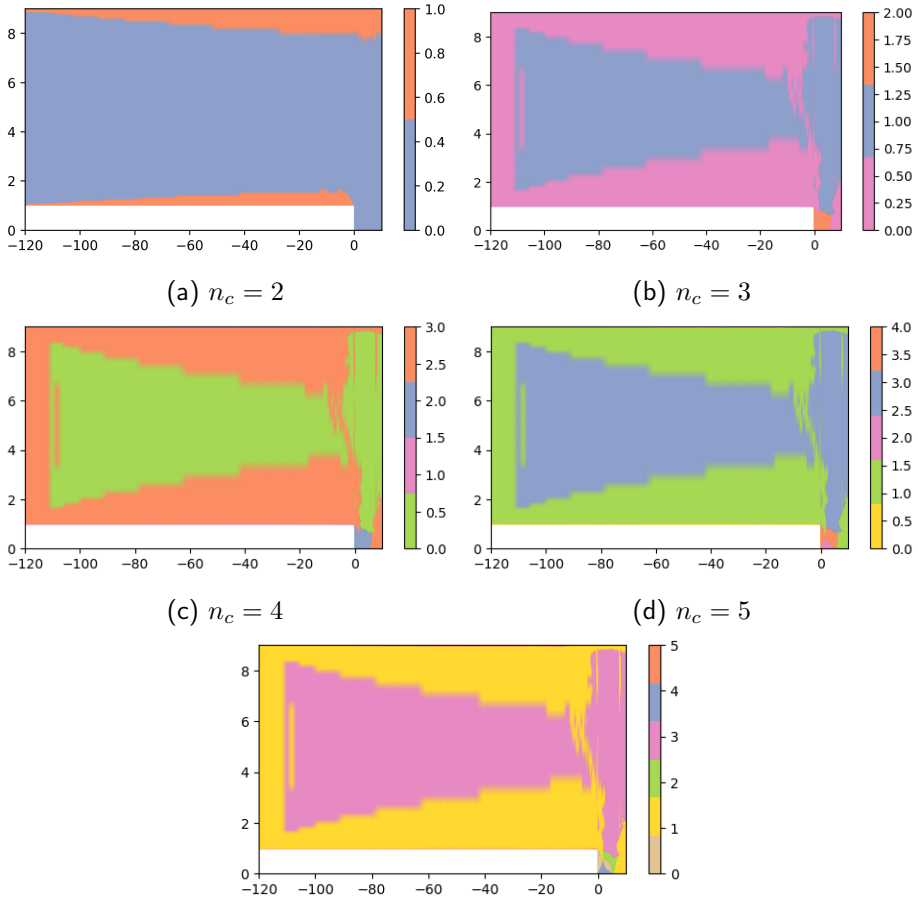


(a) $n_c = 2$ (b) $n_c = 3$

(c) $n_c = 4$ (d) $n_c = 5$

(e) $n_c = 6$

Figure C.9 – Clustering results on WF using various numbers of clusters and covariance type



(a) $n_c = 2$

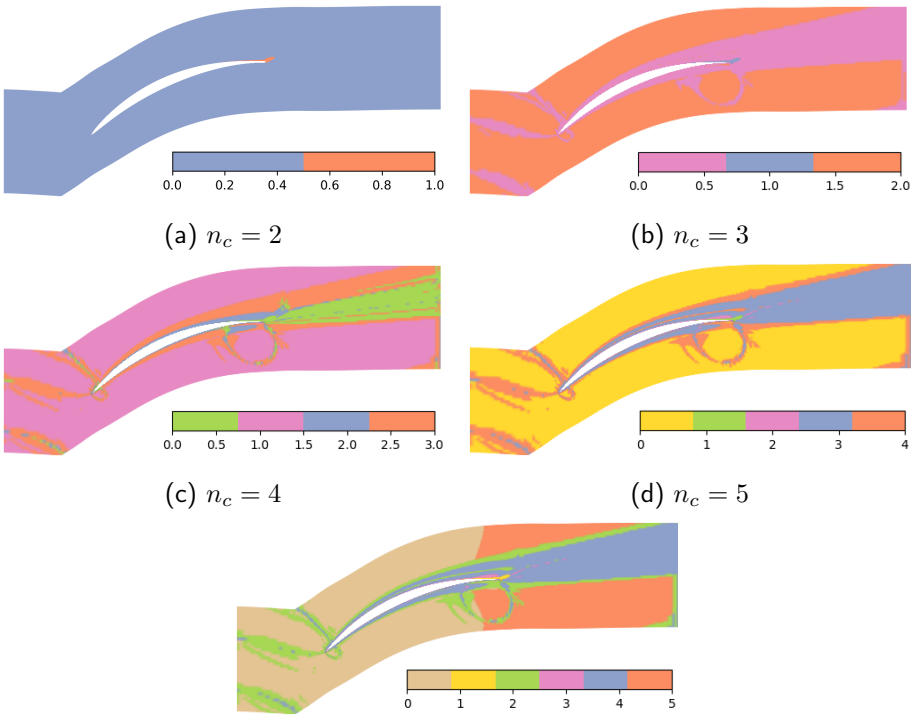
(b) $n_c = 3$

(c) $n_c = 4$

(d) $n_c = 5$

(e) $n_c = 6$

Figure C.10 – Clustering results on SBL using various numbers of clusters and covariance type



(a) $n_c = 2$ (b) $n_c = 3$

(c) $n_c = 4$ (d) $n_c = 5$

(e) $n_c = 6$

Figure C.11 – Clustering results on NACA65 using various numbers of clusters and covariance type

Bibliography

[1] Jane Amilhat. Fly the Green Deal, Europe's Vision for Sustainable Aviation, Report of the Advisory Council for Aviation Research and Innovation in Europe (ACARE). (Cited on page 16.)

[2] Paul D. Arendt, Daniel W. Apley, and Wei Chen. Quantification of Model Uncertainty: Calibration, Model Discrepancy, and Identifiability. *Journal of Mechanical Design*, 134(10):100908, October 2012. (Cited on page 19.)

[3] M Arnst, R Ghanem, and C Soize. Identification of Bayesian posteriors for coefficients of chaos expansions. *Journal of Computational Physics*, 2010. (Cited on page 89.)

[4] American Institute of Aeronautics and Astronautics. *AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*. American Institute of Aeronautics and Astronautics, 1998. Google-Books-ID: OUAHAAAACAAJ. (Cited on page 17.)

[5] Chloe-Agathe Azencott. *INTRODUCTION AU MACHINE LEARNING*. DUNOD, S.l., 2019. OCLC: 1236251161. (Cited on page 120.)

[6] Andrea D. Beck, Jonas Zeifang, Anna Schwarz, and David G. Flad. A neural network based shock detection and localization approach for discontinuous Galerkin methods. *Journal of Computational Physics*, 423:109824, December 2020. (Cited on page 120.)

[7] Peter Benner, Serkan Gugercin, and Karen Willcox. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Review*, 57(4):483–531, January 2015. (Cited on page 65.)

[8] Christophe Benoit, Stéphanie Péron, and Sam Landier. Cassiopee: A CFD pre- and post-processing tool. *Aerospace Science and Technology*, 45:272–283, 2015. (Cited on page 42.)

[9] G Berkooz. The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows. page 38. (Cited on page 65.)

[10] Biswarup Bhattacharyya. Uncertainty quantification of dynamical systems by a POD–Kriging surrogate model. *Journal of Computational Science*, 60:101602, April 2022. (Cited on page 65.)

[11] Jeff A Bilmes. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. 1998. (Cited on page 124.)

[12] Christopher M Bishop. Pattern Recognition and Machine Learning. (Cited on pages 66, 120 and 124.)

[13] Avrim Blum. Empirical Support for Winnow and Weighted-Majority Algorithms: Results on a Calendar Scheduling Domain. *Machine Learning*, 26(1):5–23, January 1997. (Cited on page 27.)

[14] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, 2020. _eprint: https://doi.org/10.1146/annurev-fluid-010719-060214. (Cited on page 120.)

[15] Jared L Callaham. Learning dominant physical processes with data-driven balance models. *Nature Communications*, 12(1016), 2021. (Cited on pages 121, 122, 123, 137 and 166.)

[16] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, 2000. (Cited on page 65.)

[17] Sai Hung Cheung, Todd A. Oliver, Ernesto E. Prudencio, Serge Prudhomme, and Robert D. Moser. Bayesian uncertainty analysis with applications to turbulence modeling. *Reliability Engineering & System Safety*, 96(9):1137–1149, September 2011. (Cited on pages 20, 25, 46, 47, 48, 55, 57 and 89.)

[18] Brendan Colvert, Mohamad Alsalman, and Eva Kanso. Classifying vortex wakes using neural networks. *Bioinspiration & Biomimetics*, 13(2):025003, February 2018. arXiv:1709.02792 [physics]. (Cited on page 120.)

[19] M. De Zordo-Banliat, G. Dergham, X. Merle, and P. Cinnella. Space-dependent turbulence model aggregation using machine learning. *Journal of Computational Physics*, 497:112628, January 2024. (Cited on pages 28, 61, 62, 63, 64, 75, 79 and 82.)

[20] 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, April 2020. (Cited on pages 26, 27, 28, 56, 62, 67 and 75.)

[21] Maximilien de Zordo-Banliat. *Quantification and reduction of turbulence modeling uncertainties in turbomachinery flows using Bayesian multi-model ensembles*. PhD thesis, HESAM Université, Ecole Nationale Supérieure d'Arts et Métiers, 2022. (Cited on pages 28, 49, 70, 75 and 213.)

[22] Maximilien de Zordo-Banliat, Xavier Merle, Gregory Dergham, and Paola 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*, ahead-of-print(ahead-of-print), January 2022. (Cited on pages 26, 27, 28, 59, 67 and 75.)

[23] Frank Dellaert. The Expectation Maximization Algorithm. 2002. (Cited on page 125.)

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1977.tb01600.x. (Cited on page 28.)

[25] Raphaël Deswarte, Véronique Gervais, Gilles Stoltz, and Sébastien Da Veiga. Sequential model aggregation for production forecasting. *Computational Geosciences*, 23(5):1107–1124, October 2019. (Cited on pages 61, 62 and 63.)

[26] Eric Dow and Qiqi Wang. Quantification of Structural Uncertainties in the k - ω Turbulence Model. page 12, 2011. (Cited on page 22.)

[27] David Draper. Assessment and Propagation of Model Uncertainty. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 57(1):45–70, January 1995. (Cited on pages 26, 55 and 58.)

[28] David M. Driver and H. Lee Seegmiller. Features of a reattaching turbulent shear layer in divergent channelflow. *AIAA Journal*, May 2012. (Cited on page 72.)

[29] Dorian Dupuy, Nicolas Odier, Corentin Lapeyre, and Dimitrios Papadogiannis. Modeling the wall shear stress in large-eddy simulation using graph neural networks. *Data-Centric Engineering*, 4:e7, 2023. (Cited on page 76.)

[30] Karthik Duraisamy. Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence. *Physical Review Fluids*, 6(5):050504, May 2021. (Cited on page 21.)

[31] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence Modeling in the Age of Data. *Annual Review of Fluid Mechanics*, 51(1):357–377, January 2019. (Cited on page 19.)

[32] Karthikeyan Duraisamy, Ze J. Zhang, and Anand Pratap Singh. New Approaches in Turbulence and Transition Modeling Using Data-driven Techniques. *53rd AIAA Aerospace Sciences Meeting*, January 2015. (Cited on page 21.)

[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*, 258:73–94, February 2014. (Cited on pages 25, 67, 70 and 89.)

[34] W. N. Edeling, G. Iaccarino, and P. Cinnella. Data-Free and Data-Driven RANS Predictions with Quantified Uncertainty. *Flow, Turbulence and Combustion*, 100(3):593–616, April 2018. (Cited on page 23.)

[35] W.N. Edeling, P. Cinnella, and R.P. Dwight. Predictive RANS simulations via Bayesian Model-Scenario Averaging. *Journal of Computational Physics*, 275:65–91, October 2014. (Cited on pages 20, 25, 26, 28, 56, 62 and 67.)

[36] 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, May 2018. Publisher: American Institute of Aeronautics and Astronautics. (Cited on pages 26, 27, 28, 57, 59, 62 and 81.)

[37] Michael Emory, Johan Larsson, and Gianluca Iaccarino. Modeling of structural uncertainties in Reynolds-averaged Navier-Stokes closures. *Physics of Fluids*, 25(11):110822, November 2013. (Cited on page 22.)

[38] L. Eça, M. Hoekstra, and Patrick Roache. Verification of Calculations: An Overview of the 2nd Lisbon Workshop. In *18th AIAA Computational Fluid Dynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, June 2007. (Cited on page 73.)

[39] Luís Eça, Martin Hoekstra, and Patrick Roache. Verification of Calculations: an Overview of the Lisbon Workshop. In *23rd AIAA Applied Aerodynamics Conference*, Fluid Dynamics

and Co-located Conferences. American Institute of Aeronautics and Astronautics, June 2005. (Cited on page 73.)

[40] Luís Eça, Martin Hoekstra, Patrick Roache, and Hugh Coleman. Code Verification, Solution Verification and Validation: An Overview of the 3rd Lisbon Workshop. In *19th AIAA Computational Fluid Dynamics*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, June 2009. (Cited on page 73.)

[41] Andrea Ferrero, Angelo Iollo, and Francesco Larocca. Field inversion for data-augmented RANS modelling in turbomachinery flows. *Computers & Fluids*, 201:104474, April 2020. (Cited on page 22.)

[42] Tiago M. Fragoso, Wesley Bertoli, and Francisco Louzada. Bayesian Model Averaging: A Systematic Review and Conceptual Classification: BMA: A Systematic Review. *International Statistical Review*, 86(1):1–28, April 2018. (Cited on page 26.)

[43] Michael Frank, Dimitris Drikakis, and Vassilis Charissis. Machine-Learning Methods for Computational Science and Engineering. *Computation*, 8(1):15, March 2020. (Cited on page 120.)

[44] F Gao, G Zambonini, J Boudet, X Ottavy, L Lu, and L Shao. Unsteady behavior of corner separation in a compressor cascade: Large eddy simulation and experimental study. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 229(5):508–519, August 2015. (Cited on page 76.)

[45] Feng Gao. *Advanced Numerical Simulation of Corner Separation in a Linear Compressor Cascade*. phdthesis, Ecole Centrale de Lyon, April 2014. (Cited on page 77.)

[46] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis, Third Edition*. 2003. (Cited on pages 20, 45, 49 and 82.)

[47] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis Third edition (with errors fixed as of 2 November 2021)*. 2003. (Cited on pages 20, 53 and 59.)

[48] James Hammond, Nick Pepper, Francesco Montomoli, and Vittorio Michelassi. Machine Learning Methods in CFD for Turbomachinery: A Review. *International Journal of Turbomachinery, Propulsion and Power*, 7(2):16, May 2022. (Cited on page 17.)

[49] Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian Model Averaging: A Tutorial. page 36, 1999. (Cited on pages 26, 55 and 56.)

[50] Matthew D Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. page 31, 2014. (Cited on pages 49, 50 and 90.)

[51] Gianluca Iaccarino, Aashwin Ananda Mishra, and Saman Ghili. Eigenspace perturbations for uncertainty estimation of single-point turbulence closures. *Physical Review Fluids*, 2(2):024605, February 2017. (Cited on page 23.)

[52] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87, March 1991. (Cited on page 27.)

[53] V. Roshan Joseph, Evren Gul, and Shan Ba. Maximum projection designs for computer experiments. *Biometrika*, 102(2):371–380, 2015. Publisher: [Oxford University Press, Biometrika Trust]. (Cited on page 66.)

[54] R.E. Kass and Raftery. Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995. (Cited on page 54.)

[55] Marc C. Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001. (Cited on pages 19, 20, 24 and 46.)

[56] Stephen Jay Kline, Donald Earl Coles, and E. A. Hirst. *Computation of Turbulent Boundary Layers–1968 AFOSR-IFP-Stanford Conference: Compiled data. Editors: D. E. Coles and E. A. Hirst. Bibliography (p. 18-19)*. Thermosciences Division, Stanford University, 1969. Google-Books-ID: 1AFRAAAAMAAJ. (Cited on page 70.)

[57] Tanya Kostova-Vassilevska and Geoffrey M. Oxberry. Model reduction of dynamical systems by proper orthogonal decomposition: Error bounds and comparison of methods using snapshots from the solution and the time derivatives. *Journal of Computational and Applied Mathematics*, 330:553–573, March 2018. (Cited on page 65.)

[58] B E Launder and B I Sharma. APPLICATION OF THE ENERGY-DISSIPATION MODEL OF TURBULENCE TO THE CALCULATION OF FLOW NEAR A SPINNING DISC. 1(2):7. (Cited on pages 39, 79, 88 and 139.)

[59] Cambier Laurent, Heib Sébastien, and Plot Sylvie. The Onera elsA CFD software: input from research and feedback from industry. 2013. (Cited on page 42.)

[60] Binglin Li, Zixuan Yang, Xing Zhang, Guowei He, Bing-Qing Deng, and Lian Shen. Using machine learning to detect the turbulent region in flow past a circular cylinder. *Journal of Fluid Mechanics*, 905(A10), 2020. (Cited on page 120.)

[61] Y.C. Liang, H.P. Lee, S.P. Lim, W.Z. Lin, K.H. Lee, and C.G. Wu. PROPER ORTHOGONAL DECOMPOSITION AND ITS APPLICATIONS—PART I: THEORY. *Journal of Sound and Vibration*, 252(3):527–544, May 2002. (Cited on page 65.)

[62] J. Ling and J. Templeton. Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty. *Physics of Fluids*, 27(8):085103, August 2015. (Cited on pages 82 and 100.)

[63] Julia Ling, Reese Jones, and Jeremy Templeton. Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318:22–35, August 2016. (Cited on pages 84 and 122.)

[64] 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, November 2016. (Cited on pages 23, 223 and 227.)

[65] Adrián Lozano-Durán and H Jane Bae. Machine learning building-block-flow wall model for large-eddy simulation. (Cited on page 170.)

[66] Adrián Lozano-Durán and Hyunji Jane Bae. Self-critical machine-learning wall-modeled LES for external aerodynamics, May 2021. arXiv:2012.10005 [physics]. (Cited on page 29.)

[67] Qinghua Lu, Li Wang, and Longsuo Li. Efficient uncertainty quantification of stochastic problems in CFD by combination of compressed sensing and POD-Kriging. *Computer Methods in Applied Mechanics and Engineering*, 396:115118, June 2022. ADS Bibcode: 2022CMAME.39615118L. (Cited on pages 57 and 65.)

[68] Kjetil O. Lye, Siddhartha Mishra, and Deep Ray. Deep learning observables in computational fluid dynamics. *Journal of Computational Physics*, 410:109339, June 2020. (Cited on page 57.)

[69] David Madigan and Adrian E. Raftery. Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam's Window. *Journal of the American Statistical Association*, 89(428):1535–1546, December 1994. (Cited on page 55.)

[70] L. Margheri, M. Meldi, M. V. Salvetti, and P. Sagaut. Epistemic uncertainties in RANS model free coefficients. *Computers & Fluids*, 102:315–335, October 2014. (Cited on page 88.)

[71] Georges Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, December 1963. (Cited on page 66.)

[72] A Melling and J H Whitelaw. Turbulent flow in a rectangular duct. (Cited on page 74.)

[73] X Merle. Robust prediction of dense gas flows under uncertain thermodynamic models. 2019. (Cited on pages 56, 57 and 89.)

[74] Philip D. Meyer, Ming Ye, Mark L. Rockhold, Shlomo P. Neuman, and Kirk J. Cantrell. Combined Estimation of Hydrogeologic Conceptual Model, Parameter, and Scenario Uncertainty with Application to Uranium Transport at the Hanford Site 300 Area. Technical Report NUREG/CR-6940, PNNL-16396, 921264, July 2007. (Cited on pages 26 and 58.)

[75] Habib N. Najm. Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics. *Annual Review of Fluid Mechanics*, 41(Volume 41, 2009):35–52, 2009. Publisher: Annual Reviews. (Cited on pages 18 and 57.)

[76] William L. Oberkampf and Frederick G. Blottner. Issues in Computational Fluid Dynamics Code Verification and Validation. *AIAA Journal*, 36(5):687–695, May 1998. Publisher: American Institute of Aeronautics and Astronautics. (Cited on page 17.)

[77] Todd A Oliver and Robert D Moser. Bayesian uncertainty quantification applied to RANS turbulence models. *Journal of Physics: Conference Series*, 318(4):042032, December 2011. (Cited on pages 25 and 54.)

[78] Kheir-Eddine Otmani, Gerasimos Ntoukas, and Esteban Ferrer. towards a robust detection of viscous and turbulent flow regions using unsupervised machine learning. *Physics of Fluids*, 35(2):027112, February 2023. arXiv:2207.02929 [physics]. (Cited on page 121.)

[79] 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, January 2016. (Cited on page 21.)

[80] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, and David Cournapeau. Scikit-learn: Machine Learning in Python. *MACHINE LEARNING IN PYTHON*. (Cited on pages 66, 84, 120 and 125.)

[81] Jacques Peter and Eric Savin. Generalized polynomial chaos and stochastic collocation methods for uncertainty quantification in aerodynamics. *VKI course on Uncertainty Quantification in Computational Fluid Dynamics*, 2019. (Cited on page 57.)

[82] S. B. Pope. A more general effective-viscosity hypothesis. *Journal of Fluid Mechanics*, 72(02):331, November 1975. (Cited on page 23.)

[83] Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, 1 edition, August 2000. (Cited on pages 18, 35 and 170.)

[84] Adrian E Raftery, Tilmann Gneiting, Fadoua Balabdaoui, and Michael Polakowski. Using Bayesian Model Averaging to Calibrate Forecast Ensembles. *MONTHLY WEATHER REVIEW*, 133:20, 2005. (Cited on page 26.)

[85] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 3. print edition, 2008. (Cited on page 66.)

[86] Rodrigo Rojas, Samalie Kahunde, Luk Peeters, Okke Batelaan, Luc Feyen, and Alain Dassargues. Application of a multimodel approach to account for conceptual model and scenario uncertainties in groundwater modelling. *Journal of Hydrology*, 394(3-4):416–435, November 2010. (Cited on page 26.)

[87] Eric Savin, Andrea Resmini, and Jacques E. Peter. Sparse polynomial surrogates for aerodynamic computations with random inputs. In *18th AIAA Non-Deterministic Approaches Conference*. American Institute of Aeronautics and Astronautics, 2016. _eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2016-0433. (Cited on page 57.)

[88] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A Review of Clustering Techniques and Developments. (Cited on page 124.)

[89] Martin Schmelzer, Richard P. Dwight, and Paola Cinnella. Discovery of Algebraic Reynolds-Stress Models Using Sparse Symbolic Regression. *Flow, Turbulence and Combustion*, 104(2-3):579–603, March 2020. (Cited on page 23.)

[90] Pranay Seshadri and Geoffrey Parks. Effective-Quadratures (EQ): Polynomials for Computational Engineering Studies. *The Journal of Open Source Software*, 2(11):166, March 2017. (Cited on page 57.)

[91] Anand Pratap Singh and Karthik Duraisamy. Using field inversion to quantify functional errors in turbulence closures. *Physics of Fluids*, 28(4):045110, April 2016. (Cited on page 21.)

[92] Anand Pratap Singh, Racheet Matai, Asitav Mishra, Karthikeyan Duraisamy, and Paul A. Durbin. Data-driven augmentation of turbulence models for adverse pressure gradient flows. In *23rd AIAA Computational Fluid Dynamics Conference*, Denver, Colorado, June 2017. American Institute of Aeronautics and Astronautics. (Cited on page 21.)

[93] Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy. Machine-Learning-Augmented Predictive Modeling of Turbulent Separated Flows over Airfoils. *AIAA Journal*, 55(7):2215–2227, July 2017. (Cited on page 21.)

[94] Brian Smith. The k-kl turbulence model and wall layer model for compressible flows. In *21st Fluid Dynamics, Plasma Dynamics and Lasers Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, June 1990. (Cited on page 38.)

[95] Brian Smith. A near wall model for the k - l two equation turbulence model. 1994. (Cited on pages 38, 79, 88, 139 and 174.)

[96] P R Spalart. Strategies for turbulence modelling and simulations. 2000. (Cited on page 40.)

[97] P.R. Spalart and S.R. Allmaras. One-equation turbulence model for aerodynamic flows. *Recherche aerospatiale*, 1:5–21, 1994. (Cited on pages 36, 79, 88, 139 and 174.)

[98] Gilles 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. 151(2), 2010. (Cited on page 27.)

[99] Carlos A. Michelén Ströfer and Heng Xiao. End-to-end differentiable learning of turbulence models from indirect observations. *Theoretical and Applied Mechanics Letters*, 11(4):100280, May 2021. (Cited on page 23.)

[100] Luning Sun and Jian-Xun Wang. Physics-Constrained Bayesian Neural Network for Fluid Flow Reconstruction with Sparse and Noisy Data. *arXiv:2001.05542 [physics]*, January 2020. arXiv: 2001.05542. (Cited on page 57.)

[101] Brendan Tracey, Karthik Duraisamy, and Juan Alonso. Application of Supervised Learning to Quantify Uncertainties in Turbulence and Combustion Modeling. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine

(Dallas/Ft. Worth Region), Texas, January 2013. American Institute of Aeronautics and Astronautics. (Cited on page 22.)

[102] Stefan Wallin and Arne V. Johansson. An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows. *Journal of Fluid Mechanics*, 403:89–132, January 2000. (Cited on pages 41 and 42.)

[103] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Physical Review Fluids*, 2(3):034603, March 2017. (Cited on page 84.)

[104] Jie Wang. An Intuitive Tutorial to Gaussian Processes Regression, April 2022. arXiv:2009.10862 [cs, stat]. (Cited on page 66.)

[105] 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, November 2016. (Cited on page 23.)

[106] James HI Weygandt and Rabindra D Mehta. ASYMPTOTIC BEHAVIOR OF A FLAT PLATE WAKE. (Cited on page 71.)

[107] K Wieghardt and W Tillmann. TECHNICAL MEMORANDUM 1314. 1951. (Cited on pages 70 and 71.)

[108] David C. Wilcox. *Turbulence modeling for CFD*. DCW Industries, La Cãnada, Calif, 3rd ed edition, 2006. (Cited on pages 33, 37, 79, 88, 139 and 176.)

[109] Zhao Wu, Jin Lee, Charles Meneveau, and Tamer Zaki. Application of a self-organizing map to identify the turbulent-boundary-layer interface in a transitional flow. *Physical Review Fluids*, 4(2):023902, February 2019. (Cited on pages 120 and 121.)

[110] Heng Xiao and Paola Cinnella. Quantification of model uncertainty in RANS simulations: A review. *Progress in Aerospace Sciences*, 108:1–31, July 2019. (Cited on pages 18, 19, 20 and 21.)

[111] Heng Xiao, Jian-Xun Wang, and Roger G. Ghanem. A random matrix approach for quantifying model-form uncertainties in turbulence modeling. *Computer Methods in Applied Mechanics and Engineering*, 313:941–965, January 2017. (Cited on page 23.)

[112] Qingzhao Yu, Steven N. MacEachern, and Mario Peruggia. Bayesian Synthesis: Combining subjective analyses, with an application to ozone data. *The Annals of Applied Statistics*, 5(2B), June 2011. (Cited on page 54.)

[113] Qingzhao Yu, Steven N. MacEachern, and Mario Peruggia. Clustered Bayesian Model Averaging. *Bayesian Analysis*, 8(4), December 2013. (Cited on pages 28, 29, 56, 60, 61 and 84.)

[114] E. Zio and G.E. Apostolakis. Two methods for the structured assessment of model uncertainty by experts in performance assessments of radioactive waste repositories. *Reliability Engineering & System Safety*, 54(2-3):225–241, November 1996. (Cited on pages 17, 24 and 26.)